

License Name: Boost Software License 1.0

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:

- minmax.hpp
- minmax_element.hpp
- case_conv.hpp
- classification.hpp
- compare.hpp
- concept.hpp
- config.hpp
- constants.hpp
- case_conv.hpp
- classification.hpp
- finder.hpp
- finder_regex.hpp
- find_format.hpp
- find_format_all.hpp
- find_format_store.hpp
- find_iterator.hpp
- formatter.hpp
- formatter_regex.hpp
- predicate.hpp
- replace_storage.hpp

sequence.hpp
trim.hpp
util.hpp
erase.hpp
find.hpp
finder.hpp
find_format.hpp
find_iterator.hpp
formatter.hpp
iter_find.hpp
join.hpp
predicate.hpp
predicate_facade.hpp
regex.hpp
regex_find_format.hpp
replace.hpp
sequence_traits.hpp
split.hpp
list_traits.hpp
rope_traits.hpp
slist_traits.hpp
string_traits.hpp
std_containers_traits.hpp
trim.hpp
yes_no_type.hpp
string.hpp
string_regex.hpp
aligned_storage.hpp
any.hpp
add_facet.hpp
archive_exception.hpp
basic_archive.hpp
basic_binary_iarchive.hpp
basic_binary_iprimitive.hpp
basic_binary_oarchive.hpp
basic_binary_oprimitive.hpp
basic_streambuf_locale_saver.hpp
basic_text_iarchive.hpp
basic_text_iprimitive.hpp
basic_text_oarchive.hpp
basic_text_oprimitive.hpp
basic_xml_archive.hpp
basic_xml_iarchive.hpp

basic_xml_oarchive.hpp
binary_iarchive.hpp
binary_iarchive_impl.hpp
binary_oarchive.hpp
binary_oarchive_impl.hpp
binary_wiarchive.hpp
binary_woarchive.hpp
codecvt_null.hpp
abi_prefix.hpp
abi_suffix.hpp
archive_serializer_map.hpp
auto_link_archive.hpp
auto_link_warchive.hpp
basic_archive_impl.hpp
basic_config.hpp
basic_iarchive.hpp
basic_iserializer.hpp
basic_oarchive.hpp
basic_oserIALIZER.hpp
basic_pointer_iserializer.hpp
basic_pointer_oserIALIZER.hpp
basic_serializer.hpp
basic_serializer_map.hpp
check.hpp
common_iarchive.hpp
common_oarchive.hpp
decl.hpp
interface_iarchive.hpp
interface_oarchive.hpp
iserializer.hpp
oserIALIZER.hpp
polymorphic_iarchive_route.hpp
polymorphic_oarchive_route.hpp
register_archive.hpp
utf8_codecvt_facet.hpp
dinkumware.hpp
archive_serializer_map.ipp
basic_binary_iarchive.ipp
basic_binary_iprimitive.ipp
basic_binary_oarchive.ipp
basic_binary_oprimitive.ipp
basic_text_iarchive.ipp
basic_text_iprimitive.ipp

basic_text_oarchive.hpp
basic_text_oprimitive.hpp
basic_xml_grammar.hpp
basic_xml_iarchive.hpp
basic_xml_oarchive.hpp
text_iarchive_impl.hpp
text_oarchive_impl.hpp
text_wiarchive_impl.hpp
text_woarchive_impl.hpp
xml_iarchive_impl.hpp
xml_oarchive_impl.hpp
xml_wiarchive_impl.hpp
xml_woarchive_impl.hpp
base64_exception.hpp
base64_from_binary.hpp
binary_from_base64.hpp
dataflow.hpp
dataflow_exception.hpp
escape.hpp
head_iterator.hpp
insert_linebreaks.hpp
istream_iterator.hpp
mb_from_wchar.hpp
ostream_iterator.hpp
remove_whitespace.hpp
transform_width.hpp
unescape.hpp
wchar_from_mb.hpp
xml_escape.hpp
xml_unescape.hpp
xml_unescape_exception.hpp
polymorphic_binary_iarchive.hpp
polymorphic_binary_oarchive.hpp
polymorphic_iarchive.hpp
polymorphic_oarchive.hpp
polymorphic_text_iarchive.hpp
polymorphic_text_oarchive.hpp
polymorphic_text_wiarchive.hpp
polymorphic_text_woarchive.hpp
polymorphic_xml_iarchive.hpp
polymorphic_xml_oarchive.hpp
polymorphic_xml_wiarchive.hpp
polymorphic_xml_woarchive.hpp

shared_ptr_helper.hpp
text_iarchive.hpp
text_oarchive.hpp
text_wiarchive.hpp
text_woarchive.hpp
tmpdir.hpp
wcslen.hpp
xml_archive_exception.hpp
xml_iarchive.hpp
xml_oarchive.hpp
xml_wiarchive.hpp
xml_woarchive.hpp
array.hpp
basic_datagram_socket.hpp
basic_deadline_timer.hpp
basic_io_object.hpp
basic_raw_socket.hpp
basic_serial_port.hpp
basic_socket.hpp
basic_socket_acceptor.hpp
basic_socket_iostream.hpp
basic_socket_streambuf.hpp
basic_streambuf.hpp
basic_streambuf_fwd.hpp
basic_stream_socket.hpp
buffer.hpp
buffered_read_stream.hpp
buffered_read_stream_fwd.hpp
buffered_stream.hpp
buffered_stream_fwd.hpp
buffered_write_stream.hpp
buffered_write_stream_fwd.hpp
buffers_iterator.hpp
completion_condition.hpp
datagram_socket_service.hpp
deadline_timer.hpp
deadline_timer_service.hpp
array_fwd.hpp
base_from_completion_cond.hpp
bind_handler.hpp
buffered_stream_storage.hpp
buffer_resize_guard.hpp
buffer_sequence_adapter.hpp

call_stack.hpp
completion_handler.hpp
config.hpp
consuming_buffers.hpp
deadline_timer_service.hpp
descriptor_ops.hpp
descriptor_read_op.hpp
descriptor_write_op.hpp
dev_poll_reactor.hpp
dev_poll_reactor_fwd.hpp
epoll_reactor.hpp
epoll_reactor_fwd.hpp
event.hpp
eventfd_select_interrupter.hpp
fd_set_adapter.hpp
fenced_block.hpp
gcc_arm_fenced_block.hpp
gcc_hppa_fenced_block.hpp
gcc_sync_fenced_block.hpp
gcc_x86_fenced_block.hpp
handler_alloc_helpers.hpp
handler_invoke_helpers.hpp
hash_map.hpp
descriptor_ops.hpp
dev_poll_reactor.hpp
dev_poll_reactor.hpp
epoll_reactor.hpp
epoll_reactor.hpp
eventfd_select_interrupter.hpp
kqueue_reactor.hpp
kqueue_reactor.hpp
pipe_select_interrupter.hpp
posix_event.hpp
posix_mutex.hpp
posix_thread.hpp
posix_tss_ptr.hpp
reactive_descriptor_service.hpp
reactive_serial_port_service.hpp
reactive_socket_service_base.hpp
resolver_service_base.hpp
select_reactor.hpp
select_reactor.hpp
service_registry.hpp

service_registry.ipp
socket_ops.ipp
socket_select_interrupter.ipp
strand_service.hpp
strand_service.ipp
task_io_service.hpp
task_io_service.ipp
throw_error.ipp
timer_queue.ipp
timer_queue_set.ipp
winsock_init.ipp
win_event.ipp
win_iocp_handle_service.ipp
win_iocp_io_service.hpp
win_iocp_io_service.ipp
win_iocp_serial_port_service.ipp
win_iocp_socket_service_base.ipp
win_mutex.ipp
win_thread.ipp
win_tss_ptr.ipp
io_control.hpp
kqueue_reactor.hpp
kqueue_reactor_fwd.hpp
local_free_on_block_exit.hpp
macos_fenced_block.hpp
mutex.hpp
noncopyable.hpp
null_event.hpp
null_fenced_block.hpp
null_mutex.hpp
null_signal_blocker.hpp
null_thread.hpp
null_tss_ptr.hpp
object_pool.hpp
old_win_sdk_compat.hpp
operation.hpp
op_queue.hpp
pipe_select_interrupter.hpp
pop_options.hpp
posix_event.hpp
posix_fd_set_adapter.hpp
posix_mutex.hpp
posix_signal_blocker.hpp

posix_thread.hpp
posix_tss_ptr.hpp
push_options.hpp
reactive_descriptor_service.hpp
reactive_null_buffers_op.hpp
reactive_serial_port_service.hpp
reactive_socket_accept_op.hpp
reactive_socket_connect_op.hpp
reactive_socket_recvfrom_op.hpp
reactive_socket_recv_op.hpp
reactive_socket_sendto_op.hpp
reactive_socket_send_op.hpp
reactive_socket_service.hpp
reactive_socket_service_base.hpp
reactor.hpp
reactor_fwd.hpp
reactor_op.hpp
reactor_op_queue.hpp
regex_fwd.hpp
resolver_service.hpp
resolver_service_base.hpp
resolve_endpoint_op.hpp
resolve_op.hpp
scoped_lock.hpp
select_interrupter.hpp
select_reactor.hpp
select_reactor_fwd.hpp
service_registry.hpp
service_registry_fwd.hpp
shared_ptr.hpp
signal_blocker.hpp
signal_init.hpp
socket_holder.hpp
socket_ops.hpp
socket_option.hpp
socket_select_interrupter.hpp
socket_types.hpp
solaris_fenced_block.hpp
strand_service.hpp
task_io_service.hpp
task_io_service_fwd.hpp
task_io_service_operation.hpp
thread.hpp

throw_error.hpp
timer_op.hpp
timer_queue.hpp
timer_queue_base.hpp
timer_queue_fwd.hpp
timer_queue_set.hpp
timer_scheduler.hpp
timer_scheduler_fwd.hpp
tss_ptr.hpp
wait_handler.hpp
weak_ptr.hpp
wince_thread.hpp
winsock_init.hpp
win_event.hpp
win_fd_set_adapter.hpp
win_fenced_block.hpp
win_iocp_handle_read_op.hpp
win_iocp_handle_service.hpp
win_iocp_handle_write_op.hpp
win_iocp_io_service.hpp
win_iocp_io_service_fwd.hpp
win_iocp_null_buffers_op.hpp
win_iocp_operation.hpp
win_iocp_overlapped_op.hpp
win_iocp_overlapped_ptr.hpp
win_iocp_serial_port_service.hpp
win_iocp_socket_accept_op.hpp
win_iocp_socket_recvfrom_op.hpp
win_iocp_socket_recv_op.hpp
win_iocp_socket_send_op.hpp
win_iocp_socket_service.hpp
win_iocp_socket_service_base.hpp
win_mutex.hpp
win_thread.hpp
win_tss_ptr.hpp
wrapped_handler.hpp
error.hpp
handler_alloc_hook.hpp
handler_invoke_hook.hpp
error.ipp
io_service.hpp
io_service.ipp
read.hpp

read_at.hpp
read_until.hpp
serial_port_base.hpp
serial_port_base.hpp
src.cpp
write.hpp
write_at.hpp
io_service.hpp
address.hpp
address_v4.hpp
address_v6.hpp
basic_endpoint.hpp
basic_resolver.hpp
basic_resolver_entry.hpp
basic_resolver_iterator.hpp
basic_resolver_query.hpp
endpoint.hpp
endpoint.hpp
socket_option.hpp
host_name.hpp
icmp.hpp
address.hpp
address.hpp
address_v4.hpp
address_v4.hpp
address_v6.hpp
address_v6.hpp
basic_endpoint.hpp
host_name.hpp
multicast.hpp
resolver_query_base.hpp
resolver_service.hpp
tcp.hpp
udp.hpp
unicast.hpp
v6_only.hpp
is_read_buffered.hpp
is_write_buffered.hpp
basic_endpoint.hpp
connect_pair.hpp
datagram_protocol.hpp
endpoint.hpp
endpoint.hpp

stream_protocol.hpp
placeholders.hpp
basic_descriptor.hpp
basic_stream_descriptor.hpp
descriptor_base.hpp
stream_descriptor.hpp
stream_descriptor_service.hpp
raw_socket_service.hpp
read.hpp
read_at.hpp
read_until.hpp
serial_port.hpp
serial_port_base.hpp
serial_port_service.hpp
socket_acceptor_service.hpp
socket_base.hpp
basic_context.hpp
context.hpp
context_base.hpp
context_service.hpp
openssl_context_service.hpp
openssl_init.hpp
openssl_operation.hpp
openssl_stream_service.hpp
openssl_types.hpp
stream.hpp
stream_base.hpp
stream_service.hpp
ssl.hpp
strand.hpp
streambuf.hpp
stream_socket_service.hpp
time_traits.hpp
version.hpp
basic_handle.hpp
basic_random_access_handle.hpp
basic_stream_handle.hpp
overlapped_ptr.hpp
random_access_handle.hpp
random_access_handle_service.hpp
stream_handle.hpp
stream_handle_service.hpp
write.hpp

write_at.hpp
asio.hpp
assert.hpp
apply.hpp
arg.hpp
bind.hpp
bind_cc.hpp
bind_mf2_cc.hpp
bind_mf_cc.hpp
bind_template.hpp
make_adaptable.hpp
mem_fn.hpp
mem_fn_cc.hpp
mem_fn_template.hpp
mem_fn_vw.hpp
placeholders.hpp
protect.hpp
storage.hpp
bind.hpp
blank.hpp
blank_fwd.hpp
call_traits.hpp
cerrno.hpp
checked_delete.hpp
base.hpp
debug.hpp
details.hpp
space_optimized.hpp
circular_buffer.hpp
circular_buffer_fwd.hpp
compressed_pair.hpp
assert.hpp
backward_compatibility.hpp
borland.hpp
concept_def.hpp
concept_undef.hpp
general.hpp
has_constraints.hpp
msvc.hpp
requires.hpp
usage.hpp
borland.hpp
general.hpp

has_constraints.hpp
msvc.hpp
concept_check.hpp
borland_prefix.hpp
borland_suffix.hpp
msvc_prefix.hpp
msvc_suffix.hpp
abi_prefix.hpp
abi_suffix.hpp
auto_link.hpp
borland.hpp
clang.hpp
codegear.hpp
comeau.hpp
common_edg.hpp
compaq_cxx.hpp
digitalmars.hpp
gcc.hpp
gcc_xml.hpp
greenhills.hpp
hp_acc.hpp
intel.hpp
kai.hpp
metrowerks.hpp
mpw.hpp
nvcc.hpp
pgi.hpp
sgi_mipspro.hpp
sunpro_cc.hpp
vacpp.hpp
visualc.hpp
cmath.hpp
complex.hpp
functional.hpp
memory.hpp
utility.hpp
aix.hpp
amigaos.hpp
beos.hpp
bsd.hpp
cygwin.hpp
hpux.hpp
irix.hpp

linux.hpp
macos.hpp
qnxnto.hpp
solaris.hpp
symbian.hpp
vxworks.hpp
win32.hpp
posix_features.hpp
requires_threads.hpp
select_compiler_config.hpp
select_platform_config.hpp
select_stdlib_config.hpp
dinkumware.hpp
libcomo.hpp
libstdcpp3.hpp
modena.hpp
msl.hpp
roguewave.hpp
sgi.hpp
stlport.hpp
vacpp.hpp
suffix.hpp
user.hpp
warning_disable.hpp
config.hpp
cregex.hpp
cstdint.hpp
current_function.hpp
adjust_functors.hpp
compiler_config.hpp
constrained_value.hpp
c_local_time_adjustor.hpp
c_time.hpp
date.hpp
date_clock_device.hpp
date_defs.hpp
date_duration.hpp
date_duration_types.hpp
date_facet.hpp
date_formatting.hpp
date_formatting_limited.hpp
date_formatting_locales.hpp
date_format_simple.hpp

date_generators.hpp
date_generator_formatter.hpp
date_generator_parser.hpp
date_iterator.hpp
date_names_put.hpp
date_parsing.hpp
dst_rules.hpp
dst_transition_generators.hpp
filetime_functions.hpp
format_date_parser.hpp
conversion.hpp
formatters.hpp
formatters_limited.hpp
gregorian.hpp
gregorian_io.hpp
gregorian_types.hpp
greg_calendar.hpp
greg_date.hpp
greg_day.hpp
greg_day_of_year.hpp
greg_duration.hpp
greg_duration_types.hpp
greg_facet.hpp
greg_month.hpp
greg_serialize.hpp
greg_weekday.hpp
greg_year.hpp
greg_ymd.hpp
parsers.hpp
gregorian_calendar.hpp
gregorian_calendar.ipp
int_adapter.hpp
iso_format.hpp
locale_config.hpp
conversion.hpp
custom_time_zone.hpp
date_duration_operators.hpp
dst_transition_day_rules.hpp
local_date_time.hpp
local_time.hpp
local_time_io.hpp
local_time_types.hpp
posix_time_zone.hpp

tz_database.hpp
local_timezone_defs.hpp
local_time_adjustor.hpp
microsec_time_clock.hpp
parse_format_base.hpp
period.hpp
period_formatter.hpp
period_parser.hpp
conversion.hpp
date_duration_operators.hpp
posix_time.hpp
posix_time_config.hpp
posix_time_duration.hpp
posix_time_io.hpp
posix_time_legacy_io.hpp
posix_time_system.hpp
posix_time_types.hpp
ptime.hpp
time_formatters.hpp
time_formatters_limited.hpp
time_parsers.hpp
time_period.hpp
time_serialize.hpp
special_defs.hpp
special_values_formatter.hpp
special_values_parser.hpp
strings_from_facet.hpp
string_convert.hpp
string_parse_tree.hpp
time.hpp
time_clock.hpp
time_defs.hpp
time_duration.hpp
time_facet.hpp
time_formatting_streams.hpp
time_iterator.hpp
time_parsing.hpp
time_resolution_traits.hpp
time_system_counted.hpp
time_system_split.hpp
time_zone_base.hpp
time_zone_names.hpp
tz_db_base.hpp

wrapping_int.hpp
year_month_day.hpp
date_time.hpp
algorithm.hpp
allocator_utilities.hpp
atomic_count.hpp
binary_search.hpp
call_traits.hpp
catch_exceptions.hpp
compressed_pair.hpp
container_fwd.hpp
dynamic_bitset.hpp
endian.hpp
has_default_constructor.hpp
identifier.hpp
indirect_traits.hpp
interlocked.hpp
is_function_ref_tester.hpp
is_incrementable.hpp
is_xxx.hpp
iterator.hpp
lcast_precision.hpp
lightweight_mutex.hpp
lightweight_test.hpp
lightweight_thread.hpp
limits.hpp
named_template_params.hpp
none_t.hpp
no_exceptions_support.hpp
numeric_traits.hpp
ob_call_traits.hpp
ob_compressed_pair.hpp
quick_allocator.hpp
reference_content.hpp
scoped_enum_emulation.hpp
select_type.hpp
sp_typeinfo.hpp
templated_streams.hpp
utf8_codecvt_facet.hpp
workaround.hpp
enable_shared_from_this.hpp
all.hpp
current_exception_cast.hpp

attribute_noreturn.hpp
error_info_impl.hpp
exception_ptr.hpp
is_output_streamable.hpp
object_hex_dump.hpp
type_info.hpp
diagnostic_information.hpp
enable_current_exception.hpp
enable_error_info.hpp
errinfo_api_function.hpp
errinfo_at_line.hpp
errinfo_errno.hpp
errinfo_file_handle.hpp
errinfo_file_name.hpp
errinfo_file_open_mode.hpp
errinfo_nested_exception.hpp
errinfo_type_info_name.hpp
error_info.hpp
exception.hpp
get_error_info.hpp
info.hpp
info_tuple.hpp
to_string.hpp
to_string_stub.hpp
exception_ptr.hpp
config.hpp
convenience.hpp
utf8_codecvt_facet.hpp
exception.hpp
fstream.hpp
operations.hpp
path.hpp
config.hpp
convenience.hpp
exception.hpp
fstream.hpp
operations.hpp
path.hpp
config.hpp
convenience.hpp
exception.hpp
fstream.hpp
operations.hpp

path.hpp
path_traits.hpp
filesystem.hpp
foreach.hpp
foreach_fwd.hpp
function_iterate.hpp
maybe_include.hpp
prologue.hpp
function_base.hpp
function_fwd.hpp
function_template.hpp
function.hpp
container_fwd.hpp
factory.hpp
forward_adapter.hpp
float_functions.hpp
hash_float.hpp
hash_float_generic.hpp
hash_float_x86.hpp
limits.hpp
extensions.hpp
hash.hpp
hash_fwd.hpp
hash.hpp
hash_fwd.hpp
lightweight_forward_adapter.hpp
value_factory.hpp
function_equal.hpp
get_pointer.hpp
algorithm.hpp
bit_aligned_pixel_iterator.hpp
bit_aligned_pixel_reference.hpp
channel.hpp
channel_algorithm.hpp
cmyk.hpp
color_base.hpp
color_base_algorithm.hpp
color_convert.hpp
deprecated.hpp
device_n.hpp
algorithm.hpp
any_image.hpp
any_image_view.hpp

apply_operation.hpp
apply_operation_base.hpp
dynamic_at_c.hpp
dynamic_image_all.hpp
image_view_factory.hpp
reduce.hpp
variant.hpp
dynamic_io.hpp
io_error.hpp
jpeg_dynamic_io.hpp
jpeg_io.hpp
jpeg_io_private.hpp
png_dynamic_io.hpp
png_io.hpp
png_io_private.hpp
tiff_dynamic_io.hpp
tiff_io.hpp
gil_all.hpp
gil_concept.hpp
gil_config.hpp
gray.hpp
image.hpp
image_view.hpp
image_view_factory.hpp
iterator_from_2d.hpp
locator.hpp
metafunctions.hpp
packed_pixel.hpp
pixel.hpp
pixel_iterator.hpp
pixel_iterator_adaptor.hpp
planar_pixel_iterator.hpp
planar_pixel_reference.hpp
position_iterator.hpp
rgb.hpp
rgba.hpp
step_iterator.hpp
typedefs.hpp
utilities.hpp
virtual_locator.hpp
implicit_cast.hpp
indirect_reference.hpp
integer_mask.hpp

static_log2.hpp
static_min_max.hpp
integer.hpp
integer_fwd.hpp
integer_traits.hpp
intrusive_ptr.hpp
quoted_manip.hpp
ios_state.hpp
io_fwd.hpp
is_placeholder.hpp
counting_iterator.hpp
any_conversion_eater.hpp
config_def.hpp
config_undef.hpp
enable_if.hpp
facade_iterator_category.hpp
minimum_category.hpp
filter_iterator.hpp
indirect_iterator.hpp
interoperable.hpp
is_lvalue_iterator.hpp
is_readable_iterator.hpp
iterator_adaptor.hpp
iterator_archetypes.hpp
iterator_categories.hpp
iterator_concepts.hpp
iterator_facade.hpp
iterator_traits.hpp
new_iterator_tests.hpp
permutation_iterator.hpp
reverse_iterator.hpp
transform_iterator.hpp
zip_iterator.hpp
iterator.hpp
iterator_adaptors.hpp
algorithm.hpp
bind.hpp
casts.hpp
closures.hpp
construct.hpp
control_structures.hpp
core.hpp
actions.hpp

arity_code.hpp
bind_functions.hpp
control_constructs_common.hpp
function_adaptors.hpp
is_instance_of.hpp
lambda_config.hpp
lambda_functors.hpp
lambda_functor_base.hpp
lambda_fwd.hpp
lambda_traits.hpp
member_ptr.hpp
operators.hpp
operator_actions.hpp
operator_lambda_func_base.hpp
operator_return_type_traits.hpp
ret.hpp
return_type_traits.hpp
select_functions.hpp
suppress_unused.hpp
exceptions.hpp
if.hpp
lambda.hpp
loops.hpp
numeric.hpp
switch.hpp
lexical_cast.hpp
limits.hpp
memory_order.hpp
mem_fn.hpp
accumulate.hpp
advance.hpp
advance_fwd.hpp
alias.hpp
always.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
arg_fwd.hpp
arithmetic.hpp
assert.hpp
as_sequence.hpp

at.hpp
at_fwd.hpp
adl_barrier.hpp
advance_backward.hpp
advance_forward.hpp
apply_1st.hpp
arg_typedef.hpp
arithmetic_op.hpp
arity.hpp
arity_spec.hpp
at_impl.hpp
back_impl.hpp
basic_bind.hpp
begin_end_impl.hpp
clear_impl.hpp
common_name_wknd.hpp
comparison_op.hpp
adl.hpp
arrays.hpp
bcc.hpp
bind.hpp
compiler.hpp
ctps.hpp
dependent_nttp.hpp
dmc_ambiguous_ctps.hpp
dtp.hpp
eti.hpp
forwarding.hpp
gcc.hpp
has_apply.hpp
has_xxx.hpp
integral.hpp
intel.hpp
lambda.hpp
msvc.hpp
msvc_typename.hpp
nttp.hpp
operators.hpp
overload_resolution.hpp
pp_counter.hpp
preprocessor.hpp
static_constant.hpp
ttp.hpp

typeof.hpp
use_preprocessed.hpp
workaround.hpp
contains_impl.hpp
count_args.hpp
count_impl.hpp
empty_impl.hpp
erase_impl.hpp
erase_key_impl.hpp
filter_iter.hpp
find_if_pred.hpp
fold_impl.hpp
fold_impl_body.hpp
fold_op.hpp
fold_pred.hpp
front_impl.hpp
full_lambda.hpp
has_apply.hpp
has_begin.hpp
has_key_impl.hpp
has_rebind.hpp
has_size.hpp
has_tag.hpp
has_type.hpp
include_preprocessed.hpp
inserter_algorithm.hpp
insert_impl.hpp
insert_range_impl.hpp
integral_wrapper.hpp
is_msvc_eti_arg.hpp
iter_apply.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
iter_push_front.hpp
joint_iter.hpp
lambda_arity_param.hpp
lambda_no_ctps.hpp
lambda_spec.hpp
lambda_support.hpp
largest_int.hpp
logical_op.hpp
msvc_dtw.hpp
msvc_eti_base.hpp

msvc_is_class.hpp
msvc_never_true.hpp
msvc_type.hpp
na.hpp
na_assert.hpp
na_fwd.hpp
na_spec.hpp
nested_type_wknd.hpp
nttp_decl.hpp
numeric_cast_utils.hpp
numeric_op.hpp
O1_size_impl.hpp
order_impl.hpp
overload_names.hpp
partition_op.hpp
pop_back_impl.hpp
pop_front_impl.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp

list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp

iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp

greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp

divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp

bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp

arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp

and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp

vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp

shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp

reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp

placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp

minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
add.hpp
default_params.hpp
def_params_tail.hpp
enum.hpp
ext_params.hpp
filter_params.hpp
is_seq.hpp
params.hpp
partial_spec_params.hpp
range.hpp
repeat.hpp
sub.hpp
token_equal.hpp
tuple.hpp
ptr_to_ref.hpp
push_back_impl.hpp
push_front_impl.hpp
back.hpp
empty.hpp
front.hpp
iterator.hpp
O1_size.hpp
size.hpp
tag.hpp
reverse_fold_impl.hpp

reverse_fold_impl_body.hpp
reverse_iter_fold_impl.hpp
sequence_wrapper.hpp
shift_op.hpp
single_element_iter.hpp
size_impl.hpp
sort_impl.hpp
static_cast.hpp
template_arity.hpp
template_arity_fwd.hpp
assert.hpp
data.hpp
test_case.hpp
test.hpp
traits_lambda_spec.hpp
transform_iter.hpp
type_wrapper.hpp
unwrap.hpp
value_wknd.hpp
yes_no.hpp
back.hpp
back_fwd.hpp
back_inserter.hpp
base.hpp
begin.hpp
begin_end.hpp
begin_end_fwd.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitwise.hpp
bitxor.hpp
bool.hpp
bool_fwd.hpp
char.hpp
char_fwd.hpp
clear.hpp
clear_fwd.hpp
comparison.hpp
contains.hpp
contains_fwd.hpp
copy.hpp

copy_if.hpp
count.hpp
count_fwd.hpp
count_if.hpp
deque.hpp
deref.hpp
distance.hpp
distance_fwd.hpp
divides.hpp
empty.hpp
empty_base.hpp
empty_fwd.hpp
empty_sequence.hpp
end.hpp
equal.hpp
equal_to.hpp
erase.hpp
erase_fwd.hpp
erase_key.hpp
erase_key_fwd.hpp
eval_if.hpp
filter_view.hpp
find.hpp
find_if.hpp
fold.hpp
for_each.hpp
front.hpp
front_fwd.hpp
front_inserter.hpp
greater.hpp
greater_equal.hpp
has_key.hpp
has_key_fwd.hpp
has_xxx.hpp
identity.hpp
if.hpp
index_if.hpp
index_of.hpp
inherit.hpp
inherit_linearly.hpp
insert.hpp
inserter.hpp
insert_fwd.hpp

insert_range.hpp
insert_range_fwd.hpp
int.hpp
integral_c.hpp
integral_c_fwd.hpp
integral_c_tag.hpp
int_fwd.hpp
is_placeholder.hpp
is_sequence.hpp
iterator_category.hpp
iterator_range.hpp
iterator_tags.hpp
iter_fold.hpp
iter_fold_if.hpp
joint_view.hpp
key_type.hpp
key_type_fwd.hpp
lambda.hpp
lambda_fwd.hpp
less.hpp
less_equal.hpp
arity.hpp
list.hpp
map.hpp
set.hpp
string.hpp
unrolling.hpp
vector.hpp
begin_end.hpp
clear.hpp
empty.hpp
front.hpp
include_preprocessed.hpp
item.hpp
iterator.hpp
numbered.hpp
numbered_c.hpp
O1_size.hpp
pop_front.hpp
list10.hpp
list10_c.hpp
list20.hpp
list20_c.hpp

list30.hpp
list30_c.hpp
list40.hpp
list40_c.hpp
list50.hpp
list50_c.hpp
push_back.hpp
push_front.hpp
size.hpp
tag.hpp
list0.hpp
list0_c.hpp
list10.hpp
list10_c.hpp
list20.hpp
list20_c.hpp
list30.hpp
list30_c.hpp
list40.hpp
list40_c.hpp
list50.hpp
list50_c.hpp
list.hpp
list_c.hpp
logical.hpp
long.hpp
long_fwd.hpp
lower_bound.hpp
at_impl.hpp
begin_end_impl.hpp
clear_impl.hpp
contains_impl.hpp
empty_impl.hpp
erase_impl.hpp
erase_key_impl.hpp
has_key_impl.hpp
include_preprocessed.hpp
insert_impl.hpp
item.hpp
iterator.hpp
key_type_impl.hpp
map0.hpp
numbered.hpp

map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
size_impl.hpp
tag.hpp
value_type_impl.hpp
map0.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
map.hpp
fixed_c.hpp
is_even.hpp
rational_c.hpp
max.hpp
max_element.hpp
min.hpp
minus.hpp
min_element.hpp
min_max.hpp
modulus.hpp
multiplies.hpp
count_impl.hpp
insert_impl.hpp
item.hpp
multiset0.hpp
tag.hpp
multiset0.hpp
negate.hpp

next.hpp
next_prior.hpp
not.hpp
not_equal_to.hpp
numeric_cast.hpp
O1_size.hpp
O1_size_fwd.hpp
or.hpp
order.hpp
order_fwd.hpp
pair.hpp
pair_view.hpp
partition.hpp
placeholders.hpp
plus.hpp
pop_back.hpp
pop_back_fwd.hpp
pop_front.hpp
pop_front_fwd.hpp
print.hpp
prior.hpp
protect.hpp
push_back.hpp
push_back_fwd.hpp
push_front.hpp
push_front_fwd.hpp
quote.hpp
range_c.hpp
remove.hpp
remove_if.hpp
replace.hpp
replace_if.hpp
reverse.hpp
reverse_fold.hpp
reverse_iter_fold.hpp
same_as.hpp
sequence_tag.hpp
sequence_tag_fwd.hpp
at_impl.hpp
begin_end_impl.hpp
clear_impl.hpp
empty_impl.hpp
erase_impl.hpp

erase_key_impl.hpp
has_key_impl.hpp
include_preprocessed.hpp
insert_impl.hpp
item.hpp
iterator.hpp
key_type_impl.hpp
numbered.hpp
numbered_c.hpp
set10.hpp
set10_c.hpp
set20.hpp
set20_c.hpp
set30.hpp
set30_c.hpp
set40.hpp
set40_c.hpp
set50.hpp
set50_c.hpp
set0.hpp
size_impl.hpp
tag.hpp
value_type_impl.hpp
set0.hpp
set0_c.hpp
set10.hpp
set10_c.hpp
set20.hpp
set20_c.hpp
set30.hpp
set30_c.hpp
set40.hpp
set40_c.hpp
set50.hpp
set50_c.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
single_view.hpp
size.hpp
sizeof.hpp
size_fwd.hpp

size_t.hpp
size_t_fwd.hpp
sort.hpp
stable_partition.hpp
string.hpp
switch.hpp
tag.hpp
times.hpp
transform.hpp
transform_view.hpp
unique.hpp
unpack_args.hpp
upper_bound.hpp
value_type.hpp
value_type_fwd.hpp
at.hpp
back.hpp
begin_end.hpp
clear.hpp
empty.hpp
front.hpp
include_preprocessed.hpp
item.hpp
iterator.hpp
numbered.hpp
numbered_c.hpp
O1_size.hpp
pop_back.hpp
pop_front.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp

vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
push_back.hpp
push_front.hpp
size.hpp
tag.hpp
vector0.hpp
vector0_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
vector.hpp
vector_c.hpp
void.hpp
void_fwd.hpp
zip_view.hpp
composite_key.hpp
access_specifier.hpp
adl_swap.hpp
archive_constructed.hpp
auto_space.hpp

base_type.hpp
bidir_node_iterator.hpp
bucket_array.hpp
converter.hpp
copy_map.hpp
duplicates_iterator.hpp
hash_index_args.hpp
hash_index_iterator.hpp
hash_index_node.hpp
has_tag.hpp
header_holder.hpp
index_base.hpp
index_loader.hpp
index_matcher.hpp
index_node_base.hpp
index_saver.hpp
invariant_assert.hpp
is_index_list.hpp
iter_adaptor.hpp
modify_key_adaptor.hpp
msvc_index_specifier.hpp
node_type.hpp
no_duplicate_tags.hpp
ord_index_args.hpp
ord_index_node.hpp
ord_index_ops.hpp
prevent_eti.hpp
rnd_index_loader.hpp
rnd_index_node.hpp
rnd_index_ops.hpp
rnd_index_ptr_array.hpp
rnd_node_iterator.hpp
safe_ctr_proxy.hpp
safe_mode.hpp
scope_guard.hpp
seq_index_node.hpp
seq_index_ops.hpp
serialization_version.hpp
uintptr_type.hpp
unbounded.hpp
value_compare.hpp
global_fun.hpp
hashed_index.hpp

hashed_index_fwd.hpp
identity.hpp
identity_fwd.hpp
indexed_by.hpp
key_extractors.hpp
member.hpp
mem_fun.hpp
ordered_index.hpp
ordered_index_fwd.hpp
random_access_index.hpp
random_access_index_fwd.hpp
safe_mode_errors.hpp
sequenced_index.hpp
sequenced_index_fwd.hpp
tag.hpp
multi_index_container.hpp
multi_index_container_fwd.hpp
next_prior.hpp
noncopyable.hpp
none.hpp
none_t.hpp
bounds.hpp
cast.hpp
conversion_traits.hpp
converter.hpp
converter_policies.hpp
bounds.hpp
conversion_traits.hpp
converter.hpp
int_float_mixture.hpp
is_subranged.hpp
meta.hpp
old_numeric_cast.hpp
sign_mixture.hpp
udt_builtin_mixture.hpp
int_float_mixture.hpp
int_float_mixture_enum.hpp
is_subranged.hpp
sign_mixture.hpp
sign_mixture_enum.hpp
udt_builtin_mixture.hpp
udt_builtin_mixture_enum.hpp
arith.hpp

arith2.hpp
arith3.hpp
checking.hpp
certain.hpp
explicit.hpp
lexicographic.hpp
possible.hpp
set.hpp
tribool.hpp
compare.hpp
constants.hpp
alpha_rounding_control.hpp
bcc_rounding_control.hpp
bugs.hpp
c99sub_rounding_control.hpp
c99_rounding_control.hpp
division.hpp
ia64_rounding_control.hpp
interval_prototype.hpp
msvc_rounding_control.hpp
ppc_rounding_control.hpp
sparc_rounding_control.hpp
test_input.hpp
x86gcc_rounding_control.hpp
x86_rounding_control.hpp
integer.hpp
x86_fast_rounding_control.hpp
hw_rounding.hpp
interval.hpp
io.hpp
limits.hpp
policies.hpp
rounded_arith.hpp
rounded_transc.hpp
rounding.hpp
transc.hpp
utility.hpp
interval.hpp
assignment.hpp
banded.hpp
blas.hpp
concepts.hpp
config.hpp

definitions.hpp
documentation.hpp
duff.hpp
iterator.hpp
matrix_assign.hpp
raw.hpp
returntype_deduction.hpp
temporary.hpp
vector_assign.hpp
doxydoc.hpp
exception.hpp
sparse_view.hpp
expression_types.hpp
functional.hpp
fwd.hpp
hermitian.hpp
io.hpp
lu.hpp
matrix.hpp
matrix_expression.hpp
matrix_proxy.hpp
matrix_sparse.hpp
begin.hpp
c_array.hpp
end.hpp
num_columns.hpp
num_rows.hpp
size.hpp
operation.hpp
operations.hpp
operation_blocked.hpp
operation_sparse.hpp
storage.hpp
storage_sparse.hpp
symmetric.hpp
tags.hpp
const_iterator_type.hpp
c_array.hpp
iterator_type.hpp
traits.hpp
triangular.hpp
vector.hpp
vector_expression.hpp

vector_of_vector.hpp
vector_proxy.hpp
vector_sparse.hpp
operators.hpp
optional.hpp
optional_fwd.hpp
optional_io.hpp
optional.hpp
pointee.hpp
ct_gcd_lcm.hpp
for.m4
gcd_lcm.hpp
guard.hpp
mutex.hpp
pool_construct.inc
pool_construct.m4
pool_construct_simple.inc
pool_construct_simple.m4
singleton.hpp
object_pool.hpp
pool.hpp
pool_fwd.hpp
pool_alloc.hpp
simple_segregated_storage.hpp
singleton_pool.hpp
add.hpp
dec.hpp
div_base.hpp
div.hpp
inc.hpp
mod.hpp
mul.hpp
sub.hpp
arithmetic.hpp
data.hpp
elem.hpp
insert.hpp
pop_back.hpp
pop_front.hpp
push_back.hpp
push_front.hpp
remove.hpp
replace.hpp

reverse.hpp
size.hpp
array.hpp
assert_msg.hpp
cat.hpp
comma.hpp
comma_if.hpp
equal.hpp
greater.hpp
greater_equal.hpp
less.hpp
less_equal.hpp
not_equal.hpp
comparison.hpp
config.hpp
limits.hpp
deduce_d.hpp
while.hpp
while.hpp
while.hpp
while.hpp
expr_if.hpp
expr_iif.hpp
if.hpp
iif.hpp
while.hpp
control.hpp
assert.hpp
error.hpp
line.hpp
debug.hpp
dec.hpp
auto_rec.hpp
check.hpp
auto_rec.hpp
is_binary.hpp
is_nullary.hpp
is_unary.hpp
null.hpp
split.hpp
empty.hpp
enum.hpp
enum_params.hpp

enum_params_with_a_default.hpp
enum_params_with_defaults.hpp
enum_shifted.hpp
enum_shifted_params.hpp
expand.hpp
expr_if.hpp
apply.hpp
empty.hpp
expand.hpp
identity.hpp
intercept.hpp
is_1.hpp
is_empty.hpp
is_empty_or_1.hpp
facilities.hpp
for.hpp
identity.hpp
if.hpp
inc.hpp
iterate.hpp
lower1.hpp
lower2.hpp
lower3.hpp
lower4.hpp
lower5.hpp
upper1.hpp
upper2.hpp
upper3.hpp
upper4.hpp
upper5.hpp
finish.hpp
forward1.hpp
forward2.hpp
forward3.hpp
forward4.hpp
forward5.hpp
reverse1.hpp
reverse2.hpp
reverse3.hpp
reverse4.hpp
reverse5.hpp
local.hpp
rlocal.hpp

self.hpp
start.hpp
iterate.hpp
local.hpp
self.hpp
iteration.hpp
library.hpp
limits.hpp
adt.hpp
append.hpp
at.hpp
cat.hpp
fold_left.hpp
fold_left.hpp
fold_right.hpp
fold_left.hpp
fold_right.hpp
enum.hpp
filter.hpp
first_n.hpp
fold_left.hpp
fold_right.hpp
for_each.hpp
for_each_i.hpp
for_each_product.hpp
rest_n.hpp
reverse.hpp
size.hpp
to_tuple.hpp
transform.hpp
list.hpp
and.hpp
bitand.hpp
bitnor.hpp
bitor.hpp
bitxor.hpp
bool.hpp
compl.hpp
nor.hpp
not.hpp
or.hpp
xor.hpp
logical.hpp

max.hpp
min.hpp
comma.hpp
comma_if.hpp
paren.hpp
paren_if.hpp
punctuation.hpp
repeat.hpp
repeat_2nd.hpp
repeat_3rd.hpp
repeat_from_to.hpp
repeat_from_to_2nd.hpp
repeat_from_to_3rd.hpp
deduce_r.hpp
deduce_z.hpp
for.hpp
for.hpp
for.hpp
for.hpp
enum.hpp
enum_binary_params.hpp
enum_params.hpp
enum_params_with_a_default.hpp
enum_params_with_defaults.hpp
enum_shifted.hpp
enum_shifted_binary_params.hpp
enum_shifted_params.hpp
enum_trailing.hpp
enum_trailing_binary_params.hpp
enum_trailing_params.hpp
for.hpp
repeat.hpp
repeat_from_to.hpp
repetition.hpp
max.hpp
min.hpp
selection.hpp
cat.hpp
split.hpp
elem.hpp
enum.hpp
filter.hpp
first_n.hpp

fold_left.hpp
fold_right.hpp
for_each.hpp
for_each_i.hpp
for_each_product.hpp
insert.hpp
pop_back.hpp
pop_front.hpp
push_back.hpp
push_front.hpp
remove.hpp
replace.hpp
rest_n.hpp
reverse.hpp
seq.hpp
size.hpp
subseq.hpp
to_array.hpp
to_tuple.hpp
transform.hpp
seq.hpp
counter.hpp
counter.hpp
def.hpp
shared.hpp
slot1.hpp
slot2.hpp
slot3.hpp
slot4.hpp
slot5.hpp
slot.hpp
slot.hpp
stringize.hpp
eat.hpp
elem.hpp
rem.hpp
reverse.hpp
to_list.hpp
to_seq.hpp
tuple.hpp
while.hpp
wstringize.hpp
progress.hpp

exception_implementation.hpp
file_parser_error.hpp
info_parser_error.hpp
info_parser_read.hpp
info_parser_utils.hpp
info_parser_write.hpp
info_parser_writer_settings.hpp
json_parser_error.hpp
json_parser_read.hpp
json_parser_write.hpp
ptree_implementation.hpp
ptree_utils.hpp
rapidxml.hpp
xml_parser_error.hpp
xml_parser_flags.hpp
xml_parser_read_rapidxml.hpp
xml_parser_utils.hpp
xml_parser_write.hpp
xml_parser_writer_settings.hpp
exceptions.hpp
id_translator.hpp
info_parser.hpp
ini_parser.hpp
json_parser.hpp
ptree.hpp
ptree_fwd.hpp
ptree_serialization.hpp
stream_translator.hpp
string_path.hpp
xml_parser.hpp
adjacent_filtered.hpp
argument_fwd.hpp
copied.hpp
define_adaptor.hpp
filtered.hpp
indexed.hpp
indirected.hpp
map.hpp
replaced.hpp
replaced_if.hpp
reversed.hpp
sliced.hpp
strided.hpp

tokenized.hpp
transformed.hpp
uniqued.hpp
adaptors.hpp
adjacent_find.hpp
binary_search.hpp
copy.hpp
copy_backward.hpp
count.hpp
count_if.hpp
equal.hpp
equal_range.hpp
fill.hpp
fill_n.hpp
find.hpp
find_end.hpp
find_first_of.hpp
find_if.hpp
for_each.hpp
generate.hpp
heap_algorithm.hpp
inplace_merge.hpp
lexicographical_compare.hpp
lower_bound.hpp
max_element.hpp
merge.hpp
min_element.hpp
mismatch.hpp
nth_element.hpp
partial_sort.hpp
partial_sort_copy.hpp
partition.hpp
permutation.hpp
random_shuffle.hpp
remove.hpp
remove_copy.hpp
remove_copy_if.hpp
remove_if.hpp
replace.hpp
replace_copy.hpp
replace_copy_if.hpp
replace_if.hpp
reverse.hpp

reverse_copy.hpp
rotate.hpp
rotate_copy.hpp
search.hpp
search_n.hpp
set_algorithm.hpp
sort.hpp
stable_partition.hpp
stable_sort.hpp
swap_ranges.hpp
transform.hpp
unique.hpp
unique_copy.hpp
upper_bound.hpp
algorithm.hpp
copy_n.hpp
erase.hpp
for_each.hpp
insert.hpp
iota.hpp
is_sorted.hpp
overwrite.hpp
push_back.hpp
push_front.hpp
algorithm_ext.hpp
as_array.hpp
as_literal.hpp
atl.hpp
begin.hpp
category.hpp
combine.hpp
concepts.hpp
config.hpp
const_iterator.hpp
const_reverse_iterator.hpp
counting_range.hpp
as_literal.hpp
begin.hpp
collection_traits.hpp
collection_traits_detail.hpp
common.hpp
const_iterator.hpp
demote_iterator_traversal_tag.hpp

detail_str.hpp
difference_type.hpp
empty.hpp
end.hpp
extract_optional_type.hpp
implementation_help.hpp
iterator.hpp
join_iterator.hpp
microsoft.hpp
misc_concept.hpp
range_return.hpp
remove_extent.hpp
sfnaf.hpp
size.hpp
sizer.hpp
size_type.hpp
str_types.hpp
value_type.hpp
end.hpp
size.hpp
difference_type.hpp
distance.hpp
empty.hpp
end.hpp
functions.hpp
irange.hpp
istream_range.hpp
iterator.hpp
iterator_range.hpp
iterator_range_core.hpp
iterator_range_io.hpp
join.hpp
metafunctions.hpp
mfc.hpp
mutable_iterator.hpp
numeric.hpp
pointer.hpp
rbegin.hpp
reference.hpp
rend.hpp
result_iterator.hpp
reverse_iterator.hpp
reverse_result_iterator.hpp

size.hpp
size_type.hpp
sub_range.hpp
value_type.hpp
ref.hpp
concepts.hpp
borland.hpp
cwchar.hpp
config.hpp
icu.hpp
mfc.hpp
pattern_except.hpp
object_cache.hpp
static_mutex.hpp
unicode_iterator.hpp
regex_traits.hpp
user.hpp
basic_regex.hpp
basic_regex_creator.hpp
basic_regex_parser.hpp
char_regex_traits.hpp
cpp_regex_traits.hpp
cregex.hpp
c_regex_traits.hpp
error_type.hpp
fileiter.hpp
instances.hpp
iterator_category.hpp
iterator_traits.hpp
match_flags.hpp
match_results.hpp
mem_block_cache.hpp
perl_matcher.hpp
perl_matcher_common.hpp
perl_matcher_non_recursive.hpp
perl_matcher_recursive.hpp
primary_transform.hpp
protected_call.hpp
regbase.hpp
regex.hpp
regex_format.hpp
regex_fwd.hpp
regex_grep.hpp

regex_iterator.hpp
regex_match.hpp
regex_merge.hpp
regex_raw_buffer.hpp
regex_replace.hpp
regex_search.hpp
regex_split.hpp
regex_token_iterator.hpp
regex_traits.hpp
regex_traits_defaults.hpp
regex_workaround.hpp
states.hpp
sub_match.hpp
syntax_type.hpp
u32regex_iterator.hpp
u32regex_token_iterator.hpp
w32_regex_traits.hpp
regex.hpp
regex_fwd.hpp
scoped_array.hpp
scoped_ptr.hpp
scope_exit.hpp
access.hpp
array.hpp
assume_abstract.hpp
base_object.hpp
binary_object.hpp
bitset.hpp
collections_load_imp.hpp
collections_save_imp.hpp
collection_size_type.hpp
collection_traits.hpp
complex.hpp
config.hpp
deque.hpp
get_data.hpp
shared_count_132.hpp
shared_ptr_132.hpp
shared_ptr_nmt_132.hpp
stack_constructor.hpp
ephemeral.hpp
export.hpp
extended_type_info.hpp

extended_type_info_no_rtti.hpp
extended_type_info_typeid.hpp
factory.hpp
force_include.hpp
hash_collections_load_imp.hpp
hash_collections_save_imp.hpp
hash_map.hpp
hash_set.hpp
is_bitwise_serializable.hpp
item_version_type.hpp
level.hpp
level_enum.hpp
list.hpp
map.hpp
nvp.hpp
optional.hpp
pfto.hpp
scoped_ptr.hpp
serialization.hpp
set.hpp
shared_ptr.hpp
shared_ptr_132.hpp
singleton.hpp
slist.hpp
smart_cast.hpp
split_free.hpp
split_member.hpp
state_saver.hpp
static_warning.hpp
string.hpp
strong_typedef.hpp
throw_exception.hpp
tracking.hpp
tracking_enum.hpp
traits.hpp
type_info_implementation.hpp
utility.hpp
valarray.hpp
variant.hpp
vector.hpp
vector_135.hpp
version.hpp
void_cast.hpp

void_cast_fwd.hpp
weak_ptr.hpp
wrapper.hpp
shared_array.hpp
shared_ptr.hpp
bad_weak_ptr.hpp
atomic_count.hpp
atomic_count_gcc.hpp
atomic_count_gcc_x86.hpp
atomic_count_pthreads.hpp
atomic_count_solaris.hpp
atomic_count_sync.hpp
atomic_count_win32.hpp
lightweight_mutex.hpp
lwm_nop.hpp
lwm_pthreads.hpp
lwm_win32_cs.hpp
operator_bool.hpp
quick_allocator.hpp
shared_array_nmt.hpp
shared_count.hpp
shared_ptr_nmt.hpp
spinlock.hpp
spinlock_gcc_arm.hpp
spinlock_nt.hpp
spinlock_pool.hpp
spinlock_pt.hpp
spinlock_sync.hpp
spinlock_w32.hpp
sp_convertible.hpp
sp_counted_base.hpp
sp_counted_base_acc_ia64.hpp
sp_counted_base_cw_ppc.hpp
sp_counted_base_cw_x86.hpp
sp_counted_base_gcc_ia64.hpp
sp_counted_base_gcc_mips.hpp
sp_counted_base_gcc_ppc.hpp
sp_counted_base_gcc_sparc.hpp
sp_counted_base_gcc_x86.hpp
sp_counted_base_nt.hpp
sp_counted_base_pt.hpp
sp_counted_base_solaris.hpp
sp_counted_base_spin.hpp

sp_counted_base_sync.hpp
sp_counted_base_w32.hpp
sp_counted_impl.hpp
sp_has_sync.hpp
yield_k.hpp
enable_shared_from_this.hpp
enable_shared_from_this2.hpp
intrusive_ptr.hpp
make_shared.hpp
scoped_array.hpp
scoped_ptr.hpp
shared_array.hpp
shared_ptr.hpp
weak_ptr.hpp
smart_ptr.hpp
assign_actor.hpp
assign_key_actor.hpp
clear_actor.hpp
decrement_actor.hpp
erase_actor.hpp
increment_actor.hpp
insert_at_actor.hpp
insert_key_actor.hpp
push_back_actor.hpp
push_front_actor.hpp
ref_actor.hpp
ref_const_ref_actor.hpp
ref_const_ref_const_ref_a.hpp
ref_const_ref_value_actor.hpp
ref_value_actor.hpp
swap_actor.hpp
typeof.hpp
actor.hpp
closure.hpp
closure_context.hpp
closure_fwd.hpp
parametric.hpp
typeof.hpp
attribute.hpp
assert.hpp
actions.hpp
alternative.hpp
composite.hpp

difference.hpp
directives.hpp
epsilon.hpp
exclusive_or.hpp
intersection.hpp
kleene_star.hpp
list.hpp
no_actions.hpp
operators.hpp
optional.hpp
positive.hpp
sequence.hpp
sequential_and.hpp
sequential_or.hpp
config.hpp
match.hpp
nil.hpp
grammar.hpp
parser_context.hpp
parser_id.hpp
rule.hpp
subrule.hpp
subrule_fwd.hpp
parser.hpp
numerics.hpp
numerics_fwd.hpp
primitives.hpp
safe_bool.hpp
scanner.hpp
scanner_fwd.hpp
skipper.hpp
skipper_fwd.hpp
typeof.hpp
core.hpp
debug_node.hpp
minimal.hpp
parser_names.hpp
typeof.hpp
debug.hpp
for.hpp
if.hpp
lazy.hpp
rule_alias.hpp

select.hpp
stored_rule.hpp
stored_rule_fwd.hpp
switch.hpp
typeof.hpp
while.hpp
dynamic.hpp
exceptions.hpp
exceptions_fwd.hpp
typeof.hpp
error_handling.hpp
assign_actor.hpp
assign_key_actor.hpp
clear_actor.hpp
decrement_actor.hpp
erase_actor.hpp
increment_actor.hpp
insert_at_actor.hpp
insert_key_actor.hpp
push_back_actor.hpp
push_front_actor.hpp
ref_actor.hpp
ref_const_ref_actor.hpp
ref_const_ref_const_ref_a.hpp
ref_const_ref_value_actor.hpp
ref_value_actor.hpp
swap_actor.hpp
typeof.hpp
actor.hpp
closure.hpp
closure_context.hpp
closure_fwd.hpp
parametric.hpp
typeof.hpp
attribute.hpp
assert.hpp
actions.hpp
alternative.hpp
composite.hpp
difference.hpp
directives.hpp
epsilon.hpp
exclusive_or.hpp

alternative.ipp
difference.ipp
directives.ipp
exclusive_or.ipp
intersection.ipp
kleene_star.ipp
list.ipp
optional.ipp
positive.ipp
sequence.ipp
sequential_and.ipp
sequential_or.ipp
intersection.hpp
kleene_star.hpp
list.hpp
no_actions.hpp
operators.hpp
optional.hpp
positive.hpp
sequence.hpp
sequential_and.hpp
sequential_or.hpp
config.hpp
match.ipp
match_attr_traits.ipp
parser.ipp
match.hpp
nil.hpp
grammar.hpp
grammar.ipp
object_with_id.ipp
rule.ipp
static.hpp
subrule.ipp
parser_context.hpp
parser_id.hpp
rule.hpp
subrule.hpp
subrule_fwd.hpp
parser.hpp
numerics.ipp
primitives.ipp
numerics.hpp

numerics_fwd.hpp
primitives.hpp
safe_bool.hpp
skipper.ipp
scanner.hpp
scanner_fwd.hpp
skipper.hpp
skipper_fwd.hpp
typeof.hpp
core.hpp
debug_node.hpp
parser_names.ipp
minimal.hpp
parser_names.hpp
typeof.hpp
debug.hpp
for.hpp
if.hpp
conditions.ipp
select.ipp
switch.ipp
lazy.hpp
rule_alias.hpp
select.hpp
stored_rule.hpp
stored_rule_fwd.hpp
switch.hpp
typeof.hpp
while.hpp
dynamic.hpp
exceptions.hpp
exceptions_fwd.hpp
exceptions.ipp
typeof.hpp
error_handling.hpp
file_iterator.hpp
file_iterator_fwd.hpp
fixed_size_queue.hpp
file_iterator.ipp
position_iterator.ipp
multi_pass.hpp
multi_pass_fwd.hpp
position_iterator.hpp

position_iterator_fwd.hpp
typeof.hpp
iterator.hpp
as_parser.hpp
fundamental.hpp
fundamental.hpp
parser_traits.hpp
refactoring.hpp
traverse.hpp
parser_traits.hpp
refactoring.hpp
traverse.hpp
meta.hpp
namespace.hpp
actor.hpp
binders.hpp
casts.hpp
closures.hpp
composite.hpp
functions.hpp
new.hpp
operators.hpp
primitives.hpp
special_ops.hpp
statements.hpp
tuples.hpp
tuple_helpers.hpp
phoenix.hpp
symbols.hpp
tst.hpp
symbols.hpp
symbols_fwd.hpp
typeof.hpp
symbols.hpp
ast.hpp
ast_fwd.hpp
common.hpp
common_fwd.hpp
parse_tree_utils.hpp
tree_to_xml.hpp
parsetree.dtd
parse_tree.hpp
parse_tree_fwd.hpp

parse_tree_utils.hpp
tree_to_xml.hpp
typeof.hpp
chset.hpp
chset_operators.hpp
confix.hpp
confix_fwd.hpp
distinct.hpp
distinct_fwd.hpp
escape_char.hpp
escape_char_fwd.hpp
flush_multi_pass.hpp
functor_parser.hpp
grammar_def.hpp
grammar_def_fwd.hpp
basic_chset.hpp
basic_chset.hpp
range_run.hpp
range_run.hpp
chset.hpp
chset_operators.hpp
confix.hpp
escape_char.hpp
lists.hpp
regex.hpp
lists.hpp
lists_fwd.hpp
loops.hpp
regex.hpp
rule_parser.hpp
scoped_lock.hpp
typeof.hpp
utility.hpp
version.hpp
classic.hpp
action.hpp
action.hpp
auto.hpp
create_generator.hpp
meta_create.hpp
auto.hpp
attr_cast.hpp
eol.hpp

eps.hpp
lazy.hpp
auxiliary.hpp
binary.hpp
padding.hpp
binary.hpp
char.hpp
char_class.hpp
char_generator.hpp
char.hpp
delimit_flag.hpp
delimit_out.hpp
alternative_function.hpp
default_width.hpp
extract_from.hpp
fail_function.hpp
generate.hpp
generate_auto.hpp
generate_to.hpp
get_casetag.hpp
output_iterator.hpp
pass_container.hpp
string_compare.hpp
string_generate.hpp
unused_delimiter.hpp
buffer.hpp
center_alignment.hpp
columns.hpp
delimit.hpp
left_alignment.hpp
maxwidth.hpp
omit.hpp
repeat.hpp
right_alignment.hpp
upper_lower_case.hpp
verbatim.hpp
directive.hpp
domain.hpp
format.hpp
format_auto.hpp
generate.hpp
generate_attr.hpp
generator.hpp

meta_compiler.hpp
fcall.hpp
generator_binder.hpp
parameterized.hpp
grammar.hpp
rule.hpp
nonterminal.hpp
bool.hpp
bool_policies.hpp
bool_utils.hpp
numeric_utils.hpp
real_utils.hpp
int.hpp
real.hpp
real_policies.hpp
uint.hpp
numeric.hpp
alternative.hpp
and_predicate.hpp
kleene.hpp
list.hpp
not_predicate.hpp
optional.hpp
plus.hpp
sequence.hpp
operator.hpp
phoenix_attributes.hpp
reference.hpp
format_manip.hpp
format_manip_auto.hpp
iterator_sink.hpp
format_manip.hpp
format_manip_attr.hpp
ostream_iterator.hpp
stream.hpp
stream.hpp
lit.hpp
symbols.hpp
string.hpp
what.hpp
karma.hpp
argument.hpp
sequence_function.hpp

domain.hpp
action.hpp
char_token_def.hpp
lexer.hpp
functor.hpp
functor_data.hpp
generate_static.hpp
iterator.hpp
iterator_tokenizer.hpp
lexer.hpp
semantic_action_data.hpp
static_functor_data.hpp
static_lexer.hpp
static_version.hpp
token.hpp
wrap_action.hpp
pass_flags.hpp
sequence.hpp
string_token_def.hpp
support_functions.hpp
terminals.hpp
token_def.hpp
lexer.hpp
lexer_lexertl.hpp
lexer_static_lexertl.hpp
lexer_type.hpp
meta_compiler.hpp
primitives.hpp
in_state.hpp
plain_token.hpp
state_switcher.hpp
qi.hpp
reference.hpp
tokenize_and_parse.hpp
tokenize_and_parse_attr.hpp
lex.hpp
algorithm.hpp
bind_function.hpp
bind_function_object.hpp
bind_member_function.hpp
bind_member_variable.hpp
bind_function.hpp
bind_function_object.hpp

bind_member_function.hpp
function_ptr.hpp
member_function_ptr.hpp
bind.hpp
container.hpp
actor.hpp
argument.hpp
as_actor.hpp
basic_environment.hpp
compose.hpp
composite.hpp
actor.hpp
basic_environment.hpp
compose.hpp
composite.hpp
composite_eval.hpp
composite_info.hpp
function_eval.hpp
is_actor.hpp
limits.hpp
nothing.hpp
reference.hpp
value.hpp
core.hpp
local_reference.hpp
type_deduction.hpp
function_call.hpp
function.hpp
function.hpp
at.hpp
fusion.hpp
construct.hpp
const_cast.hpp
delete.hpp
construct.hpp
construct_eval.hpp
new.hpp
new_eval.hpp
dynamic_cast.hpp
new.hpp
reinterpret_cast.hpp
static_cast.hpp
object.hpp

arithmetic.hpp
bitwise.hpp
comparison.hpp
binary_compose.hpp
binary_eval.hpp
io.hpp
mem_fun_ptr_eval.hpp
mem_fun_ptr_gen.hpp
mem_fun_ptr_return.hpp
unary_compose.hpp
unary_eval.hpp
if_else.hpp
io.hpp
logical.hpp
member.hpp
self.hpp
operator.hpp
local_gen.hpp
local_variable.hpp
dynamic.hpp
lambda.hpp
let.hpp
local_variable.hpp
scoped_environment.hpp
scope.hpp
catch_all_eval.hpp
catch_composite.hpp
catch_eval.hpp
switch.hpp
switch_eval.hpp
switch_eval.hpp
do_while.hpp
for.hpp
if.hpp
sequence.hpp
switch.hpp
throw.hpp
try_catch.hpp
while.hpp
statement.hpp
begin.hpp
decay_array.hpp
end.hpp

has_equal_range.hpp
has_find.hpp
has_lower_bound.hpp
has_remove.hpp
has_remove_if.hpp
has_reverse.hpp
has_sort.hpp
has_unique.hpp
has_upper_bound.hpp
is_std_hash_map.hpp
is_std_hash_set.hpp
is_std_list.hpp
is_std_map.hpp
is_std_set.hpp
std_hash_map_fwd.hpp
std_hash_set_fwd.hpp
iteration.hpp
querying.hpp
transformation.hpp
algorithm.hpp
container.hpp
container.hpp
container.hpp
stl.hpp
version.hpp
phoenix.hpp
action.hpp
action.hpp
auto.hpp
create_parser.hpp
meta_create.hpp
auto.hpp
attr.hpp
attr_cast.hpp
eoi.hpp
eol.hpp
eps.hpp
lazy.hpp
auxiliary.hpp
binary.hpp
binary.hpp
char.hpp
char_class.hpp

char_parser.hpp
char.hpp
alternative_function.hpp
assign_to.hpp
construct.hpp
expect_function.hpp
fail_function.hpp
parse.hpp
parse_auto.hpp
pass_container.hpp
pass_function.hpp
permute_function.hpp
string_parse.hpp
unused_skipper.hpp
lexeme.hpp
matches.hpp
no_case.hpp
omit.hpp
raw.hpp
repeat.hpp
skip.hpp
directive.hpp
domain.hpp
match.hpp
match_auto.hpp
meta_compiler.hpp
debug_handler.hpp
fcall.hpp
parameterized.hpp
parser_binder.hpp
error_handler.hpp
grammar.hpp
rule.hpp
simple_trace.hpp
nonterminal.hpp
bool.hpp
bool_policies.hpp
numeric_utils.hpp
real_impl.hpp
int.hpp
numeric_utils.hpp
real.hpp
real_policies.hpp

uint.hpp
numeric.hpp
alternative.hpp
and_predicate.hpp
difference.hpp
expect.hpp
kleene.hpp
list.hpp
not_predicate.hpp
optional.hpp
permutation.hpp
plus.hpp
sequence.hpp
sequence_base.hpp
sequential_or.hpp
operator.hpp
parse.hpp
parser.hpp
parse_attr.hpp
reference.hpp
skip_flag.hpp
skip_over.hpp
iterator_source.hpp
match_manip.hpp
match_manip_auto.hpp
match_manip.hpp
match_manip_attr.hpp
stream.hpp
stream.hpp
tst.hpp
lit.hpp
symbols.hpp
tst.hpp
tst_map.hpp
string.hpp
what.hpp
qi.hpp
action_dispatch.hpp
any.hpp
any_if.hpp
any_if_ns.hpp
any_ns.hpp
argument.hpp

assert_msg.hpp
attributes.hpp
attributes_fwd.hpp
meta_create.hpp
attr_cast.hpp
char_class.hpp
ascii.hpp
iso8859_1.hpp
standard.hpp
standard_wide.hpp
basic_chset.hpp
range.hpp
range_functions.hpp
range_run.hpp
range_run_impl.hpp
common_terminals.hpp
container.hpp
context.hpp
as_variant.hpp
endian.hpp
get_encoding.hpp
hold_any.hpp
cover_operators.hpp
endian.hpp
char_traits.hpp
consts.hpp
ptr_list.hpp
ptr_vector.hpp
char_state_machine.hpp
debug.hpp
file_input.hpp
generate_cpp.hpp
generate_re2c.hpp
generator.hpp
input.hpp
internals.hpp
parser.hpp
num_token.hpp
re_tokeniser.hpp
re_tokeniser_helper.hpp
re_tokeniser_state.hpp
end_node.hpp
iteration_node.hpp

leaf_node.hpp
node.hpp
selection_node.hpp
sequence_node.hpp
charset.hpp
equivset.hpp
rules.hpp
runtime_error.hpp
serialise.hpp
size_t.hpp
state_machine.hpp
string_token.hpp
make_cons.hpp
make_vector.hpp
fp_traits.hpp
fpclassify.hpp
signbit.hpp
pow10.hpp
scoped_enum_emulation.hpp
sign.hpp
what_function.hpp
has_semantic_action.hpp
info.hpp
buf_id_check_policy.hpp
combine_policies.hpp
first_owner_policy.hpp
fixed_size_queue.hpp
fixed_size_queue_policy.hpp
functor_input_policy.hpp
input_iterator_policy.hpp
istream_policy.hpp
lex_input_policy.hpp
multi_pass.hpp
no_check_policy.hpp
ref_counted_policy.hpp
split_functor_input_policy.hpp
split_std_deque_policy.hpp
istream_iterator.hpp
look_ahead.hpp
multi_pass.hpp
multi_pass_fwd.hpp
lazy.hpp
make_component.hpp

meta_compiler.hpp
modify.hpp
multi_pass.hpp
multi_pass_wrapper.hpp
expand_arg.hpp
extract_param.hpp
locals.hpp
sequence_base_id.hpp
string_traits.hpp
terminal.hpp
unused.hpp
utf8.hpp
support.hpp
classic.hpp
classic_actions.hpp
classic_actor.hpp
classic_alternative.hpp
classic_assert.hpp
classic_assign_actor.hpp
classic_assign_key_actor.hpp
classic_ast.hpp
classic_ast_fwd.hpp
classic_as_parser.hpp
classic_attribute.hpp
classic_basic_chset.hpp
classic_chset.hpp
classic_chset_operators.hpp
classic_clear_actor.hpp
classic_closure.hpp
classic_closure_context.hpp
classic_closure_fwd.hpp
classic_common.hpp
classic_common_fwd.hpp
classic_composite.hpp
classic_config.hpp
classic_confix.hpp
classic_confix_fwd.hpp
classic_core.hpp
classic_debug.hpp
classic_debug_node.hpp
classic_decrement_actor.hpp
classic_difference.hpp
classic_directives.hpp

classic_distinct.hpp
classic_distinct_fwd.hpp
classic_dynamic.hpp
classic_epsilon.hpp
classic_erase_actor.hpp
classic_error_handling.hpp
classic_escape_char.hpp
classic_escape_char_fwd.hpp
classic_exceptions.hpp
classic_exceptions_fwd.hpp
classic_exclusive_or.hpp
classic_file_iterator.hpp
classic_file_iterator_fwd.hpp
classic_fixed_size_queue.hpp
classic_flush_multi_pass.hpp
classic_for.hpp
classic_functor_parser.hpp
classic_fundamental.hpp
classic_grammar.hpp
classic_grammar_def.hpp
classic_grammar_def_fwd.hpp
classic_if.hpp
classic_increment_actor.hpp
classic_insert_at_actor.hpp
classic_insert_key_actor.hpp
classic_intersection.hpp
classic_iterator.hpp
classic_kleene_star.hpp
classic_lazy.hpp
classic_list.hpp
classic_lists.hpp
classic_lists_fwd.hpp
classic_loops.hpp
classic_match.hpp
classic_meta.hpp
classic_minimal.hpp
classic_multi_pass.hpp
classic_multi_pass_fwd.hpp
classic_nil.hpp
classic_no_actions.hpp
classic_numerics.hpp
classic_numerics_fwd.hpp
classic_operators.hpp

classic_optional.hpp
classic_parametric.hpp
classic_parser.hpp
classic_parser_context.hpp
classic_parser_id.hpp
classic_parser_names.hpp
classic_parser_traits.hpp
classic_parse_tree.hpp
classic_parse_tree_fwd.hpp
classic_parse_tree_utils.hpp
classic_position_iterator.hpp
classic_position_iterator_fwd.hpp
classic_positive.hpp
classic_primitives.hpp
classic_push_back_actor.hpp
classic_push_front_actor.hpp
classic_range_run.hpp
classic_refactoring.hpp
classic_ref_actor.hpp
classic_ref_const_ref_actor.hpp
classic_ref_const_ref_const_ref_a.hpp
classic_ref_const_ref_value_actor.hpp
classic_ref_value_actor.hpp
classic_regex.hpp
classic_rule.hpp
classic_rule_alias.hpp
classic_rule_parser.hpp
classic_safe_bool.hpp
classic_scanner.hpp
classic_scanner_fwd.hpp
classic_scoped_lock.hpp
classic_select.hpp
classic_sequence.hpp
classic_sequential_and.hpp
classic_sequential_or.hpp
classic_skipper.hpp
classic_skipper_fwd.hpp
classic_spirit.hpp
classic_static.hpp
classic_stored_rule.hpp
classic_stored_rule_fwd.hpp
classic_subrule.hpp
classic_subrule_fwd.hpp

classic_swap_actor.hpp
classic_switch.hpp
classic_symbols.hpp
classic_symbols_fwd.hpp
classic_traverse.hpp
classic_tree_to_xml.hpp
classic_typeof.hpp
classic_utility.hpp
classic_version.hpp
classic_while.hpp
karma.hpp
karma_action.hpp
karma_alternative.hpp
karma_and_predicate.hpp
karma_attr_cast.hpp
karma_auto.hpp
karma_auxiliary.hpp
karma_binary.hpp
karma_bool.hpp
karma_buffer.hpp
karma_center_alignment.hpp
karma_char.hpp
karma_char_.hpp
karma_char_class.hpp
karma_columns.hpp
karma_delimit.hpp
karma_directive.hpp
karma_domain.hpp
karma_eol.hpp
karma_eps.hpp
karma_format.hpp
karma_format_attr.hpp
karma_format_auto.hpp
karma_generate.hpp
karma_generate_attr.hpp
karma_generate_auto.hpp
karma_grammar.hpp
karma_int.hpp
karma_kleene.hpp
karma_lazy.hpp
karma_left_alignment.hpp
karma_list.hpp
karma_maxwidth.hpp

karma_nonterminal.hpp
karma_not_predicate.hpp
karma_numeric.hpp
karma_omit.hpp
karma_operator.hpp
karma_optional.hpp
karma_phoenix_attributes.hpp
karma_plus.hpp
karma_real.hpp
karma_repeat.hpp
karma_right_alignment.hpp
karma_rule.hpp
karma_sequence.hpp
karma_stream.hpp
karma_string.hpp
karma_symbols.hpp
karma_uint.hpp
karma_upper_lower_case.hpp
karma_verbatim.hpp
karma_what.hpp
lex.hpp
lex_char_token_def.hpp
lex_domain.hpp
lex_generate_static_lexertl.hpp
lex_lexer.hpp
lex_lexertl.hpp
lex_primitives.hpp
lex_static_lexertl.hpp
lex_tokenize_and_parse.hpp
lex_tokenize_and_parse_attr.hpp
phoenix.hpp
phoenix1.hpp
phoenix1_actor.hpp
phoenix1_binders.hpp
phoenix1_casts.hpp
phoenix1_closures.hpp
phoenix1_composite.hpp
phoenix1_functions.hpp
phoenix1_new.hpp
phoenix1_operators.hpp
phoenix1_primitives.hpp
phoenix1_special_ops.hpp
phoenix1_statements.hpp

phoenix1_tuples.hpp
phoenix1_tuple_helpers.hpp
phoenix_algorithm.hpp
phoenix_bind.hpp
phoenix_container.hpp
phoenix_core.hpp
phoenix_function.hpp
phoenix_fusion.hpp
phoenix_object.hpp
phoenix_operator.hpp
phoenix_scope.hpp
phoenix_statement.hpp
phoenix_stl.hpp
phoenix_version.hpp
qi.hpp
qi_action.hpp
qi_alternative.hpp
qi_and_predicate.hpp
qi_attr.hpp
qi_attr_cast.hpp
qi_auto.hpp
qi_auxiliary.hpp
qi_binary.hpp
qi_bool.hpp
qi_char.hpp
qi_char_.hpp
qi_char_class.hpp
qi_core.hpp
qi_difference.hpp
qi_directive.hpp
qi_domain.hpp
qi_eoi.hpp
qi_eol.hpp
qi_eps.hpp
qi_expect.hpp
qi_grammar.hpp
qi_int.hpp
qi_kleene.hpp
qi_lazy.hpp
qi_lexeme.hpp
qi_list.hpp
qi_lit.hpp
qi_match.hpp

qi_matches.hpp
qi_match_attr.hpp
qi_match_auto.hpp
qi_nonterminal.hpp
qi_not_predicate.hpp
qi_no_case.hpp
qi_numeric.hpp
qi_omit.hpp
qi_operator.hpp
qi_optional.hpp
qi_parse.hpp
qi_parse_attr.hpp
qi_parse_auto.hpp
qi_permutation.hpp
qi_plus.hpp
qi_raw.hpp
qi_real.hpp
qi_repeat.hpp
qi_rule.hpp
qi_sequence.hpp
qi_sequential_or.hpp
qi_skip.hpp
qi_stream.hpp
qi_string.hpp
qi_symbols.hpp
qi_uint.hpp
qi_what.hpp
support.hpp
support_any.hpp
support_any_if.hpp
support_any_if_ns.hpp
support_any_ns.hpp
support_argument.hpp
support_ascii.hpp
support_attributes.hpp
support_char_class.hpp
support_iso8859_1.hpp
support_istream_iterator.hpp
support_locals.hpp
support_look_ahead.hpp
support_modify.hpp
support_multi_pass.hpp
support_multi_pass_fwd.hpp

support_standard.hpp
support_standard_wide.hpp
support_string_traits.hpp
support_unused.hpp
version.hpp
file_iterator.hpp
file_iterator_fwd.hpp
fixed_size_queue.hpp
multi_pass.hpp
multi_pass_fwd.hpp
position_iterator.hpp
position_iterator_fwd.hpp
typeof.hpp
iterator.hpp
as_parser.hpp
fundamental.hpp
parser_traits.hpp
refactoring.hpp
traverse.hpp
meta.hpp
actor.hpp
binders.hpp
casts.hpp
closures.hpp
composite.hpp
functions.hpp
new.hpp
operators.hpp
primitives.hpp
special_ops.hpp
statements.hpp
tuples.hpp
tuple_helpers.hpp
phoenix.hpp
confix.hpp
directive.hpp
subrule.hpp
nonterminal.hpp
karma.hpp
confix.hpp
distinct.hpp
directive.hpp
subrule.hpp

nonterminal.hpp
flush_multi_pass.hpp
iter_pos.hpp
primitive.hpp
qi.hpp
confix.hpp
distinct.hpp
flush_multi_pass.hpp
subrule_context.hpp
karma.hpp
karma_confix.hpp
karma_directive.hpp
karma_nonterminal.hpp
karma_subrule.hpp
qi.hpp
qi_confix.hpp
qi_directive.hpp
qi_distinct.hpp
qi_flush_multi_pass.hpp
qi_iter_pos.hpp
qi_nonterminal.hpp
qi_primitive.hpp
qi_subrule.hpp
symbols.hpp
symbols_fwd.hpp
typeof.hpp
symbols.hpp
ast.hpp
ast_fwd.hpp
common.hpp
common_fwd.hpp
parsetree.dtd
parse_tree.hpp
parse_tree_fwd.hpp
parse_tree_utils.hpp
tree_to_xml.hpp
typeof.hpp
chset.hpp
chset_operators.hpp
confix.hpp
confix_fwd.hpp
distinct.hpp
distinct_fwd.hpp

escape_char.hpp
escape_char_fwd.hpp
flush_multi_pass.hpp
functor_parser.hpp
grammar_def.hpp
grammar_def_fwd.hpp
lists.hpp
lists_fwd.hpp
loops.hpp
regex.hpp
rule_parser.hpp
scoped_lock.hpp
typeof.hpp
utility.hpp
version.hpp
static_assert.hpp
swap.hpp
api_config.hpp
config.hpp
cygwin_error.hpp
error_code.hpp
linux_error.hpp
system_error.hpp
windows_error.hpp
barrier.hpp
condition.hpp
condition_variable.hpp
config.hpp
force_cast.hpp
move.hpp
platform.hpp
singleton.hpp
thread.hpp
thread_group.hpp
thread_heap_alloc.hpp
thread_interruption.hpp
tss_hooks.hpp
exceptions.hpp
future.hpp
locks.hpp
mutex.hpp
once.hpp
condition_variable.hpp

condition_variable_fwd.hpp
mutex.hpp
once.hpp
pthread_mutex_scoped_lock.hpp
recursive_mutex.hpp
shared_mutex.hpp
thread_data.hpp
thread_heap_alloc.hpp
timespec.hpp
recursive_mutex.hpp
shared_mutex.hpp
thread.hpp
thread_time.hpp
tss.hpp
basic_recursive_mutex.hpp
basic_timed_mutex.hpp
condition_variable.hpp
interlocked_read.hpp
mutex.hpp
once.hpp
recursive_mutex.hpp
shared_mutex.hpp
thread_data.hpp
thread_heap_alloc.hpp
thread_primitives.hpp
xtime.hpp
thread.hpp
future.hpp
locking_ptr.hpp
pool_core.hpp
scope_guard.hpp
worker_thread.hpp
future.hpp
pool.hpp
pool_adaptors.hpp
scheduling_policies.hpp
shutdown_policies.hpp
size_policies.hpp
task_adaptors.hpp
threadpool.hpp
throw_exception.hpp
timer.hpp
tokenizer.hpp

token_functions.hpp
token_iterator.hpp
tuple_basic.hpp
tuple_basic_no_partial_spec.hpp
tuple.hpp
tuple_comparison.hpp
tuple_io.hpp
type.hpp
typeof_impl.hpp
encode_decode.hpp
encode_decode_params.hpp
incr_registration_group.hpp
integral_template_param.hpp
int_encoding.hpp
message.hpp
modifiers.hpp
typeof_impl.hpp
native.hpp
pointers_data_members.hpp
register_functions.hpp
register_functions_iterate.hpp
register_fundamental.hpp
register_mem_functions.hpp
bitset.hpp
complex.hpp
deque.hpp
fstream.hpp
functional.hpp
iostream.hpp
istream.hpp
iterator.hpp
list.hpp
locale.hpp
map.hpp
memory.hpp
ostream.hpp
queue.hpp
set.hpp
sstream.hpp
stack.hpp
streambuf.hpp
string.hpp
utility.hpp

valarray.hpp
vector.hpp
template_encoding.hpp
template_template_param.hpp
typeof.hpp
typeof_impl.hpp
type_encoding.hpp
type_template_param.hpp
vector.hpp
vector100.hpp
vector150.hpp
vector200.hpp
vector50.hpp
add_const.hpp
add_cv.hpp
add_pointer.hpp
add_reference.hpp
add_volatile.hpp
aligned_storage.hpp
alignment_of.hpp
alignment_traits.hpp
arithmetic_traits.hpp
array_traits.hpp
broken_compiler_spec.hpp
composite_traits.hpp
config.hpp
conversion_traits.hpp
cv_traits.hpp
decay.hpp
bool_trait_def.hpp
bool_trait_undef.hpp
cv_traits_impl.hpp
false_result.hpp
ice_and.hpp
ice_eq.hpp
ice_not.hpp
ice_or.hpp
is_function_ptr_helper.hpp
is_function_ptr_tester.hpp
is_mem_fun_pointer_impl.hpp
is_mem_fun_pointer_tester.hpp
size_t_trait_def.hpp
size_t_trait_undef.hpp

template_arity_spec.hpp
type_trait_def.hpp
type_trait_undef.hpp
wrap.hpp
yes_no_type.hpp
extent.hpp
floating_point_promotion.hpp
function_traits.hpp
has_new_operator.hpp
has_nothrow_assign.hpp
has_nothrow_constructor.hpp
has_nothrow_copy.hpp
has_nothrow_destructor.hpp
has_trivial_assign.hpp
has_trivial_constructor.hpp
has_trivial_copy.hpp
has_trivial_destructor.hpp
has_virtual_destructor.hpp
ice.hpp
integral_constant.hpp
integral_promotion.hpp
intrinsic.hpp
is_abstract.hpp
is_arithmetic.hpp
is_array.hpp
is_base_and_derived.hpp
is_base_of.hpp
is_base_of_tr1.hpp
is_class.hpp
is_complex.hpp
is_compound.hpp
is_const.hpp
is_convertible.hpp
is_empty.hpp
is_enum.hpp
is_float.hpp
is_floating_point.hpp
is_function.hpp
is_fundamental.hpp
is_integral.hpp
is_lvalue_reference.hpp
is_member_function_pointer.hpp
is_member_object_pointer.hpp

is_member_pointer.hpp
is_object.hpp
is_pod.hpp
is_pointer.hpp
is_polymorphic.hpp
is_reference.hpp
is_rvalue_reference.hpp
is_same.hpp
is_scalar.hpp
is_signed.hpp
is_stateless.hpp
is_union.hpp
is_unsigned.hpp
is_virtual_base_of.hpp
is_void.hpp
is_volatile.hpp
make_signed.hpp
make_unsigned.hpp
remove_all_extents.hpp
remove_bounds.hpp
remove_const.hpp
remove_cv.hpp
remove_extent.hpp
remove_pointer.hpp
remove_reference.hpp
remove_volatile.hpp
typeof.hpp
object_traits.hpp
promote.hpp
rank.hpp
reference_traits.hpp
remove_all_extents.hpp
remove_bounds.hpp
remove_const.hpp
remove_cv.hpp
remove_extent.hpp
remove_pointer.hpp
remove_reference.hpp
remove_volatile.hpp
same_traits.hpp
transform_traits.hpp
transform_traits_spec.hpp
type_with_alignment.hpp

type_traits.hpp
absolute.hpp
base_dimension.hpp
base_unit.hpp
arcminute.hpp
arcsecond.hpp
degree.hpp
gradian.hpp
radian.hpp
revolution.hpp
steradian.hpp
astronomical_unit.hpp
light_day.hpp
light_hour.hpp
light_minute.hpp
light_second.hpp
light_year.hpp
parsec.hpp
biot.hpp
centimeter.hpp
gram.hpp
conversions.hpp
drachm.hpp
fluid_ounce.hpp
foot.hpp
furlong.hpp
gallon.hpp
gill.hpp
grain.hpp
hundredweight.hpp
inch.hpp
league.hpp
mile.hpp
ounce.hpp
pint.hpp
pound.hpp
quart.hpp
quarter.hpp
stone.hpp
thou.hpp
ton.hpp
yard.hpp
angstrom.hpp

are.hpp
atmosphere.hpp
bar.hpp
barn.hpp
day.hpp
fermi.hpp
hectare.hpp
hour.hpp
knot.hpp
liter.hpp
micron.hpp
minute.hpp
mmHg.hpp
nautical_mile.hpp
ton.hpp
torr.hpp
year.hpp
ampere.hpp
candela.hpp
kelvin.hpp
kilogram.hpp
meter.hpp
mole.hpp
second.hpp
celsius.hpp
conversions.hpp
fahrenheit.hpp
cup.hpp
dram.hpp
fluid_dram.hpp
fluid_ounce.hpp
foot.hpp
gallon.hpp
gill.hpp
grain.hpp
hundredweight.hpp
inch.hpp
mil.hpp
mile.hpp
minim.hpp
ounce.hpp
pint.hpp
pound.hpp

pound_force.hpp
quart.hpp
tablespoon.hpp
teaspoon.hpp
ton.hpp
yard.hpp
cmath.hpp
config.hpp
conversion.hpp
derived_dimension.hpp
absolute_impl.hpp
cmath_impl.hpp
conversion_impl.hpp
dimensionless_unit.hpp
dimension_impl.hpp
dimension_list.hpp
dim_impl.hpp
heterogeneous_conversion.hpp
linear_algebra.hpp
one.hpp
ordinal.hpp
prevent_redefinition.hpp
push_front_if.hpp
push_front_or_add.hpp
sort.hpp
static_rational_power.hpp
unscale.hpp
utility.hpp
dim.hpp
dimension.hpp
dimensionless_quantity.hpp
dimensionless_type.hpp
dimensionless_unit.hpp
get_dimension.hpp
get_system.hpp
heterogeneous_system.hpp
homogeneous_system.hpp
io.hpp
is_dim.hpp
is_dimensionless.hpp
is_dimensionless_quantity.hpp
is_dimensionless_unit.hpp
is_dimension_list.hpp

is_quantity.hpp
is_quantity_of_dimension.hpp
is_quantity_of_system.hpp
is_unit.hpp
is_unit_of_dimension.hpp
is_unit_of_system.hpp
lambda.hpp
limits.hpp
make_scaled_unit.hpp
make_system.hpp
operators.hpp
absorbed_dose.hpp
acceleration.hpp
action.hpp
activity.hpp
amount.hpp
angular_acceleration.hpp
angular_momentum.hpp
angular_velocity.hpp
area.hpp
capacitance.hpp
conductance.hpp
conductivity.hpp
current.hpp
dose_equivalent.hpp
dynamic_viscosity.hpp
electric_charge.hpp
electric_potential.hpp
energy.hpp
energy_density.hpp
force.hpp
frequency.hpp
heat_capacity.hpp
illuminance.hpp
impedance.hpp
inductance.hpp
kinematic_viscosity.hpp
length.hpp
luminance.hpp
luminous_flux.hpp
luminous_intensity.hpp
magnetic_field_intensity.hpp
magnetic_flux.hpp

magnetic_flux_density.hpp
mass.hpp
mass_density.hpp
molar_energy.hpp
molar_heat_capacity.hpp
momentum.hpp
moment_of_inertia.hpp
permeability.hpp
permittivity.hpp
plane_angle.hpp
power.hpp
pressure.hpp
reluctance.hpp
resistance.hpp
resistivity.hpp
solid_angle.hpp
specific_energy.hpp
specific_heat_capacity.hpp
specific_volume.hpp
stress.hpp
surface_density.hpp
surface_tension.hpp
temperature.hpp
thermal_conductivity.hpp
time.hpp
torque.hpp
velocity.hpp
volume.hpp
wavenumber.hpp
physical_dimensions.hpp
pow.hpp
quantity.hpp
reduce_unit.hpp
scale.hpp
scaled_base_unit.hpp
static_constant.hpp
static_rational.hpp
abstract.hpp
degrees.hpp
gradians.hpp
revolutions.hpp
acceleration.hpp
area.hpp

base.hpp
current.hpp
dimensionless.hpp
dynamic_viscosity.hpp
energy.hpp
force.hpp
frequency.hpp
io.hpp
kinematic_viscosity.hpp
length.hpp
mass.hpp
mass_density.hpp
momentum.hpp
power.hpp
pressure.hpp
time.hpp
velocity.hpp
volume.hpp
wavenumber.hpp
cgs.hpp
constants.hpp
absorbed_dose.hpp
acceleration.hpp
action.hpp
activity.hpp
amount.hpp
angular_acceleration.hpp
angular_momentum.hpp
angular_velocity.hpp
area.hpp
base.hpp
capacitance.hpp
catalytic_activity.hpp
alpha_constants.hpp
atomic-nuclear_constants.hpp
deuteron_constants.hpp
electromagnetic_constants.hpp
electron_constants.hpp
helion_constants.hpp
muon_constants.hpp
neutron_constants.hpp
physico-chemical_constants.hpp
proton_constants.hpp

tau_constants.hpp
triton_constants.hpp
typedefs.hpp
universal_constants.hpp
codata_constants.hpp
conductance.hpp
conductivity.hpp
current.hpp
dimensionless.hpp
dose_equivalent.hpp
dynamic_viscosity.hpp
electric_charge.hpp
electric_potential.hpp
energy.hpp
force.hpp
frequency.hpp
illuminance.hpp
impedance.hpp
inductance.hpp
io.hpp
kinematic_viscosity.hpp
length.hpp
luminous_flux.hpp
luminous_intensity.hpp
magnetic_field_intensity.hpp
magnetic_flux.hpp
magnetic_flux_density.hpp
mass.hpp
mass_density.hpp
momentum.hpp
moment_of_inertia.hpp
permeability.hpp
permittivity.hpp
plane_angle.hpp
power.hpp
prefixes.hpp
pressure.hpp
reluctance.hpp
resistance.hpp
resistivity.hpp
solid_angle.hpp
surface_density.hpp
surface_tension.hpp

temperature.hpp
time.hpp
torque.hpp
velocity.hpp
volume.hpp
wavenumber.hpp
si.hpp
celsius.hpp
fahrenheit.hpp
unit.hpp
units_fwd.hpp
addressof.hpp
base_from_member.hpp
binary.hpp
compare_pointees.hpp
in_place_factory_prefix.hpp
in_place_factory_suffix.hpp
result_of_iterate.hpp
enable_if.hpp
in_place_factory.hpp
result_of.hpp
swap.hpp
typed_in_place_factory.hpp
value_init.hpp
utility.hpp
apply_visitor.hpp
bad_visit.hpp
apply_visitor_binary.hpp
apply_visitor_delayed.hpp
apply_visitor_unary.hpp
backup_holder.hpp
bool_trait_def.hpp
bool_trait_undef.hpp
cast_storage.hpp
config.hpp
enable_recursive.hpp
enable_recursive_fwd.hpp
forced_return.hpp
generic_result_type.hpp
has_nothrow_move.hpp
has_trivial_move.hpp
initializer.hpp
make_variant_list.hpp

move.hpp
over_sequence.hpp
substitute.hpp
substitute_fwd.hpp
variant_io.hpp
visitation_impl.hpp
get.hpp
recursive_variant.hpp
recursive_wrapper.hpp
recursive_wrapper_fwd.hpp
static_visitor.hpp
variant.hpp
variant_fwd.hpp
visitor_ptr.hpp
variant.hpp
version.hpp
visit_each.hpp
weak_ptr.hpp
libboost_serialization.a
libboost_system.a
libboost_thread.a
libboost_serialization.a
libboost_system.a
libboost_thread.a
libboost_serialization.a
libboost_system.a
libboost_thread.a
libboost_serialization.a
libboost_system.a
libboost_thread.a
archive_exception.hpp
abi_prefix.hpp
abi_suffix.hpp
decl.hpp
assignment_exception.hpp
list_inserter.hpp
list_of.hpp
ptr_list_inserter.hpp
ptr_list_of.hpp
ptr_map_inserter.hpp
deque.hpp
list.hpp
map.hpp

queue.hpp
set.hpp
slist.hpp
stack.hpp
vector.hpp
std.hpp
assign.hpp
base.hpp
debug.hpp
details.hpp
space_optimized.hpp
circular_buffer.hpp
circular_buffer_fwd.hpp
compressed_pair.hpp
functional.hpp
access_specifier.hpp
adl_swap.hpp
archive_constructed.hpp
auto_space.hpp
base_type.hpp
bidir_node_iterator.hpp
converter.hpp
copy_map.hpp
do_not_copy_elements_tag.hpp
duplicates_iterator.hpp
has_tag.hpp
header_holder.hpp
index_base.hpp
index_loader.hpp
index_matcher.hpp
index_node_base.hpp
index_saver.hpp
invariant_assert.hpp
is_index_list.hpp
iter_adaptor.hpp
modify_key_adaptor.hpp
msvc_index_specifier.hpp
node_type.hpp
no_duplicate_tags.hpp
ord_index_args.hpp
ord_index_node.hpp
ord_index_ops.hpp
prevent_eti.hpp

safe_ctr_proxy.hpp
safe_mode.hpp
scope_guard.hpp
seq_index_node.hpp
seq_index_ops.hpp
serialization_version.hpp
uintptr_type.hpp
unbounded.hpp
value_compare.hpp
vartempl_support.hpp
identity.hpp
identity_fwd.hpp
indexed_by.hpp
member.hpp
ordered_index.hpp
ordered_index_fwd.hpp
safe_mode_errors.hpp
sequenced_index.hpp
sequenced_index_fwd.hpp
tag.hpp
multi_index_container.hpp
multi_index_container_fwd.hpp
clone_allocator.hpp
associative_ptr_container.hpp
default_deleter.hpp
is_convertible.hpp
map_iterator.hpp
meta_functions.hpp
move.hpp
reversible_ptr_container.hpp
scoped_deleter.hpp
static_move_ptr.hpp
throw_exception.hpp
void_ptr_iterator.hpp
exception.hpp
indirect_fun.hpp
nullable.hpp
ptr_array.hpp
ptr_circular_buffer.hpp
ptr_container.hpp
ptr_deque.hpp
ptr_inserter.hpp
ptr_list.hpp

ptr_map.hpp
ptr_map_adapter.hpp
ptr_sequence_adapter.hpp
ptr_set.hpp
ptr_set_adapter.hpp
ptr_unordered_map.hpp
ptr_unordered_set.hpp
ptr_vector.hpp
access.hpp
base_object.hpp
collection_size_type.hpp
force_include.hpp
is_bitwise_serializable.hpp
level.hpp
level_enum.hpp
nvp.hpp
pfto.hpp
serialization.hpp
split_free.hpp
split_member.hpp
strong_typedef.hpp
tracking.hpp
tracking_enum.hpp
traits.hpp
type_info_implementation.hpp
version.hpp
void_cast_fwd.hpp
wrapper.hpp
allocate.hpp
buckets.hpp
equivalent.hpp
extract_key.hpp
fwd.hpp
table.hpp
unique.hpp
util.hpp
unordered_map.hpp
unordered_map_fwd.hpp
unordered_set.hpp
unordered_set_fwd.hpp
unordered_map.hpp
unordered_set.hpp
borland_prefix.hpp

borland_suffix.hpp
msvc_prefix.hpp
msvc_suffix.hpp
abi_prefix.hpp
abi_suffix.hpp
auto_link.hpp
borland.hpp
clang.hpp
codegear.hpp
comeau.hpp
common_edg.hpp
compaq_cxx.hpp
cray.hpp
digitalmars.hpp
gcc.hpp
gcc_xml.hpp
greenhills.hpp
hp_acc.hpp
intel.hpp
kai.hpp
metrowerks.hpp
mpw.hpp
nvcc.hpp
pathscale.hpp
pgi.hpp
sgi_mipspro.hpp
sunpro_cc.hpp
vacpp.hpp
visualc.hpp
cmath.hpp
complex.hpp
functional.hpp
memory.hpp
utility.hpp
aix.hpp
amigaos.hpp
beos.hpp
bsd.hpp
cray.hpp
cygwin.hpp
hpux.hpp
irix.hpp
linux.hpp

macos.hpp
qnxnto.hpp
solaris.hpp
symbian.hpp
vms.hpp
vxworks.hpp
win32.hpp
posix_features.hpp
requires_threads.hpp
select_compiler_config.hpp
select_platform_config.hpp
select_stdlib_config.hpp
dinkumware.hpp
libcomo.hpp
libcpp.hpp
libstdcpp3.hpp
modena.hpp
msl.hpp
roguewave.hpp
sgi.hpp
stlport.hpp
vacpp.hpp
suffix.hpp
user.hpp
warning_disable.hpp
config.hpp
allocator_utilities.hpp
atomic_count.hpp
atomic_redef_macros.hpp
atomic_undef_macros.hpp
binary_search.hpp
call_traits.hpp
compressed_pair.hpp
container_fwd.hpp
endian.hpp
fenv.hpp
indirect_traits.hpp
interlocked.hpp
is_function_ref_tester.hpp
is_incrementable.hpp
iterator.hpp
lcast_precision.hpp
lightweight_mutex.hpp

lightweight_test.hpp
lightweight_thread.hpp
no_exceptions_support.hpp
numeric_traits.hpp
ob_call_traits.hpp
ob_compressed_pair.hpp
quick_allocator.hpp
reference_content.hpp
scoped_enum_emulation.hpp
select_type.hpp
sp_typeinfo.hpp
basic_types.hpp
GetLastError.hpp
time.hpp
timers.hpp
workaround.hpp
pointee.hpp
static_assert.hpp
version.hpp
aligned_storage.hpp
array.hpp
arg.hpp
bind.hpp
bind_cc.hpp
bind_mf2_cc.hpp
bind_mf_cc.hpp
bind_template.hpp
mem_fn.hpp
mem_fn_cc.hpp
mem_fn_template.hpp
mem_fn_vw.hpp
placeholders.hpp
storage.hpp
bind.hpp
call_traits.hpp
assert.hpp
backward_compatibility.hpp
borland.hpp
concept_def.hpp
concept_undef.hpp
general.hpp
has_constraints.hpp
msvc.hpp

usage.hpp
concept_check.hpp
allocator_traits.hpp
container_fwd.hpp
config_begin.hpp
config_end.hpp
memory_util.hpp
mpl.hpp
pair.hpp
preprocessor.hpp
type_traits.hpp
utilities.hpp
workaround.hpp
scoped_allocator.hpp
scoped_allocator_fwd.hpp
cstdint.hpp
cstdlib.hpp
foreach.hpp
foreach_fwd.hpp
float_functions.hpp
hash_float.hpp
limits.hpp
extensions.hpp
hash.hpp
hash_fwd.hpp
hash.hpp
hash_fwd.hpp
get_pointer.hpp
integer.hpp
integer_fwd.hpp
integer_traits.hpp
ios_state.hpp
io_fwd.hpp
is_placeholder.hpp
config_def.hpp
config_undef.hpp
enable_if.hpp
facade_iterator_category.hpp
minimum_category.hpp
interoperable.hpp
iterator_adaptor.hpp
iterator_categories.hpp
iterator_concepts.hpp

iterator_facade.hpp
iterator_traits.hpp
reverse_iterator.hpp
iterator.hpp
lexical_cast.hpp
limits.hpp
common_factor_rt.hpp
policy.hpp
fp_traits.hpp
round_fwd.hpp
fpclassify.hpp
math_fwd.hpp
sign.hpp
config.hpp
promotion.hpp
real_cast.hpp
user.hpp
mem_fn.hpp
next_prior.hpp
noncopyable.hpp
none.hpp
none_t.hpp
non_type.hpp
bounds.hpp
cast.hpp
conversion_traits.hpp
converter.hpp
converter_policies.hpp
bounds.hpp
conversion_traits.hpp
converter.hpp
int_float_mixture.hpp
is_subranged.hpp
meta.hpp
numeric_cast_traits.hpp
old_numeric_cast.hpp
numeric_cast_traits_common.hpp
numeric_cast_traits_long_long.hpp
sign_mixture.hpp
udt_builtin_mixture.hpp
int_float_mixture_enum.hpp
numeric_cast_traits.hpp
sign_mixture_enum.hpp

udt_builtin_mixture_enum.hpp
optional.hpp
optional_fwd.hpp
optional.hpp
alpha.h
arm.h
blackfin.h
convex.h
ia64.h
m68k.h
mips.h
parisc.h
ppc.h
pyramid.h
rs6k.h
sparc.h
superh.h
sys370.h
sys390.h
32.h
64.h
x86.h
z.h
architecture.h
endian_compat.h
os_detected.h
test.h
_cassert.h
gnu.h
_prefix.h
make.h
bsdi.h
dragonfly.h
free.h
net.h
open.h
bsd.h
macos.h
windows.h
endian.h
version_number.h
progress.hpp
equal.hpp

begin.hpp
concepts.hpp
config.hpp
const_iterator.hpp
begin.hpp
common.hpp
const_iterator.hpp
end.hpp
extract_optional_type.hpp
implementation_help.hpp
iterator.hpp
misc_concept.hpp
remove_extent.hpp
safe_bool.hpp
sfnaf.hpp
size_type.hpp
end.hpp
difference_type.hpp
distance.hpp
empty.hpp
end.hpp
functions.hpp
iterator.hpp
iterator_range.hpp
iterator_range_core.hpp
iterator_range_io.hpp
mutable_iterator.hpp
rbegin.hpp
rend.hpp
result_iterator.hpp
reverse_iterator.hpp
size.hpp
size_type.hpp
value_type.hpp
ref.hpp
swap.hpp
debug.hpp
debug_config.hpp
config.hpp
enable_warnings.hpp
fwd_decl.hpp
global_typedef.hpp
log_level.hpp

suppress_warnings.hpp
unit_test_parameters.hpp
workaround.hpp
execution_monitor.hpp
floating_point_comparison.hpp
framework.hpp
compiler_log_formatter.hpp
cpp_main.hpp
debug.hpp
exception_safety.hpp
execution_monitor.hpp
framework.hpp
interaction_based.hpp
logged_expectations.hpp
plain_report_formatter.hpp
progress_monitor.hpp
results_collector.hpp
results_reporter.hpp
test_main.hpp
test_tools.hpp
unit_test_log.hpp
unit_test_main.hpp
unit_test_monitor.hpp
unit_test_parameters.hpp
unit_test_suite.hpp
xml_log_formatter.hpp
xml_report_formatter.hpp
prg_exec_monitor.hpp
interaction_based.hpp
minimal.hpp
mock_object.hpp
compiler_log_formatter.hpp
plain_report_formatter.hpp
xml_log_formatter.hpp
xml_report_formatter.hpp
output_test_stream.hpp
predicate_result.hpp
prg_exec_monitor.hpp
progress_monitor.hpp
results_collector.hpp
results_reporter.hpp
test_observer.hpp
test_tools.hpp

unit_test.hpp
unit_test_log.hpp
unit_test_log_formatter.hpp
unit_test_monitor.hpp
unit_test_suite.hpp
unit_test_suite_impl.hpp
algorithm.hpp
assign_op.hpp
basic_cstring.hpp
basic_cstring_fwd.hpp
bcs_char_traits.hpp
compare.hpp
io.hpp
callback.hpp
class_properties.hpp
custom_manip.hpp
fixed_mapping.hpp
foreach.hpp
input_iterator_facade.hpp
token_iterator.hpp
lazy_ostream.hpp
named_params.hpp
rtti.hpp
argument.hpp
argument_factory.hpp
argv_traverser.hpp
argv_traverser.hpp
argv_traverser.ipp
basic_parameter.hpp
char_parameter.hpp
char_parameter.ipp
argument_value_usage.hpp
dual_name_parameter.hpp
dual_name_parameter.ipp
fwd.hpp
id_policy.hpp
id_policy.ipp
argument_factory.hpp
id_policy.hpp
modifier.hpp
named_parameter.hpp
named_parameter.ipp
parameter.hpp
parser.hpp

parser.ipp
typed_parameter.hpp
validation.hpp
validation.ipp
value_generator.hpp
value_handler.hpp
config.hpp
environment.hpp
environment.ipp
fwd.hpp
modifier.hpp
variable.hpp
fwd.hpp
interpret_argument_value.hpp
parameter.hpp
trace.hpp
validation.hpp
trivial_singleton.hpp
wrap_stringstream.hpp
xml_printer.hpp
timer.hpp
type.hpp
utility.hpp
visit_each.hpp
accumulate.hpp
advance.hpp
advance_fwd.hpp
alias.hpp
always.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
arg_fwd.hpp
arithmetic.hpp
assert.hpp
as_sequence.hpp
at.hpp
at_fwd.hpp
adl_barrier.hpp
advance_backward.hpp
advance_forward.hpp

apply_1st.hpp
arg_typedef.hpp
arithmetic_op.hpp
arity.hpp
arity_spec.hpp
at_impl.hpp
back_impl.hpp
basic_bind.hpp
begin_end_impl.hpp
clear_impl.hpp
common_name_wknd.hpp
comparison_op.hpp
adl.hpp
arrays.hpp
bcc.hpp
bind.hpp
compiler.hpp
ctps.hpp
dependent_nttp.hpp
dmc_ambiguous_ctps.hpp
dtp.hpp
eti.hpp
forwarding.hpp
gcc.hpp
has_apply.hpp
has_xxx.hpp
integral.hpp
intel.hpp
lambda.hpp
msvc.hpp
msvc_typename.hpp
nttp.hpp
operators.hpp
overload_resolution.hpp
pp_counter.hpp
preprocessor.hpp
static_constant.hpp
ttp.hpp
typeof.hpp
use_preprocessed.hpp
workaround.hpp
contains_impl.hpp
count_args.hpp

count_impl.hpp
empty_impl.hpp
erase_impl.hpp
erase_key_impl.hpp
filter_iter.hpp
find_if_pred.hpp
fold_impl.hpp
fold_impl_body.hpp
fold_op.hpp
fold_pred.hpp
front_impl.hpp
full_lambda.hpp
has_apply.hpp
has_begin.hpp
has_key_impl.hpp
has_rebind.hpp
has_size.hpp
has_tag.hpp
has_type.hpp
include_preprocessed.hpp
inserter_algorithm.hpp
insert_impl.hpp
insert_range_impl.hpp
integral_wrapper.hpp
is_msvc_eti_arg.hpp
iter_apply.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
iter_push_front.hpp
joint_iter.hpp
lambda_arity_param.hpp
lambda_no_ctps.hpp
lambda_spec.hpp
lambda_support.hpp
largest_int.hpp
logical_op.hpp
msvc_dtw.hpp
msvc_eti_base.hpp
msvc_is_class.hpp
msvc_never_true.hpp
msvc_type.hpp
na.hpp
na_assert.hpp

na_fwd.hpp
na_spec.hpp
nested_type_wknd.hpp
nttp_decl.hpp
numeric_cast_utils.hpp
numeric_op.hpp
O1_size_impl.hpp
order_impl.hpp
overload_names.hpp
partition_op.hpp
pop_back_impl.hpp
pop_front_impl.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp

not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp

list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp

lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp

greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp

equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp

bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp

basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp

apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp

vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp

template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp
plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp

set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
advance_backward.hpp
advance_forward.hpp
and.hpp
apply.hpp
apply_fwd.hpp
apply_wrap.hpp
arg.hpp
basic_bind.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitxor.hpp
deque.hpp
divides.hpp
equal_to.hpp
fold_impl.hpp
full_lambda.hpp
greater.hpp
greater_equal.hpp
inherit.hpp
iter_fold_if_impl.hpp
iter_fold_impl.hpp
lambda_no_ctps.hpp
less.hpp
less_equal.hpp
list.hpp
list_c.hpp
map.hpp
minus.hpp
modulus.hpp
not_equal_to.hpp
or.hpp
placeholders.hpp

plus.hpp
quote.hpp
reverse_fold_impl.hpp
reverse_iter_fold_impl.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
template_arity.hpp
times.hpp
unpack_args.hpp
vector.hpp
vector_c.hpp
add.hpp
default_params.hpp
def_params_tail.hpp
enum.hpp
ext_params.hpp
filter_params.hpp
is_seq.hpp
params.hpp
partial_spec_params.hpp
range.hpp
repeat.hpp
sub.hpp
token_equal.hpp
tuple.hpp
ptr_to_ref.hpp
push_back_impl.hpp
push_front_impl.hpp
back.hpp
empty.hpp
front.hpp
iterator.hpp
O1_size.hpp
size.hpp
tag.hpp
reverse_fold_impl.hpp
reverse_fold_impl_body.hpp
reverse_iter_fold_impl.hpp
sequence_wrapper.hpp
shift_op.hpp
single_element_iter.hpp

size_impl.hpp
sort_impl.hpp
static_cast.hpp
template_arity.hpp
template_arity_fwd.hpp
assert.hpp
data.hpp
test_case.hpp
test.hpp
traits_lambda_spec.hpp
transform_iter.hpp
type_wrapper.hpp
unwrap.hpp
value_wknd.hpp
yes_no.hpp
back.hpp
back_fwd.hpp
back_inserter.hpp
base.hpp
begin.hpp
begin_end.hpp
begin_end_fwd.hpp
bind.hpp
bind_fwd.hpp
bitand.hpp
bitor.hpp
bitwise.hpp
bitxor.hpp
bool.hpp
bool_fwd.hpp
char.hpp
char_fwd.hpp
clear.hpp
clear_fwd.hpp
comparison.hpp
contains.hpp
contains_fwd.hpp
copy.hpp
copy_if.hpp
count.hpp
count_fwd.hpp
count_if.hpp
deque.hpp

deref.hpp
distance.hpp
distance_fwd.hpp
divides.hpp
empty.hpp
empty_base.hpp
empty_fwd.hpp
empty_sequence.hpp
end.hpp
equal.hpp
equal_to.hpp
erase.hpp
erase_fwd.hpp
erase_key.hpp
erase_key_fwd.hpp
eval_if.hpp
filter_view.hpp
find.hpp
find_if.hpp
fold.hpp
for_each.hpp
front.hpp
front_fwd.hpp
front_inserter.hpp
greater.hpp
greater_equal.hpp
has_key.hpp
has_key_fwd.hpp
has_xxx.hpp
identity.hpp
if.hpp
index_if.hpp
index_of.hpp
inherit.hpp
inherit_linearly.hpp
insert.hpp
inserter.hpp
insert_fwd.hpp
insert_range.hpp
insert_range_fwd.hpp
int.hpp
integral_c.hpp
integral_c_fwd.hpp

integral_c_tag.hpp
int_fwd.hpp
is_placeholder.hpp
is_sequence.hpp
iterator_category.hpp
iterator_range.hpp
iterator_tags.hpp
iter_fold.hpp
iter_fold_if.hpp
joint_view.hpp
key_type.hpp
key_type_fwd.hpp
lambda.hpp
lambda_fwd.hpp
less.hpp
less_equal.hpp
arity.hpp
list.hpp
map.hpp
set.hpp
string.hpp
unrolling.hpp
vector.hpp
begin_end.hpp
clear.hpp
empty.hpp
front.hpp
include_preprocessed.hpp
item.hpp
iterator.hpp
numbered.hpp
numbered_c.hpp
O1_size.hpp
pop_front.hpp
list10.hpp
list10_c.hpp
list20.hpp
list20_c.hpp
list30.hpp
list30_c.hpp
list40.hpp
list40_c.hpp
list50.hpp

list50_c.hpp
push_back.hpp
push_front.hpp
size.hpp
tag.hpp
list0.hpp
list0_c.hpp
list10.hpp
list10_c.hpp
list20.hpp
list20_c.hpp
list30.hpp
list30_c.hpp
list40.hpp
list40_c.hpp
list50.hpp
list50_c.hpp
list.hpp
list_c.hpp
logical.hpp
long.hpp
long_fwd.hpp
lower_bound.hpp
at_impl.hpp
begin_end_impl.hpp
clear_impl.hpp
contains_impl.hpp
empty_impl.hpp
erase_impl.hpp
erase_key_impl.hpp
has_key_impl.hpp
include_preprocessed.hpp
insert_impl.hpp
item.hpp
iterator.hpp
key_type_impl.hpp
map0.hpp
numbered.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp

map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
size_impl.hpp
tag.hpp
value_type_impl.hpp
map0.hpp
map10.hpp
map20.hpp
map30.hpp
map40.hpp
map50.hpp
map.hpp
fixed_c.hpp
is_even.hpp
rational_c.hpp
max.hpp
max_element.hpp
min.hpp
minus.hpp
min_element.hpp
min_max.hpp
modulus.hpp
multiplies.hpp
count_impl.hpp
insert_impl.hpp
item.hpp
multiset0.hpp
tag.hpp
multiset0.hpp
negate.hpp
next.hpp
next_prior.hpp
not.hpp
not_equal_to.hpp
numeric_cast.hpp

O1_size.hpp
O1_size_fwd.hpp
or.hpp
order.hpp
order_fwd.hpp
pair.hpp
pair_view.hpp
partition.hpp
placeholders.hpp
plus.hpp
pop_back.hpp
pop_back_fwd.hpp
pop_front.hpp
pop_front_fwd.hpp
print.hpp
prior.hpp
protect.hpp
push_back.hpp
push_back_fwd.hpp
push_front.hpp
push_front_fwd.hpp
quote.hpp
range_c.hpp
remove.hpp
remove_if.hpp
replace.hpp
replace_if.hpp
reverse.hpp
reverse_fold.hpp
reverse_iter_fold.hpp
same_as.hpp
sequence_tag.hpp
sequence_tag_fwd.hpp
at_impl.hpp
begin_end_impl.hpp
clear_impl.hpp
empty_impl.hpp
erase_impl.hpp
erase_key_impl.hpp
has_key_impl.hpp
include_preprocessed.hpp
insert_impl.hpp
item.hpp

iterator.hpp
key_type_impl.hpp
numbered.hpp
numbered_c.hpp
set10.hpp
set10_c.hpp
set20.hpp
set20_c.hpp
set30.hpp
set30_c.hpp
set40.hpp
set40_c.hpp
set50.hpp
set50_c.hpp
set0.hpp
size_impl.hpp
tag.hpp
value_type_impl.hpp
set0.hpp
set0_c.hpp
set10.hpp
set10_c.hpp
set20.hpp
set20_c.hpp
set30.hpp
set30_c.hpp
set40.hpp
set40_c.hpp
set50.hpp
set50_c.hpp
set.hpp
set_c.hpp
shift_left.hpp
shift_right.hpp
single_view.hpp
size.hpp
sizeof.hpp
size_fwd.hpp
size_t.hpp
size_t_fwd.hpp
sort.hpp
stable_partition.hpp
string.hpp

switch.hpp
tag.hpp
times.hpp
transform.hpp
transform_view.hpp
unique.hpp
unpack_args.hpp
upper_bound.hpp
value_type.hpp
value_type_fwd.hpp
at.hpp
back.hpp
begin_end.hpp
clear.hpp
empty.hpp
front.hpp
include_preprocessed.hpp
item.hpp
iterator.hpp
numbered.hpp
numbered_c.hpp
O1_size.hpp
pop_back.hpp
pop_front.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp

vector50_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
push_back.hpp
push_front.hpp
size.hpp
tag.hpp
vector0.hpp
vector0.hpp
vector0_c.hpp
vector10.hpp
vector10_c.hpp
vector20.hpp
vector20_c.hpp
vector30.hpp
vector30_c.hpp
vector40.hpp
vector40_c.hpp
vector50.hpp
vector50_c.hpp
vector.hpp
vector_c.hpp
void.hpp
void_fwd.hpp
zip_view.hpp
add.hpp
dec.hpp
div_base.hpp
div.hpp
inc.hpp
mod.hpp
mul.hpp
sub.hpp
arithmetic.hpp
data.hpp

elem.hpp
enum.hpp
insert.hpp
pop_back.hpp
pop_front.hpp
push_back.hpp
push_front.hpp
remove.hpp
replace.hpp
reverse.hpp
size.hpp
to_list.hpp
to_seq.hpp
to_tuple.hpp
array.hpp
assert_msg.hpp
cat.hpp
comma.hpp
comma_if.hpp
equal.hpp
greater.hpp
greater_equal.hpp
less.hpp
less_equal.hpp
not_equal.hpp
comparison.hpp
config.hpp
limits.hpp
deduce_d.hpp
while.hpp
while.hpp
while.hpp
while.hpp
expr_if.hpp
expr_iif.hpp
if.hpp
iif.hpp
while.hpp
control.hpp
assert.hpp
error.hpp
line.hpp
debug.hpp

dec.hpp
auto_rec.hpp
check.hpp
auto_rec.hpp
is_binary.hpp
is_nullary.hpp
is_unary.hpp
null.hpp
split.hpp
empty.hpp
enum.hpp
enum_params.hpp
enum_params_with_a_default.hpp
enum_params_with_defaults.hpp
enum_shifted.hpp
enum_shifted_params.hpp
expand.hpp
expr_if.hpp
apply.hpp
empty.hpp
expand.hpp
identity.hpp
intercept.hpp
is_1.hpp
is_empty.hpp
is_empty_or_1.hpp
overload.hpp
facilities.hpp
for.hpp
identity.hpp
if.hpp
inc.hpp
iterate.hpp
lower1.hpp
lower2.hpp
lower3.hpp
lower4.hpp
lower5.hpp
upper1.hpp
upper2.hpp
upper3.hpp
upper4.hpp
upper5.hpp

finish.hpp
forward1.hpp
forward2.hpp
forward3.hpp
forward4.hpp
forward5.hpp
reverse1.hpp
reverse2.hpp
reverse3.hpp
reverse4.hpp
reverse5.hpp
local.hpp
rlocal.hpp
self.hpp
start.hpp
iterate.hpp
local.hpp
self.hpp
iteration.hpp
library.hpp
limits.hpp
adt.hpp
append.hpp
at.hpp
cat.hpp
fold_left.hpp
fold_left.hpp
fold_right.hpp
fold_left.hpp
fold_right.hpp
enum.hpp
filter.hpp
first_n.hpp
fold_left.hpp
fold_right.hpp
for_each.hpp
for_each_i.hpp
for_each_product.hpp
rest_n.hpp
reverse.hpp
size.hpp
to_array.hpp
to_seq.hpp

to_tuple.hpp
transform.hpp
list.hpp
and.hpp
bitand.hpp
bitnor.hpp
bitor.hpp
bitxor.hpp
bool.hpp
compl.hpp
nor.hpp
not.hpp
or.hpp
xor.hpp
logical.hpp
max.hpp
min.hpp
comma.hpp
comma_if.hpp
paren.hpp
paren_if.hpp
punctuation.hpp
repeat.hpp
repeat_2nd.hpp
repeat_3rd.hpp
repeat_from_to.hpp
repeat_from_to_2nd.hpp
repeat_from_to_3rd.hpp
deduce_r.hpp
deduce_z.hpp
for.hpp
for.hpp
for.hpp
for.hpp
enum.hpp
enum_binary_params.hpp
enum_params.hpp
enum_params_with_a_default.hpp
enum_params_with_defaults.hpp
enum_shifted.hpp
enum_shifted_binary_params.hpp
enum_shifted_params.hpp
enum_trailing.hpp

enum_trailing_binary_params.hpp
enum_trailing_params.hpp
for.hpp
repeat.hpp
repeat_from_to.hpp
repetition.hpp
max.hpp
min.hpp
selection.hpp
cat.hpp
binary_transform.hpp
split.hpp
elem.hpp
enum.hpp
filter.hpp
first_n.hpp
fold_left.hpp
fold_right.hpp
for_each.hpp
for_each_i.hpp
for_each_product.hpp
insert.hpp
pop_back.hpp
pop_front.hpp
push_back.hpp
push_front.hpp
remove.hpp
replace.hpp
rest_n.hpp
reverse.hpp
seq.hpp
size.hpp
subseq.hpp
to_array.hpp
to_list.hpp
to_tuple.hpp
transform.hpp
seq.hpp
counter.hpp
counter.hpp
def.hpp
shared.hpp
slot1.hpp

slot2.hpp
slot3.hpp
slot4.hpp
slot5.hpp
slot.hpp
slot.hpp
stringize.hpp
eat.hpp
elem.hpp
enum.hpp
rem.hpp
reverse.hpp
size.hpp
to_array.hpp
to_list.hpp
to_seq.hpp
tuple.hpp
elem.hpp
size.hpp
to_array.hpp
to_list.hpp
to_seq.hpp
to_tuple.hpp
variadic.hpp
while.hpp
wstringize.hpp
preprocessor.hpp
tuple_basic.hpp
tuple_basic_no_partial_spec.hpp
tuple.hpp
assert.hpp
atomic.hpp
base.hpp
cas128strong.hpp
cas32strong.hpp
cas32weak.hpp
cas64strong.hpp
config.hpp
gcc-alpha.hpp
gcc-armv6plus.hpp
gcc-atomic.hpp
gcc-cas.hpp
gcc-ppc.hpp

gcc-sparcv9.hpp
gcc-x86.hpp
interlocked.hpp
link.hpp
linux-arm.hpp
lockpool.hpp
platform.hpp
type-classification.hpp
windows.hpp
atomic.hpp
cerrno.hpp
checked_delete.hpp
ceil.hpp
chrono.hpp
clock_string.hpp
config.hpp
chrono.hpp
chrono.hpp
chrono.hpp
chrono.hpp
is_evenly_divisible_by.hpp
static_assert.hpp
system.hpp
duration.hpp
system_clocks.hpp
time_point.hpp
current_function.hpp
adjust_functors.hpp
compiler_config.hpp
constrained_value.hpp
c_time.hpp
date.hpp
date_clock_device.hpp
date_defs.hpp
date_duration.hpp
date_duration_types.hpp
date_formatting.hpp
date_formatting_limited.hpp
date_formatting_locales.hpp
date_format_simple.hpp
date_generators.hpp
date_iterator.hpp
date_names_put.hpp

date_parsing.hpp
dst_rules.hpp
filetime_functions.hpp
conversion.hpp
formatters.hpp
formatters_limited.hpp
gregorian_types.hpp
greg_calendar.hpp
greg_date.hpp
greg_day.hpp
greg_day_of_year.hpp
greg_duration.hpp
greg_duration_types.hpp
greg_facet.hpp
greg_month.hpp
greg_weekday.hpp
greg_year.hpp
greg_ymd.hpp
parsers.hpp
gregorian_calendar.hpp
gregorian_calendar.ipp
int_adapter.hpp
iso_format.hpp
locale_config.hpp
microsec_time_clock.hpp
parse_format_base.hpp
period.hpp
conversion.hpp
date_duration_operators.hpp
posix_time_config.hpp
posix_time_duration.hpp
posix_time_system.hpp
posix_time_types.hpp
ptime.hpp
time_period.hpp
special_defs.hpp
time.hpp
time_clock.hpp
time_defs.hpp
time_duration.hpp
time_iterator.hpp
time_resolution_traits.hpp
time_system_counted.hpp

time_system_split.hpp
wrapping_int.hpp
year_month_day.hpp
enable_shared_from_this.hpp
current_exception_cast.hpp
attribute_noreturn.hpp
clone_current_exception.hpp
error_info_impl.hpp
exception_ptr.hpp
is_output_streamable.hpp
object_hex_dump.hpp
type_info.hpp
diagnostic_information.hpp
exception.hpp
get_error_info.hpp
info.hpp
to_string.hpp
to_string_stub.hpp
exception_ptr.hpp
static_log2.hpp
config_begin.hpp
config_end.hpp
has_member_function_callable_with.hpp
memory_util.hpp
mpl.hpp
preprocessor.hpp
workaround.hpp
pointer_traits.hpp
intrusive_ptr.hpp
make_shared.hpp
math_fwd.hpp
memory_order.hpp
operators.hpp
pointer_cast.hpp
pointer_to_other.hpp
config.hpp
abs.hpp
gcd.hpp
lcm.hpp
sign.hpp
overflow_helpers.hpp
rational_c_tag.hpp
ratio.hpp

ratio_fwd.hpp
rational.hpp
scoped_array.hpp
scoped_ptr.hpp
shared_array.hpp
shared_ptr.hpp
allocate_shared_array.hpp
bad_weak_ptr.hpp
allocate_array_helper.hpp
array_deleter.hpp
array_traits.hpp
array_utility.hpp
atomic_count.hpp
atomic_count_gcc.hpp
atomic_count_gcc_x86.hpp
atomic_count_pthreads.hpp
atomic_count_solaris.hpp
atomic_count_sync.hpp
atomic_count_win32.hpp
lightweight_mutex.hpp
lwm_nop.hpp
lwm_pthreads.hpp
lwm_win32_cs.hpp
make_array_helper.hpp
operator_bool.hpp
quick_allocator.hpp
shared_array_nmt.hpp
shared_count.hpp
shared_ptr_nmt.hpp
spinlock.hpp
spinlock_gcc_arm.hpp
spinlock_nt.hpp
spinlock_pool.hpp
spinlock_pt.hpp
spinlock_sync.hpp
spinlock_w32.hpp
sp_convertible.hpp
sp_counted_base.hpp
sp_counted_base_acc_ia64.hpp
sp_counted_base_aix.hpp
sp_counted_base_cw_ppc.hpp
sp_counted_base_cw_x86.hpp
sp_counted_base_gcc_ia64.hpp

sp_counted_base_gcc_mips.hpp
sp_counted_base_gcc_ppc.hpp
sp_counted_base_gcc_sparc.hpp
sp_counted_base_gcc_x86.hpp
sp_counted_base_nt.hpp
sp_counted_base_pt.hpp
sp_counted_base_snc_ps3.hpp
sp_counted_base_solaris.hpp
sp_counted_base_spin.hpp
sp_counted_base_sync.hpp
sp_counted_base_vacpp_ppc.hpp
sp_counted_base_w32.hpp
sp_counted_impl.hpp
sp_forward.hpp
sp_has_sync.hpp
sp_if_array.hpp
sp_nullptr_t.hpp
yield_k.hpp
enable_shared_from_raw.hpp
enable_shared_from_this.hpp
intrusive_ptr.hpp
intrusive_ref_counter.hpp
make_shared.hpp
make_shared_array.hpp
make_shared_object.hpp
owner_less.hpp
scoped_array.hpp
scoped_ptr.hpp
shared_array.hpp
shared_ptr.hpp
weak_ptr.hpp
smart_ptr.hpp
api_config.hpp
config.hpp
error_code.hpp
system_error.hpp
condition_variable.hpp
cv_status.hpp
async_func.hpp
config.hpp
delete.hpp
invoke.hpp
is_convertible.hpp

lockable_wrapper.hpp
make_tuple_indices.hpp
memory.hpp
move.hpp
platform.hpp
thread.hpp
thread_heap_alloc.hpp
thread_interruption.hpp
tss_hooks.hpp
exceptions.hpp
future.hpp
future_error_code.hpp
is_locked_by_this_thread.hpp
lockable_traits.hpp
locks.hpp
lock_algorithms.hpp
lock_guard.hpp
lock_options.hpp
lock_types.hpp
mutex.hpp
once.hpp
condition_variable.hpp
condition_variable_fwd.hpp
mutex.hpp
once.hpp
once_atomic.hpp
pthread_mutex_scoped_lock.hpp
shared_mutex.hpp
thread_data.hpp
thread_heap_alloc.hpp
timespec.hpp
shared_mutex.hpp
thread_only.hpp
thread_time.hpp
tss.hpp
thread.hpp
basic_timed_mutex.hpp
condition_variable.hpp
interlocked_read.hpp
mutex.hpp
once.hpp
shared_mutex.hpp
thread_data.hpp

thread_heap_alloc.hpp
thread_primitives.hpp
xtime.hpp
throw_exception.hpp
tokenizer.hpp
token_functions.hpp
token_iterator.hpp
addressof.hpp
base_from_member.hpp
binary.hpp
compare_pointees.hpp
declval.hpp
in_place_factory_prefix.hpp
in_place_factory_suffix.hpp
result_of_iterate.hpp
enable_if.hpp
identity_type.hpp
in_place_factory.hpp
result_of.hpp
swap.hpp
value_init.hpp
weak_ptr.hpp
function_iterate.hpp
gen_maybe_include.pl
maybe_include.hpp
prologue.hpp
function0.hpp
function1.hpp
function10.hpp
function2.hpp
function3.hpp
function4.hpp
function5.hpp
function6.hpp
function7.hpp
function8.hpp
function9.hpp
function_base.hpp
function_fwd.hpp
function_template.hpp
function.hpp
function_equal.hpp
algorithm.hpp

core.hpp
config_begin.hpp
config_end.hpp
meta_utils.hpp
iterator.hpp
move.hpp
traits.hpp
utility.hpp
add_const.hpp
add_cv.hpp
add_lvalue_reference.hpp
add_pointer.hpp
add_reference.hpp
add_rvalue_reference.hpp
add_volatile.hpp
aligned_storage.hpp
alignment_of.hpp
alignment_traits.hpp
arithmetic_traits.hpp
array_traits.hpp
broken_compiler_spec.hpp
common_type.hpp
composite_traits.hpp
conditional.hpp
config.hpp
conversion_traits.hpp
cv_traits.hpp
decay.hpp
bool_trait_def.hpp
bool_trait_undef.hpp
common_type_imp.hpp
cv_traits_impl.hpp
false_result.hpp
has_binary_operator.hpp
has_postfix_operator.hpp
has_prefix_operator.hpp
ice_and.hpp
ice_eq.hpp
ice_not.hpp
ice_or.hpp
is_function_ptr_helper.hpp
is_function_ptr_tester.hpp
is_mem_fun_pointer_impl.hpp

is_mem_fun_pointer_tester.hpp
size_t_trait_def.hpp
size_t_trait_undef.hpp
template_arity_spec.hpp
type_trait_def.hpp
type_trait_undef.hpp
wrap.hpp
yes_no_type.hpp
extent.hpp
floating_point_promotion.hpp
function_traits.hpp
has_bit_and.hpp
has_bit_and_assign.hpp
has_bit_or.hpp
has_bit_or_assign.hpp
has_bit_xor.hpp
has_bit_xor_assign.hpp
has_complement.hpp
has_dereference.hpp
has_divides.hpp
has_divides_assign.hpp
has_equal_to.hpp
has_greater.hpp
has_greater_equal.hpp
has_left_shift.hpp
has_left_shift_assign.hpp
has_less.hpp
has_less_equal.hpp
has_logical_and.hpp
has_logical_not.hpp
has_logical_or.hpp
has_minus.hpp
has_minus_assign.hpp
has_modulus.hpp
has_modulus_assign.hpp
has_multiplies.hpp
has_multiplies_assign.hpp
has_negate.hpp
has_new_operator.hpp
has_nothrow_assign.hpp
has_nothrow_constructor.hpp
has_nothrow_copy.hpp
has_nothrow_destructor.hpp

has_not_equal_to.hpp
has_operator.hpp
has_plus.hpp
has_plus_assign.hpp
has_post_decrement.hpp
has_post_increment.hpp
has_pre_decrement.hpp
has_pre_increment.hpp
has_right_shift.hpp
has_right_shift_assign.hpp
has_trivial_assign.hpp
has_trivial_constructor.hpp
has_trivial_copy.hpp
has_trivial_destructor.hpp
has_trivial_move_assign.hpp
has_trivial_move_constructor.hpp
has_unary_minus.hpp
has_unary_plus.hpp
has_virtual_destructor.hpp
ice.hpp
integral_constant.hpp
integral_promotion.hpp
intrinsics.hpp
is_abstract.hpp
is_arithmetic.hpp
is_array.hpp
is_base_and_derived.hpp
is_base_of.hpp
is_base_of_tr1.hpp
is_class.hpp
is_complex.hpp
is_compound.hpp
is_const.hpp
is_convertible.hpp
is_copy_constructible.hpp
is_empty.hpp
is_enum.hpp
is_float.hpp
is_floating_point.hpp
is_function.hpp
is_fundamental.hpp
is_integral.hpp
is_lvalue_reference.hpp

is_member_function_pointer.hpp
is_member_object_pointer.hpp
is_member_pointer.hpp
is_nothrow_move_assignable.hpp
is_nothrow_move_constructible.hpp
is_object.hpp
is_pod.hpp
is_pointer.hpp
is_polymorphic.hpp
is_reference.hpp
is_rvalue_reference.hpp
is_same.hpp
is_scalar.hpp
is_signed.hpp
is_stateless.hpp
is_union.hpp
is_unsigned.hpp
is_virtual_base_of.hpp
is_void.hpp
is_volatile.hpp
make_signed.hpp
make_unsigned.hpp
remove_all_extents.hpp
remove_bounds.hpp
remove_const.hpp
remove_cv.hpp
remove_extent.hpp
remove_pointer.hpp
remove_reference.hpp
remove_volatile.hpp
typeof.hpp
object_traits.hpp
promote.hpp
rank.hpp
reference_traits.hpp
remove_all_extents.hpp
remove_bounds.hpp
remove_const.hpp
remove_cv.hpp
remove_extent.hpp
remove_pointer.hpp
remove_reference.hpp
remove_volatile.hpp

same_traits.hpp
transform_traits.hpp
transform_traits_spec.hpp
type_with_alignment.hpp
type_traits.hpp
pool_construct.bat
pool_construct.sh
pool_construct_simple.bat
pool_construct_simple.sh
LICENSE_1_0.txt
must_read_library_version.txt

Academic Free License v2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

- a) to reproduce the Original Work in copies;
- b) to prepare derivative works ("Derivative Works") based upon the Original Work;
- c) to distribute copies of the Original Work and Derivative Works to the public;
- d) to perform the Original Work publicly; and
- e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as

Licensors continue to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensors, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensors except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensors even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensors from licensing under different terms from this License any Original Work that Licensors otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensors warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensors are owned by the Licensors or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensors be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensors' negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensors and

You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. § 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You. This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

Files:

File = 30574.c

```
#include <config.h>

#include <stdio.h>
#include <stdlib.h>
#include <dbus/dbus.h>
#include <glib.h>
#include <dbus/dbus-glib-lowlevel.h>

DBusConnection *bus;
GMainContext *main_context;

typedef struct _SpiReentrantCallClosure
{
    GMainLoop *loop;
    DBusMessage *reply;
} SpiReentrantCallClosure;

static void
set_reply (DBusPendingCall * pending, void *user_data)
{
    SpiReentrantCallClosure* closure = (SpiReentrantCallClosure *)
user_data;

    closure->reply = dbus_pending_call_steal_reply (pending);
    dbus_connection_setup_with_g_main (bus, NULL);

    g_main_loop_quit (closure->loop);
}

static DBusMessage *
send_and_allow_reentry (DBusConnection * bus, DBusMessage * message)
{
    DBusPendingCall *pending;
    SpiReentrantCallClosure closure;

    closure.loop = g_main_loop_new (main_context, FALSE);
    dbus_connection_setup_with_g_main (bus, main_context);

    if (!dbus_connection_send_with_reply (bus, message, &pending, 3000))
    {
        dbus_connection_setup_with_g_main (bus, NULL);
        return NULL;
    }
    dbus_pending_call_set_notify (pending, set_reply, (void *) &closure,
NULL);
    g_main_loop_run (closure.loop);

    g_main_loop_unref (closure.loop);
    return closure.reply;
}
```

```

int
main(int argc, const char *argv[])
{
    DBusError error;
    DBusMessage *message = NULL, *reply = NULL;
    const char *str;

    main_context = g_main_context_new ();
    dbus_error_init (&error);
    bus = dbus_bus_get (DBUS_BUS_SESSION, &error);
    if (!bus)
    {
        fprintf(stderr, "Couldn't connect to bus: %s\n", error.name);
        return 1;
    }
    dbus_connection_setup_with_g_main (bus, NULL);
    message = dbus_message_new_method_call ("org.freedesktop.DBus",
"/org/freedesktop/DBus", DBUS_INTERFACE_DBUS, "GetId");
    reply = send_and_allow_reentry (bus, message);
    if (dbus_message_get_type (reply) == DBUS_MESSAGE_TYPE_ERROR)
    {
        char *err;
        dbus_message_get_args (reply, NULL, DBUS_TYPE_STRING, &err,
DBUS_TYPE_INVALID);
        fprintf (stderr, "Got error: %s\n", err);
        return 1;
    }
    dbus_message_unref (reply);
    dbus_message_unref (message);
    message = dbus_message_new_method_call ("org.freedesktop.DBus",
"/org/freedesktop/DBus", DBUS_INTERFACE_DBUS, "GetId");
    reply = send_and_allow_reentry (bus, message);
    if (!reply)
    {
        fprintf(stderr, "Sorry; dbus wouldn't answer me: %s\n",
error.message);
        exit(1);
    }
    if (dbus_message_get_type (reply) == DBUS_MESSAGE_TYPE_ERROR)
    {
        char *err;
        dbus_message_get_args (reply, NULL, DBUS_TYPE_STRING, &err,
DBUS_TYPE_INVALID);
        fprintf (stderr, "Got error: %s\n", err);
        return 1;
    }
    if (!dbus_message_get_args (reply, &error, DBUS_TYPE_STRING, &str,
DBUS_TYPE_INVALID))
    {
        fprintf(stderr, "Sorry; can't communicate: %s\n", error.message);
        exit(1);
    }
}

```



```
    }  
  
    return 0;  
}
```

File = activation-exit-codes.h

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* activation-exit-codes.h Return values for the launch helper which  
is set  
*  
* in the helper and read in dbus-spawn.  
*  
* Copyright (C) 2007 Red Hat, Inc.  
*  
* Licensed under the Academic Free License version 2.1  
*  
* This program is free software; you can redistribute it and/or  
modify  
* it under the terms of the GNU General Public License as published  
by  
* the Free Software Foundation; either version 2 of the License, or  
* (at your option) any later version.  
*  
* This program is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program; if not, write to the Free Software  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA  
*  
*/
```

```
#ifndef BUS_ACTIVATION_EXIT_CODES_H  
#define BUS_ACTIVATION_EXIT_CODES_H
```

```
/** Return codes from the launch helper - not public API. However,  
* presumably if some third party did write their own launch helper,  
* they would have to rely on these, or at least always return  
* 1 for GENERIC_FAILURE.  
*/
```

```
#define BUS_SPAWN_EXIT_CODE_GENERIC_FAILURE 1  
#define BUS_SPAWN_EXIT_CODE_NO_MEMORY 2  
#define BUS_SPAWN_EXIT_CODE_CONFIG_INVALID 3  
#define BUS_SPAWN_EXIT_CODE_SETUP_FAILED 4  
#define BUS_SPAWN_EXIT_CODE_NAME_INVALID 5  
#define BUS_SPAWN_EXIT_CODE_SERVICE_NOT_FOUND 6  
#define BUS_SPAWN_EXIT_CODE_PERMISSIONS_INVALID 7
```

```
#define BUS_SPAWN_EXIT_CODE_FILE_INVALID      8
#define BUS_SPAWN_EXIT_CODE_EXEC_FAILED     9
#define BUS_SPAWN_EXIT_CODE_INVALID_ARGS    10
#define BUS_SPAWN_EXIT_CODE_CHILD_SIGNALED  11
```

```
#endif /* BUS_ACTIVATION_EXIT_CODES_H */
```

```
File = activation-helper-bin.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* activation-helper-bin.c Setuid helper for launching programs as a
custom
```

```
*
*                               user. This file is security sensitive.
*
* Copyright (C) 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/
```

```
#include <config.h>
```

```
#include "utils.h"
#include "activation-helper.h"
#include "activation-exit-codes.h"
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
static int
convert_error_to_exit_code (DBusError *error)
{
```

```

if (dbus_error_has_name (error, DBUS_ERROR_NO_MEMORY))
    return BUS_SPAWN_EXIT_CODE_NO_MEMORY;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_CONFIG_INVALID))
    return BUS_SPAWN_EXIT_CODE_CONFIG_INVALID;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_SETUP_FAILED))
    return BUS_SPAWN_EXIT_CODE_SETUP_FAILED;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_SERVICE_INVALID))
    return BUS_SPAWN_EXIT_CODE_SERVICE_NOT_FOUND;

if (dbus_error_has_name (error,
DBUS_ERROR_SPAWN_PERMISSIONS_INVALID))
    return BUS_SPAWN_EXIT_CODE_PERMISSIONS_INVALID;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_FILE_INVALID))
    return BUS_SPAWN_EXIT_CODE_FILE_INVALID;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_EXEC_FAILED))
    return BUS_SPAWN_EXIT_CODE_EXEC_FAILED;

if (dbus_error_has_name (error, DBUS_ERROR_INVALID_ARGS))
    return BUS_SPAWN_EXIT_CODE_INVALID_ARGS;

if (dbus_error_has_name (error, DBUS_ERROR_SPAWN_CHILD_SIGNALED))
    return BUS_SPAWN_EXIT_CODE_CHILD_SIGNALED;

/* should we assert? */
fprintf(stderr, "%s: %s\n", error->name, error->message);

return BUS_SPAWN_EXIT_CODE_SETUP_FAILED;
}

int
main (int argc, char **argv)
{
    DBusError error;
    int retval;

    /* default is all okay */
    retval = 0;

    /* have we used a help option or not specified the correct
arguments? */
    if (argc != 2 ||
        strcmp (argv[1], "--help") == 0 ||
        strcmp (argv[1], "-h") == 0 ||
        strcmp (argv[1], "-?") == 0)
    {
        fprintf (stderr, "dbus-daemon-activation-helper
service.to.activate\n");
    }
}

```

```

        exit (0);
    }

    dbus_error_init (&error);
    if (!run_launch_helper (argv[1], &error))
    {
        /* convert error to an exit code */
        retval = convert_error_to_exit_code (&error);
        dbus_error_free (&error);
    }

    return retval;
}

```

File = activation-helper.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* activation-helper.c Setuid helper for launching programs as a
custom
*
* user. This file is security sensitive.
*
* Copyright (C) 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>

#include "bus.h"
#include "driver.h"
#include "utils.h"

```

```

#include "desktop-file.h"
#include "config-parser-trivial.h"
#include "activation-helper.h"
#include "activation-exit-codes.h"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>
#include <grp.h>

#include <dbus/dbus-shell.h>
#include <dbus/dbus-marshal-validate.h>

static BusDesktopFile *
desktop_file_for_name (BusConfigParser *parser,
                      const char *name,
                      DBusError *error)
{
    BusDesktopFile *desktop_file;
    DBusList **service_dirs;
    DBusList *link;
    DBusError tmp_error;
    DBusString full_path;
    DBusString filename;
    const char *dir;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    desktop_file = NULL;

    if (!_dbus_string_init (&filename))
    {
        BUS_SET_OOM (error);
        goto out_all;
    }

    if (!_dbus_string_init (&full_path))
    {
        BUS_SET_OOM (error);
        goto out_filename;
    }

    if (!_dbus_string_append (&filename, name) ||
        !_dbus_string_append (&filename, ".service"))
    {
        BUS_SET_OOM (error);
        goto out;
    }
}

```

```

service_dirs = bus_config_parser_get_service_dirs (parser);
for (link = _dbus_list_get_first_link (service_dirs);
    link != NULL;
    link = _dbus_list_get_next_link (service_dirs, link))
{
    dir = link->data;
    _dbus_verbose ("Looking at '%s'\n", dir);

    dbus_error_init (&tmp_error);

    /* clear the path from last time */
    _dbus_string_set_length (&full_path, 0);

    /* build the full path */
    if (!_dbus_string_append (&full_path, dir) ||
        !_dbus_concat_dir_and_file (&full_path, &filename))
    {
        BUS_SET_OOM (error);
        goto out;
    }

    _dbus_verbose ("Trying to load file '%s'\n",
        _dbus_string_get_data (&full_path));
    desktop_file = bus_desktop_file_load (&full_path, &tmp_error);
    if (desktop_file == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);
        _dbus_verbose ("Could not load %s: %s: %s\n",
            _dbus_string_get_const_data (&full_path),
            tmp_error.name, tmp_error.message);

        /* we may have failed if the file is not found; this is not
fatal */
        if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY))
        {
            dbus_move_error (&tmp_error, error);
            /* we only bail out on OOM */
            goto out;
        }
        dbus_error_free (&tmp_error);
    }

    /* did we find the desktop file we want? */
    if (desktop_file != NULL)
        break;
}

/* Didn't find desktop file; set error */
if (desktop_file == NULL)
{
    dbus_set_error (error, DBUS_ERROR_SPAWN_SERVICE_NOT_FOUND,

```

```

        "The name %s was not provided by any .service
files",
        name);
    }

out:
    _dbus_string_free (&full_path);
out_filename:
    _dbus_string_free (&filename);
out_all:
    return desktop_file;
}

/* Clears the environment, except for DBUS_STARTER_x,
 * which we hardcode to the system bus.
 */
static dbus_bool_t
clear_environment (DBusError *error)
{
#ifdef ACTIVATION_LAUNCHER_TEST
    /* totally clear the environment */
    if (!_dbus_clearenv ())
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "could not clear environment\n");
        return FALSE;
    }
#endif

    /* Ensure the bus is set to system */
    _dbus_setenv ("DBUS_STARTER_ADDRESS",
        DBUS_SYSTEM_BUS_DEFAULT_ADDRESS);
    _dbus_setenv ("DBUS_STARTER_BUS_TYPE", "system");

    return TRUE;
}

static dbus_bool_t
check_permissions (const char *dbus_user, DBusError *error)
{
#ifdef ACTIVATION_LAUNCHER_TEST
    uid_t uid, euid;
    struct passwd *pw;

    pw = NULL;
    uid = 0;
    euid = 0;

    /* bail out unless the dbus user is invoking the helper */
    pw = getpwnam(dbus_user);
    if (!pw)
    {

```

```

        dbus_set_error (error, DBUS_ERROR_SPAWN_PERMISSIONS_INVALID,
                        "cannot find user '%s'", dbus_user);
        return FALSE;
    }
    uid = getuid();
    if (pw->pw_uid != uid)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_PERMISSIONS_INVALID,
                        "not invoked from user '%s'", dbus_user);
        return FALSE;
    }

    /* bail out unless we are setuid to user root */
    euid = geteuid();
    if (euid != 0)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_PERMISSIONS_INVALID,
                        "not setuid root");
        return FALSE;
    }
#endif

    return TRUE;
}

static dbus_bool_t
check_service_name (BusDesktopFile *desktop_file,
                   const char      *service_name,
                   DBusError        *error)
{
    char *name_tmp;
    dbus_bool_t retval;

    retval = FALSE;

    /* try to get Name */
    if (!bus_desktop_file_get_string (desktop_file,
                                     DBUS_SERVICE_SECTION,
                                     DBUS_SERVICE_NAME,
                                     &name_tmp,
                                     error))

        goto failed;

    /* verify that the name is the same as the file service name */
    if (strcmp (service_name, name_tmp) != 0)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_FILE_INVALID,
                        "Service '%s' does not match expected value",
name_tmp);
        goto failed_free;
    }
}

```



```

    retval = TRUE;

failed_free:
    /* we don't return the name, so free it here */
    dbus_free (name_tmp);
failed:
    return retval;
}

static dbus_bool_t
get_parameters_for_service (BusDesktopFile *desktop_file,
                           const char     *service_name,
                           char           **exec,
                           char           **user,
                           DBusError      *error)
{
    char *exec_tmp;
    char *user_tmp;

    exec_tmp = NULL;
    user_tmp = NULL;

    /* check the name of the service */
    if (!check_service_name (desktop_file, service_name, error))
        goto failed;

    /* get the complete path of the executable */
    if (!bus_desktop_file_get_string (desktop_file,
                                      DBUS_SERVICE_SECTION,
                                      DBUS_SERVICE_EXEC,
                                      &exec_tmp,
                                      error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }

    /* get the user that should run this service - user is compulsory
    for system activation */
    if (!bus_desktop_file_get_string (desktop_file,
                                      DBUS_SERVICE_SECTION,
                                      DBUS_SERVICE_USER,
                                      &user_tmp,
                                      error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }

    /* only assign if all the checks passed */
    *exec = exec_tmp;
    *user = user_tmp;
}

```

```

    return TRUE;

failed:
    dbus_free (exec_tmp);
    dbus_free (user_tmp);
    return FALSE;
}

static dbus_bool_t
switch_user (char *user, DBusError *error)
{
#ifdef ACTIVATION_LAUNCHER_TEST
    struct passwd *pw;

    /* find user */
    pw = getpwnam (user);
    if (!pw)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "cannot find user '%s'\n", user);
        return FALSE;
    }

    /* initialize the group access list */
    if (initgroups (user, pw->pw_gid))
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "could not initialize groups");
        return FALSE;
    }

    /* change to the primary group for the user */
    if (setgid (pw->pw_gid))
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "cannot setgid group %i", pw->pw_gid);
        return FALSE;
    }

    /* change to the user specified */
    if (setuid (pw->pw_uid) < 0)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "cannot setuid user %i", pw->pw_uid);
        return FALSE;
    }
#endif
    return TRUE;
}

static dbus_bool_t
exec_for_correct_user (char *exec, char *user, DBusError *error)

```

```

{
    char **argv;
    int argc;
    dbus_bool_t retval;

    argc = 0;
    retval = TRUE;
    argv = NULL;

    if (!switch_user (user, error))
        return FALSE;

    /* convert command into arguments */
    if (!_dbus_shell_parse_argv (exec, &argc, &argv, error))
        return FALSE;

#ifdef ACTIVATION_LAUNCHER_DO_OOM
    /* replace with new binary, with no environment */
    if (execv (argv[0], argv) < 0)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_EXEC_FAILED,
            "Failed to exec: %s", argv[0]);
        retval = FALSE;
    }
#endif

    dbus_free_string_array (argv);
    return retval;
}

static dbus_bool_t
check_bus_name (const char *bus_name,
                DBusError *error)
{
    DBusString str;

    _dbus_string_init_const (&str, bus_name);
    if (!_dbus_validate_bus_name (&str, 0, _dbus_string_get_length
(&str)))
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SERVICE_NOT_FOUND,
            "bus name '%s' is not a valid bus name\n",
            bus_name);
        return FALSE;
    }

    return TRUE;
}

static dbus_bool_t
get_correct_parser (BusConfigParser **parser, DBusError *error)
{

```

```

    DBusString config_file;
    dbus_bool_t retval;
#ifdef ACTIVATION_LAUNCHER_TEST
    const char *test_config_file;
#endif

    retval = FALSE;

#ifdef ACTIVATION_LAUNCHER_TEST
    test_config_file = NULL;

    /* there is no _way_ we should be setuid if this define is set.
     * but we should be doubly paranoid and check... */
    if (getuid() != geteuid())
        _dbus_assert_not_reached ("dbus-daemon-launch-helper-test binary
is setuid!");

    /* this is not a security hole. The environment variable is only
    passed in the
     * dbus-daemon-lauch-helper-test NON-SETUID launcher */
    test_config_file = _dbus_getenv ("TEST_LAUNCH_HELPER_CONFIG");
    if (test_config_file == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
            "the TEST_LAUNCH_HELPER_CONFIG env variable is
not set");
        goto out;
    }
#endif

    /* we _only_ use the predefined system config file */
    if (!_dbus_string_init (&config_file))
    {
        BUS_SET_OOM (error);
        goto out;
    }
#ifdef ACTIVATION_LAUNCHER_TEST
    if (!_dbus_string_append (&config_file, DBUS_SYSTEM_CONFIG_FILE))
    {
        BUS_SET_OOM (error);
        goto out_free_config;
    }
#else
    if (!_dbus_string_append (&config_file, test_config_file))
    {
        BUS_SET_OOM (error);
        goto out_free_config;
    }
#endif

    /* where are we pointing.... */

```

```

_dbus_verbose ("dbus-daemon-activation-helper: using config file:
%s\n",
               _dbus_string_get_const_data (&config_file));

/* get the dbus user */
*parser = bus_config_load (&config_file, TRUE, NULL, error);
if (*parser == NULL)
{
    goto out_free_config;
}

/* woot */
retval = TRUE;

out_free_config:
_dbus_string_free (&config_file);
out:
return retval;
}

static dbus_bool_t
launch_bus_name (const char *bus_name, BusConfigParser *parser,
DBusError *error)
{
    BusDesktopFile *desktop_file;
    char *exec, *user;
    dbus_bool_t retval;

    exec = NULL;
    user = NULL;
    retval = FALSE;

    /* get the correct service file for the name we are trying to
activate */
    desktop_file = desktop_file_for_name (parser, bus_name, error);
    if (desktop_file == NULL)
        return FALSE;

    /* get exec and user for service name */
    if (!get_parameters_for_service (desktop_file, bus_name, &exec,
&user, error))
        goto finish;

    _dbus_verbose ("dbus-daemon-activation-helper: Name='%s'\n",
bus_name);
    _dbus_verbose ("dbus-daemon-activation-helper: Exec='%s'\n", exec);
    _dbus_verbose ("dbus-daemon-activation-helper: User='%s'\n", user);

    /* actually execute */
    if (!exec_for_correct_user (exec, user, error))
        goto finish;
}

```

```

    retval = TRUE;

finish:
    dbus_free (exec);
    dbus_free (user);
    bus_desktop_file_free (desktop_file);
    return retval;
}

static dbus_bool_t
check_dbus_user (BusConfigParser *parser, DBusError *error)
{
    const char *dbus_user;

    dbus_user = bus_config_parser_get_user (parser);
    if (dbus_user == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_SPAWN_CONFIG_INVALID,
            "could not get user from config file\n");
        return FALSE;
    }

    /* check to see if permissions are correct */
    if (!check_permissions (dbus_user, error))
        return FALSE;

    return TRUE;
}

dbus_bool_t
run_launch_helper (const char *bus_name,
                  DBusError *error)
{
    BusConfigParser *parser;
    dbus_bool_t retval;

    parser = NULL;
    retval = FALSE;

    /* clear the environment, apart from a few select settings */
    if (!clear_environment (error))
        goto error;

    /* check to see if we have a valid bus name */
    if (!check_bus_name (bus_name, error))
        goto error;

    /* get the correct parser, either the test or default parser */
    if (!get_correct_parser (&parser, error))
        goto error;

    /* check we are being invoked by the correct dbus user */

```

```

if (!check_dbus_user (parser, error))
    goto error_free_parser;

/* launch the bus with the service defined user */
if (!launch_bus_name (bus_name, parser, error))
    goto error_free_parser;

/* woohoo! */
retval = TRUE;

error_free_parser:
    bus_config_parser_unref (parser);
error:
    return retval;
}

```

File = activation-helper.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* activation-helper.h The actual activation helper split from the
main
*
*           function for testing.
*
* Copyright (C) 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/

#ifdef BUS_ACTIVATION_HELPER_H
#define BUS_ACTIVATION_HELPER_H

```

```
dbus_bool_t run_launch_helper (const char *bus_name, DBusError
*error);
```

```
#endif /* BUS_ACTIVATION_HELPER_H */
```

```
File = activation.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* activation.c Activation of services
 *
 * Copyright (C) 2003 CodeFactory AB
 * Copyright (C) 2003 Red Hat, Inc.
 * Copyright (C) 2004 Imendio HB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */
```

```
#include <config.h>
#include "activation.h"
#include "activation-exit-codes.h"
#include "desktop-file.h"
#include "dispatch.h"
#include "services.h"
#include "test.h"
#include "utils.h"
#include <dbus/dbus-internals.h>
#include <dbus/dbus-hash.h>
#include <dbus/dbus-list.h>
#include <dbus/dbus-shell.h>
#include <dbus/dbus-spawn.h>
#include <dbus/dbus-timeout.h>
```



```

#include <dbus/dbus-sysdeps.h>
#ifdef HAVE_ERRNO_H
#include <errno.h>
#endif

struct BusActivation
{
    int refcount;
    DBusHashTable *entries;
    DBusHashTable *pending_activations;
    char *server_address;
    BusContext *context;
    int n_pending_activations; /**< This is in fact the number of
BusPendingActivationEntry,
requests, not pending
* i.e. number of pending activation
* activations per se
*/
    DBusHashTable *directories;
    DBusHashTable *environment;
};

typedef struct
{
    int refcount;
    char *dir_c;
    DBusHashTable *entries;
} BusServiceDirectory;

typedef struct
{
    int refcount;
    char *name;
    char *exec;
    char *user;
    char *systemd_service;
    unsigned long mtime;
    BusServiceDirectory *s_dir;
    char *filename;
} BusActivationEntry;

typedef struct BusPendingActivationEntry BusPendingActivationEntry;

struct BusPendingActivationEntry
{
    DBusMessage *activation_message;
    DBusConnection *connection;

    dbus_bool_t auto_activation;
};

typedef struct

```

```

{
    int refcount;
    BusActivation *activation;
    char *service_name;
    char *exec;
    char *systemd_service;
    DBusList *entries;
    int n_entries;
    DBusBabysitter *babysitter;
    DBusTimeout *timeout;
    unsigned int timeout_added : 1;
} BusPendingActivation;

#if 0
static BusServiceDirectory *
bus_service_directory_ref (BusServiceDirectory *dir)
{
    _dbus_assert (dir->refcount);

    dir->refcount++;

    return dir;
}
#endif

static void
bus_service_directory_unref (BusServiceDirectory *dir)
{
    if (dir == NULL)
        return;

    _dbus_assert (dir->refcount > 0);
    dir->refcount--;

    if (dir->refcount > 0)
        return;

    if (dir->entries)
        _dbus_hash_table_unref (dir->entries);

    dbus_free (dir->dir_c);
    dbus_free (dir);
}

static void
bus_pending_activation_entry_free (BusPendingActivationEntry *entry)
{
    if (entry->activation_message)
        dbus_message_unref (entry->activation_message);

    if (entry->connection)
        dbus_connection_unref (entry->connection);
}

```

```

    dbus_free (entry);
}

static BusPendingActivation *
bus_pending_activation_ref (BusPendingActivation *pending_activation)
{
    _dbus_assert (pending_activation->refcount > 0);
    pending_activation->refcount += 1;

    return pending_activation;
}

static void
bus_pending_activation_unref (BusPendingActivation
*pending_activation)
{
    DBusList *link;

    if (pending_activation == NULL) /* hash table requires this */
        return;

    _dbus_assert (pending_activation->refcount > 0);
    pending_activation->refcount -= 1;

    if (pending_activation->refcount > 0)
        return;

    if (pending_activation->timeout_added)
    {
        _dbus_loop_remove_timeout (bus_context_get_loop
(pending_activation->activation->context),
                                pending_activation->timeout);
        pending_activation->timeout_added = FALSE;
    }

    if (pending_activation->timeout)
        _dbus_timeout_unref (pending_activation->timeout);

    if (pending_activation->babysitter)
    {
        if (!_dbus_babysitter_set_watch_functions (pending_activation->babysitter,
                                                NULL, NULL, NULL,
                                                pending_activation->babysitter,
                                                NULL))
            _dbus_assert_not_reached ("setting watch functions to NULL
failed");

        _dbus_babysitter_unref (pending_activation->babysitter);
    }
}

```

```

dbus_free (pending_activation->service_name);
dbus_free (pending_activation->exec);
dbus_free (pending_activation->systemd_service);

link = _dbus_list_get_first_link (&pending_activation->entries);

while (link != NULL)
{
    BusPendingActivationEntry *entry = link->data;

    bus_pending_activation_entry_free (entry);

    link = _dbus_list_get_next_link (&pending_activation->entries,
link);
}
_dbus_list_clear (&pending_activation->entries);

pending_activation->activation->n_pending_activations -=
    pending_activation->n_entries;

_dbus_assert (pending_activation->activation->n_pending_activations
>= 0);

dbus_free (pending_activation);
}

static BusActivationEntry *
bus_activation_entry_ref (BusActivationEntry *entry)
{
    _dbus_assert (entry->refcount > 0);
    entry->refcount++;

    return entry;
}

static void
bus_activation_entry_unref (BusActivationEntry *entry)
{
    if (entry == NULL) /* hash table requires this */
        return;

    _dbus_assert (entry->refcount > 0);
    entry->refcount--;

    if (entry->refcount > 0)
        return;

    dbus_free (entry->name);
    dbus_free (entry->exec);
    dbus_free (entry->user);
    dbus_free (entry->filename);
}

```

```

dbus_free (entry->systemd_service);

dbus_free (entry);
}

static dbus_bool_t
update_desktop_file_entry (BusActivation      *activation,
                           BusServiceDirectory *s_dir,
                           DBusString        *filename,
                           BusDesktopFile    *desktop_file,
                           DBusError         *error)
{
    char *name, *exec, *user, *exec_tmp, *systemd_service;
    BusActivationEntry *entry;
    DBusStat stat_buf;
    DBusString file_path;
    DBusError tmp_error;
    dbus_bool_t retval;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    retval = FALSE;
    name = NULL;
    exec = NULL;
    user = NULL;
    exec_tmp = NULL;
    entry = NULL;
    systemd_service = NULL;

    dbus_error_init (&tmp_error);

    if (!_dbus_string_init (&file_path))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!_dbus_string_append (&file_path, s_dir->dir_c) ||
        !_dbus_concat_dir_and_file (&file_path, filename))
    {
        BUS_SET_OOM (error);
        goto out;
    }

    if (!_dbus_stat (&file_path, &stat_buf, NULL))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Can't stat the service file\n");
        goto out;
    }

    if (!bus_desktop_file_get_string (desktop_file,

```

```

                                DBUS_SERVICE_SECTION,
                                DBUS_SERVICE_NAME,
                                &name,
                                error))

    goto out;

if (!bus_desktop_file_get_string (desktop_file,
                                DBUS_SERVICE_SECTION,
                                DBUS_SERVICE_EXEC,
                                &exec_tmp,
                                error))

    goto out;

exec = _dbus_strdup (_dbus_replace_install_prefix (exec_tmp));
dbus_free (exec_tmp);
exec_tmp = NULL;

/* user is not _required_ unless we are using system activation */
if (!bus_desktop_file_get_string (desktop_file,
                                DBUS_SERVICE_SECTION,
                                DBUS_SERVICE_USER,
                                &user, &tmp_error))
{
    _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);
    /* if we got OOM, then exit */
    if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY))
    {
        dbus_move_error (&tmp_error, error);
        goto out;
    }
    else
    {
        /* if we have error because we didn't find anything then
continue */
        dbus_error_free (&tmp_error);
        dbus_free (user);
        user = NULL;
    }
}
_DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);

/* systemd service is never required */
if (!bus_desktop_file_get_string (desktop_file,
                                DBUS_SERVICE_SECTION,
                                DBUS_SERVICE_SYSTEMD_SERVICE,
                                &systemd_service, &tmp_error))
{
    _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);
    /* if we got OOM, then exit */
    if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY))
    {
        dbus_move_error (&tmp_error, error);

```

```

        goto out;
    }
    else
    {
        /* if we have error because we didn't find anything then
continue */
        dbus_error_free (&tmp_error);
        dbus_free (systemd_service);
        systemd_service = NULL;
    }
}

_DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);

entry = _dbus_hash_table_lookup_string (s_dir->entries,
                                       _dbus_string_get_const_data
(filename));

if (entry == NULL) /* New file */
{
    /* FIXME we need a better-defined algorithm for which service
file to
    * pick than "whichever one is first in the directory listing"
    */
    if (_dbus_hash_table_lookup_string (activation->entries, name))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Service %s already exists in activation
entry list\n", name);
        goto out;
    }

    entry = dbus_new0 (BusActivationEntry, 1);
    if (entry == NULL)
    {
        BUS_SET_OOM (error);
        goto out;
    }

    entry->name = name;
    entry->exec = exec;
    entry->user = user;
    entry->systemd_service = systemd_service;
    entry->refcount = 1;

    /* ownership has been transferred to entry, do not free
separately */
    name = NULL;
    exec = NULL;
    user = NULL;
    systemd_service = NULL;

```

```

    entry->s_dir = s_dir;
    entry->filename = _dbus_strdup (_dbus_string_get_const_data
(filename));
    if (!entry->filename)
    {
        BUS_SET_OOM (error);
        goto out;
    }

    if (!_dbus_hash_table_insert_string (activation->entries, entry-
>name, bus_activation_entry_ref (entry))
    {
        BUS_SET_OOM (error);
        goto out;
    }

    if (!_dbus_hash_table_insert_string (s_dir->entries, entry-
>filename, bus_activation_entry_ref (entry)))
    {
        /* Revert the insertion in the entries table */
        _dbus_hash_table_remove_string (activation->entries, entry-
>name);
        BUS_SET_OOM (error);
        goto out;
    }

    _dbus_verbose ("Added \"%s\" to list of services\n", entry-
>name);
    }
    else /* Just update the entry */
    {
        bus_activation_entry_ref (entry);
        _dbus_hash_table_remove_string (activation->entries, entry-
>name);

        if (_dbus_hash_table_lookup_string (activation->entries, name))
        {
            _dbus_verbose ("The new service name \"%s\" of service file
\"%s\" is already in cache, ignoring\n",
                name, _dbus_string_get_const_data
(&file_path));
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "The new service name \"%s\" of service file
\"%s\" is already in cache, ignoring\n",
                name, _dbus_string_get_const_data
(&file_path));
            goto out;
        }

        /* ownership has been transferred to entry, do not free
separately */
        dbus_free (entry->name);

```



```

    entry->name = name;
    name = NULL;

    dbus_free (entry->exec);
    entry->exec = exec;
    exec = NULL;

    dbus_free (entry->user);
    entry->user = user;
    user = NULL;

    dbus_free (entry->systemd_service);
    entry->systemd_service = systemd_service;
    systemd_service = NULL;

    if (!_dbus_hash_table_insert_string (activation->entries,
                                        entry->name,
bus_activation_entry_ref(entry)))
    {
        BUS_SET_OOM (error);
        /* Also remove path to entries hash since we want this in
sync with
        * the entries hash table */
        _dbus_hash_table_remove_string (entry->s_dir->entries,
                                        entry->filename);
        goto out;
    }
}

entry->mtime = stat_buf.mtime;
retval = TRUE;

out:
/* if these have been transferred into entry, the variables will be
NULL */
dbus_free (name);
dbus_free (exec);
dbus_free (user);
dbus_free (systemd_service);
_dbus_string_free (&file_path);

if (entry)
    bus_activation_entry_unref (entry);

return retval;
}

static dbus_bool_t
check_service_file (BusActivation      *activation,
                   BusActivationEntry *entry,
                   BusActivationEntry **updated_entry,
                   DBusError          *error)

```

```

{
    DBusStat stat_buf;
    dbus_bool_t retval;
    BusActivationEntry *tmp_entry;
    DBusString file_path;
    DBusString filename;

    retval = TRUE;
    tmp_entry = entry;

    _dbus_string_init_const (&filename, entry->filename);

    if (!_dbus_string_init (&file_path))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!_dbus_string_append (&file_path, entry->s_dir->dir_c) ||
        !_dbus_concat_dir_and_file (&file_path, &filename))
    {
        BUS_SET_OOM (error);
        retval = FALSE;
        goto out;
    }

    if (!_dbus_stat (&file_path, &stat_buf, NULL))
    {
        _dbus_verbose ("***** Can't stat file \"%s\", removing from
cache\n",
                      _dbus_string_get_const_data (&file_path));

        _dbus_hash_table_remove_string (activation->entries, entry-
>name);
        _dbus_hash_table_remove_string (entry->s_dir->entries, entry-
>filename);

        tmp_entry = NULL;
        retval = TRUE;
        goto out;
    }
    else
    {
        if (stat_buf.mtime > entry->mtime)
        {
            BusDesktopFile *desktop_file;
            DBusError tmp_error;

            dbus_error_init (&tmp_error);

            desktop_file = bus_desktop_file_load (&file_path,
&tmp_error);

```

```

    if (desktop_file == NULL)
    {
        _dbus_verbose ("Could not load %s: %s\n",
            _dbus_string_get_const_data (&file_path),
            tmp_error.message);
        if (dbus_error_has_name (&tmp_error,
            DBUS_ERROR_NO_MEMORY))
        {
            dbus_move_error (&tmp_error, error);
            retval = FALSE;
            goto out;
        }
        dbus_error_free (&tmp_error);
        retval = TRUE;
        goto out;
    }

    /* @todo We can return OOM or a DBUS_ERROR_FAILED error
     *       Handle these both better
     */
    if (!update_desktop_file_entry (activation, entry->s_dir,
        &filename, desktop_file, &tmp_error))
    {
        bus_desktop_file_free (desktop_file);
        if (dbus_error_has_name (&tmp_error,
            DBUS_ERROR_NO_MEMORY))
        {
            dbus_move_error (&tmp_error, error);
            retval = FALSE;
            goto out;
        }
        dbus_error_free (&tmp_error);
        retval = TRUE;
        goto out;
    }

    bus_desktop_file_free (desktop_file);
    retval = TRUE;
}

out:
    _dbus_string_free (&file_path);

    if (updated_entry != NULL)
        *updated_entry = tmp_entry;
    return retval;
}

/* warning: this doesn't fully "undo" itself on failure, i.e. doesn't
strip

```

```

* hash entries it already added.
*/
static dbus_bool_t
update_directory (BusActivation      *activation,
                 BusServiceDirectory *s_dir,
                 DBusError           *error)
{
    DBusDirIter *iter;
    DBusString dir, filename;
    BusDesktopFile *desktop_file;
    DBusError tmp_error;
    dbus_bool_t retval;
    BusActivationEntry *entry;
    DBusString full_path;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    iter = NULL;
    desktop_file = NULL;

    _dbus_string_init_const (&dir, s_dir->dir_c);

    if (!_dbus_string_init (&filename))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!_dbus_string_init (&full_path))
    {
        BUS_SET_OOM (error);
        _dbus_string_free (&filename);
        return FALSE;
    }

    retval = FALSE;

    /* from this point it's safe to "goto out" */

    iter = _dbus_directory_open (&dir, error);
    if (iter == NULL)
    {
        _dbus_verbose ("Failed to open directory %s: %s\n",
                     s_dir->dir_c,
                     error ? error->message : "unknown");
        goto out;
    }

    /* Now read the files */
    dbus_error_init (&tmp_error);
    while (_dbus_directory_get_next_file (iter, &filename, &tmp_error))
    {

```

```

_dbus_assert (!dbus_error_is_set (&tmp_error));

_dbus_string_set_length (&full_path, 0);

if (!_dbus_string_ends_with_c_str (&filename, ".service"))
{
    _dbus_verbose ("Skipping non-.service file %s\n",
        _dbus_string_get_const_data (&filename));
    continue;
}

entry = _dbus_hash_table_lookup_string (s_dir->entries,
_dbus_string_get_const_data (&filename));
if (entry) /* Already has this service file in the cache */
{
    if (!check_service_file (activation, entry, NULL, error))
        goto out;

    continue;
}

if (!_dbus_string_append (&full_path, s_dir->dir_c) ||
    !_dbus_concat_dir_and_file (&full_path, &filename))
{
    BUS_SET_OOM (error);
    goto out;
}

/* New file */
desktop_file = bus_desktop_file_load (&full_path, &tmp_error);
if (desktop_file == NULL)
{
    _dbus_verbose ("Could not load %s: %s\n",
        _dbus_string_get_const_data (&full_path),
        tmp_error.message);

    if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY))
    {
        dbus_move_error (&tmp_error, error);
        goto out;
    }

    dbus_error_free (&tmp_error);
    continue;
}

/* @todo We can return OOM or a DBUS_ERROR_FAILED error
 *       Handle these both better
 */
if (!update_desktop_file_entry (activation, s_dir, &filename,
desktop_file, &tmp_error))
{

```

```

        bus_desktop_file_free (desktop_file);
        desktop_file = NULL;

        _dbus_verbose ("Could not add %s to activation entry list:
%s\n",
                        _dbus_string_get_const_data (&full_path),
tmp_error.message);

        if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY)
            {
                dbus_move_error (&tmp_error, error);
                goto out;
            }

        dbus_error_free (&tmp_error);
        continue;
    }
    else
    {
        bus_desktop_file_free (desktop_file);
        desktop_file = NULL;
        continue;
    }
}

if (dbus_error_is_set (&tmp_error))
{
    dbus_move_error (&tmp_error, error);
    goto out;
}

retval = TRUE;

out:
if (!retval)
    _DBUS_ASSERT_ERROR_IS_SET (error);
else
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

if (iter != NULL)
    _dbus_directory_close (iter);
_dbus_string_free (&filename);
_dbus_string_free (&full_path);

return retval;
}

static dbus_bool_t
populate_environment (BusActivation *activation)
{
    DBusString    key;
    DBusString    value;

```

```

int          i;
char         **environment;
dbus_bool_t  retval = FALSE;

environment = _dbus_get_environment ();

if (environment == NULL)
    return FALSE;

if (!_dbus_string_init (&key))
{
    dbus_free_string_array (environment);
    return FALSE;
}

if (!_dbus_string_init (&value))
{
    _dbus_string_free (&key);
    dbus_free_string_array (environment);
    return FALSE;
}

for (i = 0; environment[i] != NULL; i++)
{
    if (!_dbus_string_append (&key, environment[i]))
        break;

    if (_dbus_string_split_on_byte (&key, '=', &value))
    {
        char *hash_key, *hash_value;

        if (!_dbus_string_steal_data (&key, &hash_key))
            break;

        if (!_dbus_string_steal_data (&value, &hash_value))
            break;

        if (!_dbus_hash_table_insert_string (activation-
>environment,
                                           hash_key, hash_value))
            break;
    }
    _dbus_string_set_length (&key, 0);
    _dbus_string_set_length (&value, 0);
}

if (environment[i] != NULL)
    goto out;

retval = TRUE;
out:

```

```

    _dbus_string_free (&key);
    _dbus_string_free (&value);
    dbus_free_string_array (environment);

    return retval;
}

dbus_bool_t
bus_activation_reload (BusActivation      *activation,
                      const DBusString   *address,
                      DBusList           **directories,
                      DBusError          *error)
{
    DBusList      *link;
    char          *dir;

    if (activation->server_address != NULL)
        dbus_free (activation->server_address);
    if (!_dbus_string_copy_data (address, &activation->server_address))
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    if (activation->entries != NULL)
        _dbus_hash_table_unref (activation->entries);
    activation->entries = _dbus_hash_table_new (DBUS_HASH_STRING, NULL,
(DBusFreeFunction)bus_activation_entry_unref);
    if (activation->entries == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    if (activation->directories != NULL)
        _dbus_hash_table_unref (activation->directories);
    activation->directories = _dbus_hash_table_new (DBUS_HASH_STRING,
NULL,
(DBusFreeFunction)bus_service_directory_unref);

    if (activation->directories == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    link = _dbus_list_get_first_link (directories);
    while (link != NULL)
    {
        BusServiceDirectory *s_dir;

```



```

dir = _dbus_strdup ((const char *) link->data);
if (!dir)
{
    BUS_SET_OOM (error);
    goto failed;
}

s_dir = dbus_new0 (BusServiceDirectory, 1);
if (!s_dir)
{
    dbus_free (dir);
    BUS_SET_OOM (error);
    goto failed;
}

s_dir->refcount = 1;
s_dir->dir_c = dir;

s_dir->entries = _dbus_hash_table_new (DBUS_HASH_STRING, NULL,
(DBusFreeFunction)bus_activation_entry_unref);

if (!s_dir->entries)
{
    bus_service_directory_unref (s_dir);
    BUS_SET_OOM (error);
    goto failed;
}

if (!_dbus_hash_table_insert_string (activation->directories,
s_dir->dir_c, s_dir))
{
    bus_service_directory_unref (s_dir);
    BUS_SET_OOM (error);
    goto failed;
}

/* only fail on OOM, it is ok if we can't read the directory */
if (!update_directory (activation, s_dir, error))
{
    if (dbus_error_has_name (error, DBUS_ERROR_NO_MEMORY))
        goto failed;
    else
        dbus_error_free (error);
}

link = _dbus_list_get_next_link (directories, link);
}

return TRUE;
failed:

```

```

    return FALSE;
}

BusActivation*
bus_activation_new (BusContext      *context,
                  const DBusString *address,
                  DBusList         **directories,
                  DBusError        *error)
{
    BusActivation *activation;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    activation = dbus_new0 (BusActivation, 1);
    if (activation == NULL)
    {
        BUS_SET_OOM (error);
        return NULL;
    }

    activation->refcount = 1;
    activation->context = context;
    activation->n_pending_activations = 0;

    if (!bus_activation_reload (activation, address, directories,
                                error))
        goto failed;

    /* Initialize this hash table once, we don't want to lose pending
     * activations on reload. */
    activation->pending_activations = _dbus_hash_table_new
(DBUS_HASH_STRING, NULL,
(DBusFreeFunction)bus_pending_activation_unref);

    if (activation->pending_activations == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    activation->environment = _dbus_hash_table_new (DBUS_HASH_STRING,
                                                    (DBusFreeFunction)
dbus_free,
                                                    (DBusFreeFunction)
dbus_free);

    if (activation->environment == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }
}

```

```

    if (!populate_environment (activation))
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    return activation;

failed:
    bus_activation_unref (activation);
    return NULL;
}

BusActivation *
bus_activation_ref (BusActivation *activation)
{
    _dbus_assert (activation->refcount > 0);

    activation->refcount += 1;

    return activation;
}

void
bus_activation_unref (BusActivation *activation)
{
    _dbus_assert (activation->refcount > 0);

    activation->refcount -= 1;

    if (activation->refcount > 0)
        return;

    dbus_free (activation->server_address);
    if (activation->entries)
        _dbus_hash_table_unref (activation->entries);
    if (activation->pending_activations)
        _dbus_hash_table_unref (activation->pending_activations);
    if (activation->directories)
        _dbus_hash_table_unref (activation->directories);
    if (activation->environment)
        _dbus_hash_table_unref (activation->environment);

    dbus_free (activation);
}

static dbus_bool_t
add_bus_environment (BusActivation *activation,
                    DBusError *error)
{
    const char *type;

```

```

    if (!bus_activation_set_environment_variable (activation,
"DBUS_STARTER_ADDRESS",
activation-
>server_address,
error))
    return FALSE;

type = bus_context_get_type (activation->context);
if (type != NULL)
{
    if (!bus_activation_set_environment_variable (activation,
"DBUS_STARTER_BUS_TYPE", type,
error))
        return FALSE;

    if (strcmp (type, "session") == 0)
    {
        if (!bus_activation_set_environment_variable (activation,
"DBUS_SESSION_BUS_ADDRESS",
activation-
>server_address,
error))
            return FALSE;
    }
    else if (strcmp (type, "system") == 0)
    {
        if (!bus_activation_set_environment_variable (activation,
"DBUS_SYSTEM_BUS_ADDRESS",
activation-
>server_address,
error))
            return FALSE;
    }
}

return TRUE;
}

typedef struct
{
    BusPendingActivation *pending_activation;
    DBusPreallocatedHash *hash_entry;
} RestorePendingData;

static void
restore_pending (void *data)
{

```

```

RestorePendingData *d = data;

_dbus_assert (d->pending_activation != NULL);
_dbus_assert (d->hash_entry != NULL);

_dbus_verbose ("Restoring pending activation for service %s, has
timeout = %d\n",
              d->pending_activation->service_name,
              d->pending_activation->timeout_added);

_dbus_hash_table_insert_string_preallocated (d->pending_activation-
>activation->pending_activations,
                                           d->hash_entry,
                                           d->pending_activation-
>service_name, d->pending_activation);

bus_pending_activation_ref (d->pending_activation);

d->hash_entry = NULL;
}

static void
free_pending_restore_data (void *data)
{
    RestorePendingData *d = data;

    if (d->hash_entry)
        _dbus_hash_table_free_preallocated_entry (d->pending_activation-
>activation->pending_activations,
                                                d->hash_entry);

    bus_pending_activation_unref (d->pending_activation);

    dbus_free (d);
}

static dbus_bool_t
add_restore_pending_to_transaction (BusTransaction *transaction,
                                   BusPendingActivation
*pending_activation)
{
    RestorePendingData *d;

    d = dbus_new (RestorePendingData, 1);
    if (d == NULL)
        return FALSE;

    d->pending_activation = pending_activation;
    d->hash_entry = _dbus_hash_table_preallocate_entry (d-
>pending_activation->activation->pending_activations);

    bus_pending_activation_ref (d->pending_activation);
}

```

```

    if (d->hash_entry == NULL ||
        !bus_transaction_add_cancel_hook (transaction, restore_pending,
d,
                                         free_pending_restore_data))
    {
        free_pending_restore_data (d);
        return FALSE;
    }

    _dbus_verbose ("Saved pending activation to be restored if the
transaction fails\n");

    return TRUE;
}

dbus_bool_t
bus_activation_service_created (BusActivation *activation,
                                const char *service_name,
                                BusTransaction *transaction,
                                DBusError *error)
{
    BusPendingActivation *pending_activation;
    DBusMessage *message;
    DBusList *link;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    /* Check if it's a pending activation */
    pending_activation = _dbus_hash_table_lookup_string (activation-
>pending_activations, service_name);

    if (!pending_activation)
        return TRUE;

    bus_context_log (activation->context,
                    DBUS_SYSTEM_LOG_INFO, "Successfully activated
service '%s'",
                    service_name);

    link = _dbus_list_get_first_link (&pending_activation->entries);
    while (link != NULL)
    {
        BusPendingActivationEntry *entry = link->data;
        DBusList *next = _dbus_list_get_next_link (&pending_activation-
>entries, link);

        if (dbus_connection_get_is_connected (entry->connection))
        {
            /* Only send activation replies to regular activation
requests. */
            if (!entry->auto_activation)

```


DBusError

```
*error)
{
    BusPendingActivation *pending_activation;
    DBusList *link;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    /* Check if it's a pending activation */
    pending_activation = _dbus_hash_table_lookup_string (activation-
>pending_activations,

bus_service_get_name (service));

    if (!pending_activation)
        return TRUE;

    link = _dbus_list_get_first_link (&pending_activation->entries);
    while (link != NULL)
    {
        BusPendingActivationEntry *entry = link->data;
        DBusList *next = _dbus_list_get_next_link (&pending_activation-
>entries, link);

        if (entry->auto_activation && dbus_connection_get_is_connected
(entry->connection))
        {
            DBusConnection *addressed_recipient;

            addressed_recipient =
bus_service_get_primary_owners_connection (service);

            /* Resume dispatching where we left off in bus_dispatch() */
            if (!bus_dispatch_matches (transaction,
                                     entry->connection,
                                     addressed_recipient,
                                     entry->activation_message,
error))
                goto error;
        }

        link = next;
    }

    if (!add_restore_pending_to_transaction (transaction,
pending_activation))
    {
        _dbus_verbose ("Could not add cancel hook to transaction to
revert removing pending activation\n");
        BUS_SET_OOM (error);
        goto error;
    }
}
```



```

    _dbus_hash_table_remove_string (activation->pending_activations,
bus_service_get_name (service));

    return TRUE;

error:
    return FALSE;
}

/**
 * FIXME @todo the error messages here would ideally be preallocated
 * so we don't need to allocate memory to send them.
 * Using the usual tactic, prealloc an OOM message, then
 * if we can't alloc the real error send the OOM error instead.
 */
static dbus_bool_t
try_send_activation_failure (BusPendingActivation *pending_activation,
                            const DBusError      *how)
{
    BusActivation *activation;
    DBusList *link;
    BusTransaction *transaction;

    activation = pending_activation->activation;

    transaction = bus_transaction_new (activation->context);
    if (transaction == NULL)
        return FALSE;

    link = _dbus_list_get_first_link (&pending_activation->entries);
    while (link != NULL)
    {
        BusPendingActivationEntry *entry = link->data;
        DBusList *next = _dbus_list_get_next_link (&pending_activation-
>entries, link);

        if (dbus_connection_get_is_connected (entry->connection))
        {
            if (!bus_transaction_send_error_reply (transaction,
                                                    entry->connection,
                                                    how,
                                                    entry-
>activation_message))
                goto error;
        }

        link = next;
    }

    bus_transaction_execute_and_free (transaction);
}

```

```

return TRUE;

error:
    if (transaction)
        bus_transaction_cancel_and_free (transaction);
    return FALSE;
}

/**
 * Free the pending activation and send an error message to all the
 * connections that were waiting for it.
 */
static void
pending_activation_failed (BusPendingActivation *pending_activation,
                           const DBusError      *how)
{
    /* FIXME use preallocated OOM messages instead of
    bus_wait_for_memory() */
    while (!try_send_activation_failure (pending_activation, how))
        _dbus_wait_for_memory ();

    /* Destroy this pending activation */
    _dbus_hash_table_remove_string (pending_activation->activation-
>pending_activations,
                                   pending_activation->service_name);
}

/**
 * Depending on the exit code of the helper, set the error accordingly
 */
static void
handle_servicehelper_exit_error (int      exit_code,
                                DBusError *error)
{
    switch (exit_code)
    {
        {
        case BUS_SPAWN_EXIT_CODE_NO_MEMORY:
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY,
                           "Launcher could not run (out of memory)");
            break;
        case BUS_SPAWN_EXIT_CODE_SETUP_FAILED:
            dbus_set_error (error, DBUS_ERROR_SPAWN_SETUP_FAILED,
                           "Failed to setup environment correctly");
            break;
        case BUS_SPAWN_EXIT_CODE_NAME_INVALID:
            dbus_set_error (error, DBUS_ERROR_SPAWN_SERVICE_INVALID,
                           "Bus name is not valid or missing");
            break;
        case BUS_SPAWN_EXIT_CODE_SERVICE_NOT_FOUND:
            dbus_set_error (error, DBUS_ERROR_SPAWN_SERVICE_NOT_FOUND,
                           "Bus name not found in system service
directory");

```

```

        break;
    case BUS_SPAWN_EXIT_CODE_PERMISSIONS_INVALID:
        dbus_set_error (error, DBUS_ERROR_SPAWN_PERMISSIONS_INVALID,
            "The permission of the setuid helper is not
correct");
        break;
    case BUS_SPAWN_EXIT_CODE_FILE_INVALID:
        dbus_set_error (error, DBUS_ERROR_SPAWN_PERMISSIONS_INVALID,
            "The service file is incorrect or does not have
all required attributes");
        break;
    case BUS_SPAWN_EXIT_CODE_EXEC_FAILED:
        dbus_set_error (error, DBUS_ERROR_SPAWN_EXEC_FAILED,
            "Cannot launch daemon, file not found or
permissions invalid");
        break;
    case BUS_SPAWN_EXIT_CODE_INVALID_ARGS:
        dbus_set_error (error, DBUS_ERROR_INVALID_ARGS,
            "Invalid arguments to command line");
        break;
    case BUS_SPAWN_EXIT_CODE_CHILD_SIGNALED:
        dbus_set_error (error, DBUS_ERROR_SPAWN_CHILD_SIGNALED,
            "Launched child was signaled, it probably
crashed");
        break;
    default:
        dbus_set_error (error, DBUS_ERROR_SPAWN_CHILD_EXITED,
            "Launch helper exited with unknown return code
%i", exit_code);
        break;
    }
}

```

```

static void
pending_activation_finished_cb (DBusBabysitter *babysitter,
                                void *data)
{
    BusPendingActivation *pending_activation = data;
    dbus_bool_t uses_servicehelper;

    _dbus_assert (babysitter == pending_activation->babysitter);
    _dbus_babysitter_ref (babysitter);

    /* There are two major cases here; are we the system bus or the
session? Here this
    * is distinguished by whether or not we use a setuid helper
launcher. With the launch helper,
    * some process exit codes are meaningful, processed by
handle_servicehelper_exit_error.
    *
    * In both cases though, just ignore when a process exits with
status 0; it's possible for

```

```

    * a program to (misguidedly) "daemonize", and that appears to us as
an exit. This closes a race
    * condition between this code and the child process claiming the
bus name.
    */
    uses_servicehelper = bus_context_get_servicehelper
(pending_activation->activation->context) != NULL;

    /* strictly speaking this is redundant with the check in dbus-spawn
now */
    if (_dbus_babysitter_get_child_exited (babysitter))
    {
        DBusError error;
        DBusHashIter iter;
        dbus_bool_t activation_failed;
        int exit_code = 0;

        dbus_error_init (&error);

        _dbus_babysitter_set_child_exit_error (babysitter, &error);

        /* Explicitly check for SPAWN_CHILD_EXITED to avoid overwriting
an
        * exec error */
        if (dbus_error_has_name (&error, DBUS_ERROR_SPAWN_CHILD_EXITED)
            && _dbus_babysitter_get_child_exit_status (babysitter,
&exit_code))
        {
            activation_failed = exit_code != 0;

            dbus_error_free(&error);

            if (activation_failed)
            {
                if (uses_servicehelper)
                    handle_servicehelper_exit_error (exit_code, &error);
                else
                    _dbus_babysitter_set_child_exit_error (babysitter,
&error);
            }
        }
        else
        {
            activation_failed = TRUE;
        }

        if (activation_failed)
        {
            bus_context_log (pending_activation->activation->context,
                DBUS_SYSTEM_LOG_INFO, "Activated service
's' failed: %s",
                pending_activation->service_name,

```

```

        error.message);

        /* Destroy all pending activations with the same exec */
        _dbus_hash_iter_init (pending_activation->activation-
>pending_activations,
                            &iter);
        while (_dbus_hash_iter_next (&iter))
        {
            BusPendingActivation *p = _dbus_hash_iter_get_value
(&iter);

            if (p != pending_activation && strcmp (p->exec,
pending_activation->exec) == 0)
                pending_activation_failed (p, &error);
        }

        /* Destroys the pending activation */
        pending_activation_failed (pending_activation, &error);

        dbus_error_free (&error);
    }
}

_dbus_babysitter_unref (babysitter);
}

static dbus_bool_t
add_babysitter_watch (DBusWatch      *watch,
                      void           *data)
{
    BusPendingActivation *pending_activation = data;

    return _dbus_loop_add_watch (
        bus_context_get_loop (pending_activation->activation->context),
        watch);
}

static void
remove_babysitter_watch (DBusWatch      *watch,
                          void           *data)
{
    BusPendingActivation *pending_activation = data;

    _dbus_loop_remove_watch (bus_context_get_loop (pending_activation-
>activation->context),
                             watch);
}

static void
toggle_babysitter_watch (DBusWatch      *watch,
                          void           *data)
{

```

```

    BusPendingActivation *pending_activation = data;

    _dbus_loop_toggle_watch (bus_context_get_loop (pending_activation-
>activation->context),
                            watch);
}

static dbus_bool_t
pending_activation_timed_out (void *data)
{
    BusPendingActivation *pending_activation = data;
    DBusError error;

    /* Kill the spawned process, since it sucks
     * (not sure this is what we want to do, but
     * may as well try it for now)
     */
    if (pending_activation->babysitter)
        _dbus_babysitter_kill_child (pending_activation->babysitter);

    dbus_error_init (&error);

    dbus_set_error (&error, DBUS_ERROR_TIMED_OUT,
                   "Activation of %s timed out",
                   pending_activation->service_name);
    bus_context_log (pending_activation->activation->context,
                    DBUS_SYSTEM_LOG_INFO,
                    "Failed to activate service '%s': timed out",
                    pending_activation->service_name);

    pending_activation_failed (pending_activation, &error);

    dbus_error_free (&error);

    return TRUE;
}

static void
cancel_pending (void *data)
{
    BusPendingActivation *pending_activation = data;

    _dbus_verbose ("Canceling pending activation of %s\n",
                  pending_activation->service_name);

    if (pending_activation->babysitter)
        _dbus_babysitter_kill_child (pending_activation->babysitter);

    _dbus_hash_table_remove_string (pending_activation->activation-
>pending_activations,
                                   pending_activation->service_name);
}

```

```

static void
free_pending_cancel_data (void *data)
{
    BusPendingActivation *pending_activation = data;

    bus_pending_activation_unref (pending_activation);
}

static dbus_bool_t
add_cancel_pending_to_transaction (BusTransaction      *transaction,
                                   BusPendingActivation
                                   *pending_activation)
{
    if (!bus_transaction_add_cancel_hook (transaction, cancel_pending,
                                         pending_activation,
                                         free_pending_cancel_data))

        return FALSE;

    bus_pending_activation_ref (pending_activation);

    _dbus_verbose ("Saved pending activation to be canceled if the
transaction fails\n");

    return TRUE;
}

static dbus_bool_t
update_service_cache (BusActivation *activation, DBusError *error)
{
    DBusHashIter iter;

    _dbus_hash_iter_init (activation->directories, &iter);
    while (_dbus_hash_iter_next (&iter))
    {
        DBusError tmp_error;
        BusServiceDirectory *s_dir;

        s_dir = _dbus_hash_iter_get_value (&iter);

        dbus_error_init (&tmp_error);
        if (!update_directory (activation, s_dir, &tmp_error))
        {
            if (dbus_error_has_name (&tmp_error, DBUS_ERROR_NO_MEMORY))
            {
                dbus_move_error (&tmp_error, error);
                return FALSE;
            }

            dbus_error_free (&tmp_error);
            continue;
        }
    }
}

```

```

    }

    return TRUE;
}

static BusActivationEntry *
activation_find_entry (BusActivation *activation,
                      const char    *service_name,
                      DBusError     *error)
{
    BusActivationEntry *entry;

    entry = _dbus_hash_table_lookup_string (activation->entries,
service_name);
    if (!entry)
        {
            if (!update_service_cache (activation, error))
                return NULL;

            entry = _dbus_hash_table_lookup_string (activation->entries,
service_name);
        }
    else
        {
            BusActivationEntry *updated_entry;

            if (!check_service_file (activation, entry, &updated_entry,
error))
                return NULL;

            entry = updated_entry;
        }

    if (!entry)
        {
            dbus_set_error (error, DBUS_ERROR_SERVICE_UNKNOWN,
"The name %s was not provided by any .service
files",
service_name);

            return NULL;
        }

    return entry;
}

static char **
bus_activation_get_environment (BusActivation *activation)
{
    char **environment;
    int i, length;
    DBusString entry;
    DBusHashIter iter;

```



```

length = _dbus_hash_table_get_n_entries (activation->environment);
environment = dbus_new0 (char *, length + 1);

if (environment == NULL)
    return NULL;

i = 0;
_dbus_hash_iter_init (activation->environment, &iter);

if (!_dbus_string_init (&entry))
{
    dbus_free_string_array (environment);
    return NULL;
}

while (_dbus_hash_iter_next (&iter))
{
    const char *key, *value;

    key = (const char *) _dbus_hash_iter_get_string_key (&iter);
    value = (const char *) _dbus_hash_iter_get_value (&iter);

    if (!_dbus_string_append_printf (&entry, "%s=%s", key, value))
        break;

    if (!_dbus_string_steal_data (&entry, environment + i))
        break;
    i++;
}

_dbus_string_free (&entry);

if (i != length)
{
    dbus_free_string_array (environment);
    environment = NULL;
}

return environment;
}

dbus_bool_t
bus_activation_set_environment_variable (BusActivation
*activation,
                                     const char    *key,
                                     const char    *value,
                                     DBusError     *error)
{
    char    *hash_key;
    char    *hash_value;

```

```

dbus_bool_t  retval;

retval = FALSE;
hash_key = NULL;
hash_value = NULL;
hash_key = _dbus_strdup (key);

if (hash_key == NULL)
    goto out;

hash_value = _dbus_strdup (value);

if (hash_value == NULL)
    goto out;

if (!_dbus_hash_table_insert_string (activation->environment,
                                     hash_key, hash_value))
    goto out;

retval = TRUE;
out:
if (retval == FALSE)
    {
        dbus_free (hash_key);
        dbus_free (hash_value);
        BUS_SET_OOM (error);
    }

return retval;
}

dbus_bool_t
bus_activation_activate_service (BusActivation  *activation,
                                DBusConnection *connection,
                                BusTransaction *transaction,
                                dbus_bool_t    auto_activation,
                                DBusMessage    *activation_message,
                                const char     *service_name,
                                DBusError      *error)
{
    DBusError tmp_error;
    BusActivationEntry *entry;
    BusPendingActivation *pending_activation;
    BusPendingActivationEntry *pending_activation_entry;
    DBusMessage *message;
    DBusString service_str;
    const char *servicehelper;
    char **argv;
    char **envp = NULL;
    int argc;
    dbus_bool_t retval;
    dbus_bool_t was_pending_activation;

```



```

        _dbus_verbose ("No memory to set args of reply to
activate message\n");
        BUS_SET_OOM (error);
        dbus_message_unref (message);
        return FALSE;
    }

    retval = bus_transaction_send_from_driver (transaction,
connection, message);
    dbus_message_unref (message);
    if (!retval)
    {
        _dbus_verbose ("Failed to send reply\n");
        BUS_SET_OOM (error);
    }

    return retval;
}
}

pending_activation_entry = dbus_new0 (BusPendingActivationEntry, 1);
if (!pending_activation_entry)
{
    _dbus_verbose ("Failed to create pending activation entry\n");
    BUS_SET_OOM (error);
    return FALSE;
}

pending_activation_entry->auto_activation = auto_activation;

pending_activation_entry->activation_message = activation_message;
dbus_message_ref (activation_message);
pending_activation_entry->connection = connection;
dbus_connection_ref (connection);

/* Check if the service is being activated */
pending_activation = _dbus_hash_table_lookup_string (activation-
>pending_activations, service_name);
was_pending_activation = (pending_activation != NULL);
if (was_pending_activation)
{
    if (!_dbus_list_append (&pending_activation->entries,
pending_activation_entry))
    {
        _dbus_verbose ("Failed to append a new entry to pending
activation\n");

        BUS_SET_OOM (error);
        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }
}

```

```

    pending_activation->n_entries += 1;
    pending_activation->activation->n_pending_activations += 1;
}
else
{
    pending_activation = dbus_new0 (BusPendingActivation, 1);
    if (!pending_activation)
    {
        _dbus_verbose ("Failed to create pending activation\n");

        BUS_SET_OOM (error);
        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }

    pending_activation->activation = activation;
    pending_activation->refcount = 1;

    pending_activation->service_name = _dbus_strdup (service_name);
    if (!pending_activation->service_name)
    {
        _dbus_verbose ("Failed to copy service name for pending
activation\n");

        BUS_SET_OOM (error);
        bus_pending_activation_unref (pending_activation);
        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }

    pending_activation->exec = _dbus_strdup (entry->exec);
    if (!pending_activation->exec)
    {
        _dbus_verbose ("Failed to copy service exec for pending
activation\n");
        BUS_SET_OOM (error);
        bus_pending_activation_unref (pending_activation);
        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }

    if (entry->systemd_service)
    {
        pending_activation->systemd_service = _dbus_strdup (entry-
>systemd_service);
        if (!pending_activation->systemd_service)
        {

```

```

        _dbus_verbose ("Failed to copy systemd service for
pending activation\n");
        BUS_SET_OOM (error);
        bus_pending_activation_unref (pending_activation);
        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }
}

pending_activation->timeout =
    _dbus_timeout_new (bus_context_get_activation_timeout
(activation->context),
                    pending_activation_timed_out,
                    pending_activation,
                    NULL);
if (!pending_activation->timeout)
{
    _dbus_verbose ("Failed to create timeout for pending
activation\n");

    BUS_SET_OOM (error);
    bus_pending_activation_unref (pending_activation);
    bus_pending_activation_entry_free
(pending_activation_entry);
    return FALSE;
}

if (!_dbus_loop_add_timeout (bus_context_get_loop (activation-
>context),
                            pending_activation->timeout))
{
    _dbus_verbose ("Failed to add timeout for pending
activation\n");

    BUS_SET_OOM (error);
    bus_pending_activation_unref (pending_activation);
    bus_pending_activation_entry_free
(pending_activation_entry);
    return FALSE;
}

pending_activation->timeout_added = TRUE;

if (!_dbus_list_append (&pending_activation->entries,
pending_activation_entry))
{
    _dbus_verbose ("Failed to add entry to just-created pending
activation\n");

    BUS_SET_OOM (error);
    bus_pending_activation_unref (pending_activation);

```

```

        bus_pending_activation_entry_free
(pending_activation_entry);
        return FALSE;
    }

    pending_activation->n_entries += 1;
    pending_activation->activation->n_pending_activations += 1;

    if (!_dbus_hash_table_insert_string (activation->pending_activations,
                                        pending_activation->service_name,
                                        pending_activation))
    {
        _dbus_verbose ("Failed to put pending activation in hash
table\n");

        BUS_SET_OOM (error);
        bus_pending_activation_unref (pending_activation);
        return FALSE;
    }

    if (!add_cancel_pending_to_transaction (transaction,
pending_activation))
    {
        _dbus_verbose ("Failed to add pending activation cancel hook to
transaction\n");
        BUS_SET_OOM (error);
        _dbus_hash_table_remove_string (activation->pending_activations,
                                        pending_activation->service_name);

        return FALSE;
    }

    if (was_pending_activation)
        return TRUE;

    if (bus_context_get_systemd_activation (activation->context))
    {
        if (strcmp (service_name, "org.freedesktop.systemd1") == 0)
            /* systemd itself is missing apparently. That can happen
            only during early startup. Let's just wait until systemd
            connects to us and do nothing. */
            return TRUE;

        if (entry->systemd_service)
        {
            BusTransaction *activation_transaction;
            DBusString service_string;
            BusService *service;

```

```

BusRegistry *registry;

/* OK, we have a systemd service configured for this entry,
   hence let's enqueue an activation request message. This
   is implemented as a directed signal, not a method call,
   for three reasons: 1) we don't expect a response on
   success, where we just expect a name appearing on the
   bus; 2) at this time the systemd service might not yet
   have connected, so we wouldn't know the message serial
   at this point to set up a pending call; 3) it is ugly if
   the bus suddenly becomes the caller of a remote method. */

message = dbus_message_new_signal (DBUS_PATH_DBUS,
"org.freedesktop.systemd1.Activator",
                                "ActivationRequest");

    if (!message)
    {
        _dbus_verbose ("No memory to create activation
message\n");
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!dbus_message_set_sender (message, DBUS_SERVICE_DBUS) ||
        !dbus_message_set_destination (message,
"org.freedesktop.systemd1") ||
        !dbus_message_append_args (message,
                                DBUS_TYPE_STRING, &entry-
>systemd_service,
                                DBUS_TYPE_INVALID))
    {
        _dbus_verbose ("No memory to set args of activation
message\n");
        dbus_message_unref (message);
        BUS_SET_OOM (error);
        return FALSE;
    }

    /* Create our transaction */
    activation_transaction = bus_transaction_new (activation-
>context);
    if (activation_transaction == NULL)
    {
        _dbus_verbose ("No memory to create activation
transaction\n");
        dbus_message_unref (message);
        BUS_SET_OOM (error);
        return FALSE;
    }

    /* Check whether systemd is already connected */

```



```

        registry = bus_connection_get_registry (connection);
        _dbus_string_init_const (&service_string,
"org.freedesktop.systemd1");
        service = bus_registry_lookup (registry, &service_string);

        if (service != NULL)
        {
            bus_context_log (activation->context,
                DBUS_SYSTEM_LOG_INFO, "Activating via
systemd: service name='%s' unit='%s'",
                service_name,
                entry->systemd_service);
            /* Wonderful, systemd is connected, let's just send the
msg */
            retval = bus_dispatch_matches (activation_transaction,
NULL, bus_service_get_primary_owners_connection (service),
                message, error);
        }
        else
        {
            bus_context_log (activation->context,
                DBUS_SYSTEM_LOG_INFO, "Activating
systemd to hand-off: service name='%s' unit='%s'",
                service_name,
                entry->systemd_service);
            /* systemd is not around, let's "activate" it. */
            retval = bus_activation_activate_service (activation,
connection, activation_transaction, TRUE,
                message,
"org.freedesktop.systemd1", error);
        }

        dbus_message_unref (message);

        if (!retval)
        {
            bus_context_log (activation->context,
                DBUS_SYSTEM_LOG_INFO, "Failed to
activate via systemd: service name='%s' unit='%s'",
                service_name,
                entry->systemd_service);
            _DBUS_ASSERT_ERROR_IS_SET (error);
            _dbus_verbose ("failed to send activation message:
%s\n", error->name);
            bus_transaction_cancel_and_free
(activation_transaction);
            return FALSE;
        }

        bus_transaction_execute_and_free (activation_transaction);
        return TRUE;
    }

```

```

        /* OK, we have no configured systemd service, hence let's
           proceed with traditional activation. */
    }

    /* use command as system and session different */
    if (!_dbus_string_init (&command))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    /* does the bus use a helper? */
    servicehelper = bus_context_get_servicehelper (activation->context);
    if (servicehelper != NULL)
    {
        if (entry->user == NULL)
        {
            _dbus_string_free (&command);
            dbus_set_error (error, DBUS_ERROR_SPAWN_FILE_INVALID,
                "Cannot do system-bus activation with no
user\n");
            return FALSE;
        }

        /* join the helper path and the service name */
        if (!_dbus_string_append (&command, servicehelper))
        {
            _dbus_string_free (&command);
            BUS_SET_OOM (error);
            return FALSE;
        }
        if (!_dbus_string_append (&command, " "))
        {
            _dbus_string_free (&command);
            BUS_SET_OOM (error);
            return FALSE;
        }
        if (!_dbus_string_append (&command, service_name))
        {
            _dbus_string_free (&command);
            BUS_SET_OOM (error);
            return FALSE;
        }
    }
    else
    {
        /* the bus does not use a helper, so we can append arguments
           with the exec line */
        if (!_dbus_string_append (&command, entry->exec))
        {
            _dbus_string_free (&command);

```

```

        BUS_SET_OOM (error);
        return FALSE;
    }
}

/* convert command into arguments */
if (!_dbus_shell_parse_argv (_dbus_string_get_const_data (&command),
&argc, &argv, error))
{
    _dbus_verbose ("Failed to parse command line: %s\n", entry-
>exec);
    _DBUS_ASSERT_ERROR_IS_SET (error);

    _dbus_hash_table_remove_string (activation->pending_activations,
pending_activation-
>service_name);

    _dbus_string_free (&command);
    return FALSE;
}
_dbus_string_free (&command);

if (!add_bus_environment (activation, error))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_free_string_array (argv);
    return FALSE;
}

envp = bus_activation_get_environment (activation);

if (envp == NULL)
{
    BUS_SET_OOM (error);
    dbus_free_string_array (argv);
    return FALSE;
}

_dbus_verbose ("Spawning %s ...\n", argv[0]);
if (servicehelper != NULL)
    bus_context_log (activation->context,
                    DBUS_SYSTEM_LOG_INFO, "Activating service
name='%s' (using servicehelper)",
                    service_name);
else
    bus_context_log (activation->context,
                    DBUS_SYSTEM_LOG_INFO, "Activating service
name='%s'",
                    service_name);

dbus_error_init (&tmp_error);

```



```

int i, j, len;
char **retval;
DBusHashIter iter;

len = _dbus_hash_table_get_n_entries (activation->entries);
retval = dbus_new (char *, len + 1);

if (retval == NULL)
    return FALSE;

_dbus_hash_iter_init (activation->entries, &iter);
i = 0;
while (_dbus_hash_iter_next (&iter))
{
    BusActivationEntry *entry = _dbus_hash_iter_get_value (&iter);

    retval[i] = _dbus_strdup (entry->name);
    if (retval[i] == NULL)
        goto error;

    i++;
}

retval[i] = NULL;

if (array_len)
    *array_len = len;

*listp = retval;
return TRUE;

error:
for (j = 0; j < i; j++)
    dbus_free (retval[j]);
dbus_free (retval);

return FALSE;
}

dbus_bool_t
dbus_activation_systemd_failure (BusActivation *activation,
                                DBusMessage *message)
{
    DBusError error;
    const char *code, *str, *unit = NULL;

    dbus_error_init(&error);

    /* This is called whenever the systemd activator sent us a
       response. We'll invalidate all pending activations that match the
       unit name. */

```

```

if (dbus_message_get_args (message, &error,
                           DBUS_TYPE_STRING, &unit,
                           DBUS_TYPE_STRING, &code,
                           DBUS_TYPE_STRING, &str,
                           DBUS_TYPE_INVALID))
    dbus_set_error(&error, code, str);

if (unit)
{
    DBusHashIter iter;

    bus_context_log (activation->context,
                    DBUS_SYSTEM_LOG_INFO, "Activation via systemd
failed for unit '%s': %s",
                    unit,
                    str);

    _dbus_hash_iter_init (activation->pending_activations,
                          &iter);

    while (_dbus_hash_iter_next (&iter))
    {
        BusPendingActivation *p = _dbus_hash_iter_get_value (&iter);

        if (p->systemd_service && strcmp (p->systemd_service, unit)
== 0)
            pending_activation_failed(p, &error);
    }
}

dbus_error_free(&error);

return TRUE;
}

#ifdef DBUS_BUILD_TESTS

#include <stdio.h>

#define SERVICE_NAME_1 "MyService1"
#define SERVICE_NAME_2 "MyService2"
#define SERVICE_NAME_3 "MyService3"

#define SERVICE_FILE_1 "service-1.service"
#define SERVICE_FILE_2 "service-2.service"
#define SERVICE_FILE_3 "service-3.service"

static dbus_bool_t
test_create_service_file (DBusString *dir,
                          const char *filename,
                          const char *name,

```

```

                                const char *exec)
{
    DBusString  file_name, full_path;
    FILE        *file;
    dbus_bool_t  ret_val;

    ret_val = TRUE;
    _dbus_string_init_const (&file_name, filename);

    if (!_dbus_string_init (&full_path))
        return FALSE;

    if (!_dbus_string_append (&full_path, _dbus_string_get_const_data
(dir)) ||
        !_dbus_concat_dir_and_file (&full_path, &file_name))
    {
        ret_val = FALSE;
        goto out;
    }

    file = fopen (_dbus_string_get_const_data (&full_path), "w");
    if (!file)
    {
        ret_val = FALSE;
        goto out;
    }

    fprintf (file, "[D-BUS Service]\nName=%s\nExec=%s\n", name, exec);
    fclose (file);

out:
    _dbus_string_free (&full_path);
    return ret_val;
}

static dbus_bool_t
test_remove_service_file (DBusString *dir, const char *filename)
{
    DBusString  file_name, full_path;
    dbus_bool_t  ret_val;

    ret_val = TRUE;

    _dbus_string_init_const (&file_name, filename);

    if (!_dbus_string_init (&full_path))
        return FALSE;

    if (!_dbus_string_append (&full_path, _dbus_string_get_const_data
(dir)) ||
        !_dbus_concat_dir_and_file (&full_path, &file_name))
    {

```

```

        ret_val = FALSE;
        goto out;
    }

    if (!_dbus_delete_file (&full_path, NULL))
    {
        ret_val = FALSE;
        goto out;
    }

out:
    _dbus_string_free (&full_path);
    return ret_val;
}

static dbus_bool_t
test_remove_directory (DBusString *dir)
{
    DBusDirIter *iter;
    DBusString filename, full_path;
    dbus_bool_t ret_val;

    ret_val = TRUE;

    if (!_dbus_string_init (&filename))
        return FALSE;

    if (!_dbus_string_init (&full_path))
    {
        _dbus_string_free (&filename);
        return FALSE;
    }

    iter = _dbus_directory_open (dir, NULL);
    if (iter == NULL)
    {
        ret_val = FALSE;
        goto out;
    }

    while (_dbus_directory_get_next_file (iter, &filename, NULL))
    {
        if (!test_remove_service_file (dir, _dbus_string_get_const_data
(&filename)))
        {
            ret_val = FALSE;
            goto out;
        }
    }
    _dbus_directory_close (iter);

    if (!_dbus_delete_directory (dir, NULL))

```



```

    {
        ret_val = FALSE;
        goto out;
    }

out:
    _dbus_string_free (&filename);
    _dbus_string_free (&full_path);

    return ret_val;
}

static dbus_bool_t
init_service_reload_test (DBusString *dir)
{
    DBusStat stat_buf;

    if (!_dbus_stat (dir, &stat_buf, NULL))
    {
        if (!_dbus_create_directory (dir, NULL))
            return FALSE;
    }
    else
    {
        if (!test_remove_directory (dir))
            return FALSE;

        if (!_dbus_create_directory (dir, NULL))
            return FALSE;
    }

    /* Create one initial file */
    if (!test_create_service_file (dir, SERVICE_FILE_1, SERVICE_NAME_1,
    "exec-1"))
        return FALSE;

    return TRUE;
}

static dbus_bool_t
cleanup_service_reload_test (DBusString *dir)
{
    if (!test_remove_directory (dir))
        return FALSE;

    return TRUE;
}

typedef struct
{
    BusActivation *activation;
    const char     *service_name;
}

```

```

    dbus_bool_t    expecting_find;
} CheckData;

static dbus_bool_t
check_func (void *data)
{
    CheckData      *d;
    BusActivationEntry *entry;
    DBusError       error;
    dbus_bool_t     ret_val;

    ret_val = TRUE;
    d = data;

    dbus_error_init (&error);

    entry = activation_find_entry (d->activation, d->service_name,
&error);
    if (entry == NULL)
    {
        if (dbus_error_has_name (&error, DBUS_ERROR_NO_MEMORY))
        {
            ret_val = TRUE;
        }
        else
        {
            if (d->expecting_find)
                ret_val = FALSE;
        }

        dbus_error_free (&error);
    }
    else
    {
        if (!d->expecting_find)
            ret_val = FALSE;
    }

    return ret_val;
}

static dbus_bool_t
do_test (const char *description, dbus_bool_t oom_test, CheckData
*data)
{
    dbus_bool_t err;

    if (oom_test)
        err = !_dbus_test_oom_handling (description, check_func, data);
    else
        err = !check_func (data);
}

```

```

    if (err)
        _dbus_assert_not_reached ("Test failed");

    return TRUE;
}

static dbus_bool_t
do_service_reload_test (DBusString *dir, dbus_bool_t oom_test)
{
    BusActivation *activation;
    DBusString     address;
    DBusList       *directories;
    CheckData      d;

    directories = NULL;
    _dbus_string_init_const (&address, "");

    if (!_dbus_list_append (&directories, _dbus_string_get_data (dir)))
        return FALSE;

    activation = bus_activation_new (NULL, &address, &directories,
NULL);
    if (!activation)
        return FALSE;

    d.activation = activation;

    /* Check for existing service file */
    d.expecting_find = TRUE;
    d.service_name = SERVICE_NAME_1;

    if (!do_test ("Existing service file", oom_test, &d))
        return FALSE;

    /* Check for non-existing service file */
    d.expecting_find = FALSE;
    d.service_name = SERVICE_NAME_3;

    if (!do_test ("Nonexisting service file", oom_test, &d))
        return FALSE;

    /* Check for added service file */
    if (!test_create_service_file (dir, SERVICE_FILE_2, SERVICE_NAME_2,
"exec-2"))
        return FALSE;

    d.expecting_find = TRUE;
    d.service_name = SERVICE_NAME_2;

    if (!do_test ("Added service file", oom_test, &d))
        return FALSE;
}

```

```

/* Check for removed service file */
if (!test_remove_service_file (dir, SERVICE_FILE_2))
    return FALSE;

d.expecting_find = FALSE;
d.service_name = SERVICE_FILE_2;

if (!do_test ("Removed service file", oom_test, &d))
    return FALSE;

/* Check for updated service file */

_dbus_sleep_milliseconds (1000); /* Sleep a second to make sure the
mtime is updated */

if (!test_create_service_file (dir, SERVICE_FILE_1, SERVICE_NAME_3,
"exec-3"))
    return FALSE;

d.expecting_find = TRUE;
d.service_name = SERVICE_NAME_3;

if (!do_test ("Updated service file, part 1", oom_test, &d))
    return FALSE;

d.expecting_find = FALSE;
d.service_name = SERVICE_NAME_1;

if (!do_test ("Updated service file, part 2", oom_test, &d))
    return FALSE;

bus_activation_unref (activation);
_dbus_list_clear (&directories);

return TRUE;
}

dbus_bool_t
bus_activation_service_reload_test (const DBusString *test_data_dir)
{
    DBusString directory;

    if (!_dbus_string_init (&directory))
        return FALSE;

    if (!_dbus_string_append (&directory, _dbus_get_tmpdir()))
        return FALSE;

    if (!_dbus_string_append (&directory, "/dbus-reload-test-") ||
        !_dbus_generate_random_ascii (&directory, 6))
    {
        return FALSE;
    }
}

```

```

    }

    /* Do normal tests */
    if (!init_service_reload_test (&directory))
        _dbus_assert_not_reached ("could not initiate service reload
test");

    if (!do_service_reload_test (&directory, FALSE))
    {
        /* Do nothing? */
    }

    /* Do OOM tests */
    if (!init_service_reload_test (&directory))
        _dbus_assert_not_reached ("could not initiate service reload
test");

    if (!do_service_reload_test (&directory, TRUE))
    {
        /* Do nothing? */
    }

    /* Cleanup test directory */
    if (!cleanup_service_reload_test (&directory))
        return FALSE;

    _dbus_string_free (&directory);

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = activation.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* activation.h  Activation of services
 *
 * Copyright (C) 2003  CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,

```



```

dbus_bool_t    bus_activation_list_services    (BusActivation
*registry,
                                char            ***listp,
                                int             *array_len);
dbus_bool_t    dbus_activation_systemd_failure (BusActivation
*activation,
                                DBusMessage
*message);

dbus_bool_t    bus_activation_send_pending_auto_activation_messages
(BusActivation *activation,
                                BusService
*service,
                                BusTransaction
*transaction,
                                DBusError
*error);

```

```
#endif /* BUS_ACTIVATION_H */
```

File = alsa.m4

```

dnl Configure Paths for Alsa
dnl Some modifications by Richard Boulton <richard-alsa@tartarus.org>
dnl Christopher Lansdown <lansdoct@cs.alfred.edu>
dnl Jaroslav Kysela <perex@perex.cz>
dnl Last modification: $Id: alsa.m4,v 1.24 2004/09/15 18:48:07 tiwai
Exp $
dnl AM_PATH_ALSA([MINIMUM-VERSION [, ACTION-IF-FOUND [, ACTION-IF-NOT-
FOUND]])
dnl Test for libasound, and define ALSA_CFLAGS and ALSA_LIBS as
appropriate.
dnl enables arguments --with-alsa-prefix=
dnl                    --with-alsa-enc-prefix=
dnl                    --disable-alsatest
dnl
dnl For backwards compatibility, if ACTION_IF_NOT_FOUND is not
specified,
dnl and the alsa libraries are not found, a fatal AC_MSG_ERROR() will
result.
dnl
AC_DEFUN([AM_PATH_ALSA],
[dnl Save the original CFLAGS, LDFLAGS, and LIBS
alsa_save_CFLAGS="$CFLAGS"
alsa_save_LDFLAGS="$LDFLAGS"
alsa_save_LIBS="$LIBS"
alsa_found=yes

dnl

```

```

dnl Get the cflags and libraries for alsa
dnl
AC_ARG_WITH(alsa-prefix,
[ --with-alsa-prefix=PREFIX Prefix where Alsa library is
installed(optional)],
[alsa_prefix="$withval"], [alsa_prefix=""])

AC_ARG_WITH(alsa-inc-prefix,
[ --with-alsa-inc-prefix=PREFIX Prefix where include libraries are
(optional)],
[alsa_inc_prefix="$withval"], [alsa_inc_prefix=""])

dnl FIXME: this is not yet implemented
AC_ARG_ENABLE(alsatest,
[ --disable-alsatest Do not try to compile and run a test Alsa
program],
[enable_alsatest="$enableval"],
[enable_alsatest=yes])

dnl Add any special include directories
AC_MSG_CHECKING(for ALSA CFLAGS)
if test "$alsa_inc_prefix" != "" ; then
    ALSA_CFLAGS="$ALSA_CFLAGS -I$alsa_inc_prefix"
    CFLAGS="$CFLAGS -I$alsa_inc_prefix"
fi
AC_MSG_RESULT($ALSA_CFLAGS)

dnl add any special lib dirs
AC_MSG_CHECKING(for ALSA LDFLAGS)
if test "$alsa_prefix" != "" ; then
    ALSA_LIBS="$ALSA_LIBS -L$alsa_prefix"
    LDFLAGS="$LDFLAGS $ALSA_LIBS"
fi

dnl add the alsa library
ALSA_LIBS="$ALSA_LIBS -lasound -lm -ldl -lpthread"
LIBS="$ALSA_LIBS $LIBS"
AC_MSG_RESULT($ALSA_LIBS)

dnl Check for a working version of libasound that is of the right
version.
if test "x$enable_alsatest" = "xyes"; then
min_alsa_version=ifelse([$1], ,0.1.1,$1)
AC_MSG_CHECKING(for libasound headers version >= $min_alsa_version)
no_alsa=""
    alsa_min_major_version=`echo $min_alsa_version | \
sed 's/\([[0-9]]*\)\.\([[0-9]]*\)\.\([[0-9]]*\)/\1/'`
    alsa_min_minor_version=`echo $min_alsa_version | \
sed 's/\([[0-9]]*\)\.\([[0-9]]*\)\.\([[0-9]]*\)/\2/'`
    alsa_min_micro_version=`echo $min_alsa_version | \
sed 's/\([[0-9]]*\)\.\([[0-9]]*\)\.\([[0-9]]*\)/\3/'`

```



```

AC_LANG_SAVE
AC_LANG_C
AC_TRY_COMPILE([
#include <alsa/asoundlib.h>
], [
/* ensure backward compatibility */
#if !defined(SND_LIB_MAJOR) && defined(SOUNDLIB_VERSION_MAJOR)
#define SND_LIB_MAJOR SOUNDLIB_VERSION_MAJOR
#endif
#if !defined(SND_LIB_MINOR) && defined(SOUNDLIB_VERSION_MINOR)
#define SND_LIB_MINOR SOUNDLIB_VERSION_MINOR
#endif
#if !defined(SND_LIB_SUBMINOR) && defined(SOUNDLIB_VERSION_SUBMINOR)
#define SND_LIB_SUBMINOR SOUNDLIB_VERSION_SUBMINOR
#endif

# if(SND_LIB_MAJOR > $alsa_min_major_version)
  exit(0);
# else
#   if(SND_LIB_MAJOR < $alsa_min_major_version)
#     error not present
#   endif

#   if(SND_LIB_MINOR > $alsa_min_minor_version)
  exit(0);
#   else
#     if(SND_LIB_MINOR < $alsa_min_minor_version)
#       error not present
#     endif

#     if(SND_LIB_SUBMINOR < $alsa_min_micro_version)
#       error not present
#     endif
#   endif
# endif
exit(0);
],
  [AC_MSG_RESULT(found.)],
  [AC_MSG_RESULT(not present.)
  ifelse([$3], , [AC_MSG_ERROR(Sufficiently new version of libasound
not found.)])
  alsa_found=no]
)
AC_LANG_RESTORE
fi

```

```

dnl Now that we know that we have the right version, let's see if we
have the library and not just the headers.
if test "x$enable_alsatest" = "xyes"; then
AC_CHECK_LIB([asound], [snd_ctl_open],,
  [ifelse([$3], , [AC_MSG_ERROR(No linkable libasound was found.)])
  alsa_found=no]

```

```

)
fi

if test "x$alsa_found" = "xyes" ; then
    ifelse([$2], , :, [$2])
    LIBS=`echo $LIBS | sed 's/-lasound//g'`
    LIBS=`echo $LIBS | sed 's/ //'`
    LIBS="-lasound $LIBS"
fi
if test "x$alsa_found" = "xno" ; then
    ifelse([$3], , :, [$3])
    CFLAGS="$alsa_save_CFLAGS"
    LDFLAGS="$alsa_save_LDFLAGS"
    LIBS="$alsa_save_LIBS"
    ALSA_CFLAGS=""
    ALSA_LIBS=""
fi

dnl That should be it.  Now just export out symbols:
AC_SUBST(ALSA_CFLAGS)
AC_SUBST(ALSA_LIBS)
])

```

File = anonymous-client-successful.auth-script

this tests that a client can login anonymously

CLIENT

```

## Reject whatever mechanism the client picks first
EXPECT_COMMAND AUTH
SEND 'REJECTED DBUS_TEST_NONEXISTENT_MECH1 ANONYMOUS
DBUS_TEST_NONEXISTENT_MECH2'

```

And this time we get ANONYMOUS

```

EXPECT_COMMAND AUTH
## of course real DBUS_COOKIE_SHA1 would not send this here...
SEND 'OK 1234deadbeef'

```

```

EXPECT_COMMAND BEGIN
EXPECT_STATE AUTHENTICATED

```

File = anonymous-server-successful.auth-script

this tests the server side in a successful auth of type ANONYMOUS

```
SERVER
## verify that prior to doing anything, we haven't authed as anyone
EXPECT_HAVE_NO_CREDENTIALS
SEND 'AUTH ANONYMOUS 442d42757320312e312e31'
EXPECT_COMMAND OK
EXPECT_STATE WAITING_FOR_INPUT
SEND 'BEGIN'
EXPECT_STATE AUTHENTICATED
## verify that we are still anonymous
EXPECT_HAVE_NO_CREDENTIALS
```

File = arm-oe-linux-gnueabi-libtool

```
#!/bin/sh
```

```
# arm-oe-linux-gnueabi-libtool - Provide generalized library-building
support services.
# Generated automatically by config.status (dbus-glib) 0.100.2
# Libtool was configured on host pc45:
# NOTE: Changes made to this file will be lost: look at ltmain.sh.
#
# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005,
#           2006, 2007, 2008, 2009, 2010, 2011 Free Software
#           Foundation, Inc.
# Written by Gordon Matzigkeit, 1996
#
# This file is part of GNU Libtool.
#
# GNU Libtool is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of
# the License, or (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html, or
# obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
# The names of the tagged configurations supported by this script.
available_tags=""

# ### BEGIN LIBTOOL CONFIG

# Which release of libtool.m4 was used?
macro_version=2.4.2
macro_revision=1.3337

# Whether or not to build shared libraries.
build_libtool_libs=yes

# Whether or not to build static libraries.
build_old_libs=yes

# What type of objects to build.
pic_mode=default

# Whether or not to optimize for fast installation.
fast_install=yes

# Shell to use when invoking shell scripts.
SHELL="/bin/sh"

# An echo program that protects backslashes.
ECHO="printf %s\\n"

# The PATH separator for the build system.
PATH_SEPARATOR=":"

# The host system.
host_alias=arm-oe-linux-gnueabi
host=arm-oe-linux-gnueabi
host_os=linux-gnueabi

# The build system.
build_alias=i686-linux
build=i686-pc-linux-gnu
build_os=linux-gnu

# A sed program that does not truncate output.
SED="/bin/sed"

# Sed that helps us avoid accidentally triggering echo(1) options like
-n.
Xsed="$SED -e 1s/^X/"

# A grep program that handles long lines.
GREP="/bin/grep"
```

```
# An ERE matcher.
EGREP="/bin/grep -E"

# A literal string matcher.
FGREP="/bin/grep -F"

# A BSD- or MS-compatible name lister.
NM="arm-linux-gnueabi-hf-nm"

# Whether we need soft or hard links.
LN_S="ln -s"

# What is the maximum length of a command?
max_cmd_len=1572864

# Object file suffix (normally "o").
objext=o

# Executable file suffix (normally "").
exeext=

# whether the shell understands "unset".
lt_unset=unset

# turn spaces into newlines.
SP2NL="tr \\040 \\012"

# turn newlines into spaces.
NL2SP="tr \\015\\012 \\040\\040"

# convert $build file names to $host format.
to_host_file_cmd=func_convert_file_noop

# convert $build files to toolchain format.
to_tool_file_cmd=func_convert_file_noop

# An object symbol dumper.
OBJDUMP="arm-linux-gnueabi-hf-objdump"

# Method to check whether dependent libraries are shared objects.
deplibs_check_method="pass_all"

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd="\$MAGIC_CMD"

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=""

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob="no"
```

```

# DLL creation program.
DLLTOOL="false"

# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd="printf %s\\n"

# The archiver.
AR="arm-linux-gnueabi-hf-ar"

# Flags to create an archive.
AR_FLAGS="cru"

# How to feed a file listing to the archiver.
archiver_list_spec="@ "

# A symbol stripping program.
STRIP="arm-linux-gnueabi-hf-strip"

# Commands used to install an old-style archive.
RANLIB="arm-linux-gnueabi-hf-ranlib"
old_postinstall_cmds="chmod 644 \\$oldlib~\\$RANLIB \\$tool_oldlib"
old_postuninstall_cmds=""

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=no

# A C compiler.
LTCC="arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork
-mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# LTCC compiler flags.
LTCFLAGS="-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types -Wall -Wchar-subscripts -
Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-
arith -Wcast-align -Wfloat-equal -Wsign-compare -fno-strict-aliasing"

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe="sed -n -e 's/^.*[
]\\([ABCDGIRSTW][ABCDGIRSTW]*\\)[
]*\\([_A-Za-z][_A-Za-z0-
9]*\\)\\$/\\1 \\2 \\2/p' | sed '/ __gnu_lto/d'"

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl="sed -n -e 's/^T .* \\([.*\\)\\$/extern int
\\1();/p' -e 's/^[ABCDGIRSTW]* .* \\([.*\\)\\$/extern char \\1;/p'"

# Transform the output of nm in a C name address pair.

```

```

global_symbol_to_c_name_address="sed -n -e 's/^: \\([^\ ]*\)[^\ ]*$/
{\\\\"\\1\\", (void *) 0},/p' -e 's/^[ABCDGIRSTW]* \\([^\ ]*\) \\([^\ ]*\)
\$/ {\\"\\2\\", (void *) \\&\\2},/p'"

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix="sed -n -e 's/^: \\([^\ ]*\)[^\ ]*$/
{\\\\"\\1\\", (void *) 0},/p' -e 's/^[ABCDGIRSTW]* \\([^\ ]*\) \\(lib[^\ ]*\)
\$/ {\\"\\2\\", (void *) \\&\\2},/p' -e
's/^[ABCDGIRSTW]* \\([^\ ]*\) \\([^\ ]*\) \$/ {\\"lib\\2\\", (void *)
\\&\\2},/p'"

# Specify filename containing input files for $NM.
nm_file_list_spec="@

# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-vp1

# The name of the directory that contains temporary libtool files.
objdir=.libs

# Used to examine libraries when file_magic_cmd begins with "file".
MAGIC_CMD=file

# Must we lock files when doing compilation?
need_locks="no"

# Manifest tool.
MANIFEST_TOOL=":"

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
DSYMUTIL=""

# Tool to change global to local symbols on Mac OS X.
NMEDIT=""

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=""

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=""

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=""

# Old archive suffix (normally "a").
libext=a

# Shared library suffix (normally ".so").
shrext_cmds=".so"

```

```
# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=""

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink="PATH LD_LIBRARY_PATH LD_RUN_PATH
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH"

# Do we need the "lib" prefix for modules?
need_lib_prefix=no

# Do we need a version for libraries?
need_version=no

# Library versioning type.
version_type=linux

# Shared library runtime path variable.
runpath_var=LD_RUN_PATH

# Shared library path variable.
shlibpath_var=LD_LIBRARY_PATH

# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=no

# Format of library name prefix.
libname_spec="lib$name"

# List of archive names.  First name is the real one, the rest are
links.
# The last name is the one that the linker finds with -lNAME
library_names_spec="\${libname}\${release}\${shared_ext}\${versuffix}
\${libname}\${release}\${shared_ext}\$major \${libname}\${shared_ext}"

# The coded name of the library, if different from the real name.
soname_spec="\${libname}\${release}\${shared_ext}\$major"

# Permission mode override for installation of shared libraries.
install_override_mode=""

# Command to use after installation of a shared archive.
postinstall_cmds=""

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=""

# Commands used to finish a libtool library installation in a
directory.
```



```

finish_cmds="PATH=\\\\"$PATH:/sbin\\" ldconfig -n \${libdir}"

# As "finish_cmds", except a single script fragment to be eval'd but
# not shown.
finish_eval=""

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=yes

# Compile-time system search path for libraries.
sys_lib_search_path_spec="/home/gangadhar/gcc-linaro-arm-linux-
gnueabihf-4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf/4.7.3
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/lib/gcc /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib "

# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec="/usr/lib /lib"

# Whether dlopen is supported.
dlopen_support=unknown

# Whether dlopen of programs is supported.
dlopen_self=unknown

# Whether dlopen of statically linked programs is supported.
dlopen_self_static=unknown

# Commands to strip libraries.
old_striplib="arm-linux-gnueabihf-strip --strip-debug"
striplib="arm-linux-gnueabihf-strip --strip-unneeded"

# The linker used to build libraries.
LD="arm-linux-gnueabihf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# How to create reloadable object files.
reload_flag=" -r"
reload_cmds="\${LD}\${reload_flag} -o \${output}\${reload_objs}"

# Commands used to build an old-style archive.
old_archive_cmds="\${AR} \${AR_FLAGS} \${oldlib}\${oldobjs}~\${RANLIB}
\${tool_oldlib}"

```

```
# A language specific compiler.
CC="arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# Is the compiler the GNU compiler?
with_gcc=yes

# Compiler flag to turn off builtin functions.
no_builtin_flag=" -fno-builtin"

# Additional compiler flags for building library objects.
pic_flag=" -fPIC -DPIC"

# How to pass a linker flag through the compiler.
wl="-Wl,"

# Compiler flag to prevent dynamic linking.
link_static_flag="-static"

# Does compiler simultaneously support -c and -o options?
compiler_c_o="yes"

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=no

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=no

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec="\${wl}--export-dynamic"

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec="\${wl}--whole-archive\${convenience} \${wl}--
no-whole-archive"

# Whether the compiler copes with passing no objects directly.
compiler_needs_object="no"

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=""

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=""

# Commands used to build a shared archive.
archive_cmds="\${CC} -shared \${pic_flag} \${libobjs} \${deplibs}
\${compiler_flags} \${wl}-soname \${wl}\${soname} -o \${lib}"
archive_expsym_cmds="echo \{\{ global:\{\{\} >
\${output_objdir}/\${libname}.ver~
```

```

        cat \${export_symbols} | sed -e \\\\"s/\\\\\\(.*\\\\\\)/\\\\\\1;/\\\\\\\"
>> \${output_objdir}/\${libname.ver~
        echo \\\\"local: *; };\\\\\\\" >> \${output_objdir}/\${libname.ver~
        \${CC} -shared \${pic_flag} \${libobjs} \${deplibs} \${compiler_flags}
\${wl}-soname \${wl}\${soname} \${wl}-version-script
\${wl}\${output_objdir}/\${libname.ver} -o \${lib}

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=""
module_expsym_cmds=""

# Whether we are building with GNU ld or not.
with_gnu_ld="yes"

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=""

# Flag that enforces no undefined symbols.
no_undefined_flag=""

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec="\${wl}-rpath \${wl}\${libdir}"

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=""

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=no

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=no

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=no

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=unsupported

```

```
# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=no

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=no

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=unknown

# Set to "yes" if exported symbols are required.
always_export_symbols=no

# The commands to list exported symbols.
export_symbols_cmds="\$NM \$libobjs \$convenience |
\$global_symbol_pipe | \$SED 's/.* //' | sort | uniq >
\$export_symbols"

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms="_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*"

# Symbols that must always be exported.
include_expsyms=""

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=""

# Commands necessary for finishing linking programs.
postlink_cmds=""

# Specify filename containing input files.
file_list_spec=""

# How to hardcode a shared library path into an executable.
hardcode_action=immediate

# ### END LIBTOOL CONFIG

# libtool (GNU libtool) 2.4.2
# Written by Gordon Matzigkeit <gord@gnu.ai.mit.edu>, 1996

# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004, 2005,
2006,
# 2007, 2008, 2009, 2010, 2011 Free Software Foundation, Inc.
# This is free software; see the source for copying conditions.  There
is NO
```

```
# warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.

# GNU Libtool is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING.  If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html,
# or obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

# Usage: $progname [OPTION]... [MODE-ARG]...
#
# Provide generalized library-building support services.
#
#       --config                show all configuration variables
#       --debug                 enable verbose shell tracing
#   -n, --dry-run               display commands without modifying any
files
#       --features              display basic configuration information
and exit
#       --mode=MODE             use operation mode MODE
#       --preserve-dup-deps     don't remove duplicate dependency
libraries
#       --quiet, --silent       don't print informational messages
#       --no-quiet, --no-silent
#                               print informational messages (default)
#       --no-warn               don't display warning messages
#       --tag=TAG               use configuration variables from tag TAG
#   -v, --verbose               print more informational messages than
default
#       --no-verbose            don't print the extra informational
messages
#       --version               print version information
#   -h, --help, --help-all    print short, long, or detailed help
message
#
# MODE must be one of the following:
#
```

```

#       clean                remove files from the build directory
#       compile              compile a source file into a libtool
object
#       execute              automatically set library path, then run
a program
#       finish               complete the installation of libtool
libraries
#       install              install libraries or executables
#       link                 create a library or an executable
#       uninstall            remove libraries from an installed
directory
#
# MODE-ARGS vary depending on the MODE.  When passed as first option,
# `--mode=MODE' may be abbreviated as `MODE' or a unique abbreviation
of that.
# Try `$progname --help --mode=MODE' for a more detailed description
of MODE.
#
# When reporting a bug, please describe a test case to reproduce it
and
# include the following information:
#
#       host-triplet:       $host
#       shell:              $SHELL
#       compiler:           $LTCC
#       compiler flags:     $LTCFLAGS
#       linker:             $LD (gnu? $with_gnu_ld)
#       $progname:         (GNU libtool) 2.4.2
#       automake:           $automake_version
#       autoconf:          $autoconf_version
#
# Report bugs to <bug-libtool@gnu.org>.
# GNU libtool home page: <http://www.gnu.org/software/libtool/>.
# General help using GNU software: <http://www.gnu.org/gethelp/>.

PROGRAM=libtool
PACKAGE=libtool
VERSION=2.4.2
TIMESTAMP=""
package_revision=1.3337

# Be Bourne compatible
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
  emulate sh
  NULLCMD=:
  # Zsh 3.x and 4.x performs word splitting on ${1+"$@"}, which
  # is contrary to our usage.  Disable this feature.
  alias -g '${1+"$@"}'='"$@"'
  setopt NO_GLOB_SUBST
else
  case `(set -o) 2>/dev/null` in *posix*) set -o posix;; esac
fi

```

```

BIN_SH=xpg4; export BIN_SH # for Tru64
DUALCASE=1; export DUALCASE # for MKS sh

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
    eval 'cat <<_LTECHO_EOF'
$1
_LTECHO_EOF'
}

# NLS nuisances: We save the old values to restore during execute
mode.
lt_user_locale=
lt_safe_locale=
for lt_var in LANG LANGUAGE LC_ALL LC_CTYPE LC_COLLATE LC_MESSAGES
do
    eval "if test \"\${$lt_var+set}\" = set; then
        save_$lt_var=\${$lt_var}
        $lt_var=C
        export $lt_var
        lt_user_locale=\"\$lt_var=\\\$save_\$lt_var; \$lt_user_locale\"
        lt_safe_locale=\"\$lt_var=C; \$lt_safe_locale\"
    fi"
done
LC_ALL=C
LANGUAGE=C
export LANGUAGE LC_ALL

$lt_unset CDPATH

# Work around backward compatibility issue on IRIX 6.5. On IRIX 6.4+,
sh
# is ksh but when the shell is invoked as "sh" and the current value
of
# the _XPG environment variable is not equal to 1 (one), the special
# positional parameter $0, within a function call, is the name of the
# function.
progpach="$0"

: ${CP="cp -f"}
test "${ECHO+set}" = set || ECHO=${as_echo-'printf %s\n'}
: ${EGREP="egrep"}
: ${FGREP="fgrep"}
: ${GREP="grep"}
: ${LN_S="ln -s"}
: ${MAKE="make"}
: ${MKDIR="mkdir"}
: ${MV="mv -f"}

```

```

: ${RM="rm -f"}
: ${SED="sed"}
: ${SHELL="${CONFIG_SHELL-/bin/sh}"}
: ${Xsed="$SED -e 1s/^X//"}

# Global variables:
EXIT_SUCCESS=0
EXIT_FAILURE=1
EXIT_MISMATCH=63 # $? = 63 is used to indicate version mismatch to
missing.
EXIT_SKIP=77      # $? = 77 is used to indicate a skipped test to
automake.

exit_status=$EXIT_SUCCESS

# Make sure IFS has a sensible default
lt_nl='
'
IFS="      $lt_nl"

dirname="s,/[^/]*$,,"
basename="s,^.*/,,"

# func_dirname file append nondir_replacement
# Compute the dirname of FILE.  If nonempty, add APPEND to the result,
# otherwise set result to NONDIR_REPLACEMENT.
func_dirname ()
{
    case ${1} in
        /*) func_dirname_result="${1%/*}${2}" ;;
        *  ) func_dirname_result="${3}" ;;
    esac
} # Extended-shell func_dirname implementation

# func_basename file
func_basename ()
{
    func_basename_result="${1##*/}"
} # Extended-shell func_basename implementation

# func_dirname_and_basename file append nondir_replacement
# perform func_basename and func_dirname in a single function
# call:
#   dirname: Compute the dirname of FILE.  If nonempty,
#             add APPEND to the result, otherwise set result
#             to NONDIR_REPLACEMENT.
#             value returned in "$func_dirname_result"
#   basename: Compute filename of FILE.
#             value returned in "$func_basename_result"
# Implementation must be kept synchronized with func_dirname

```



```

# and func_basename. For efficiency, we do not delegate to
# those functions but instead duplicate the functionality here.
func_dirname_and_basename ()
{
    case ${1} in
        /*) func_dirname_result="${1%/*}"${2}" ;;
        * ) func_dirname_result="${3}" ;;
    esac
    func_basename_result="${1##*/}"
} # Extended-shell func_dirname_and_basename implementation

# func_stripname prefix suffix name
# strip PREFIX and SUFFIX off of NAME.
# PREFIX and SUFFIX must not contain globbing or regex special
# characters, hashes, percent signs, but SUFFIX may contain a leading
# dot (in which case that matches only a dot).
# func_strip_suffix prefix name
func_stripname ()
{
    # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are
    # positional parameters, so assign one to ordinary parameter
    first.
    func_stripname_result=${3}
    func_stripname_result=${func_stripname_result#"${1}" }
    func_stripname_result=${func_stripname_result%"${2}" }
} # Extended-shell func_stripname implementation

# These SED scripts presuppose an absolute path with a trailing slash.
pathcar='s,^\([^/]*\)$.*,\1,'
pathcdr='s,^[^/]*,, '
removedotparts=':dotsl
s@/\./@/g
t dotsl
s,/\.$/,/,'
collapseslashes='s@/\{1,\}@/g'
finalslash='s,/*$,/,,'

# func_normal_abspath PATH
# Remove doubled-up and trailing slashes, "." path components,
# and cancel out any ".." path components in PATH after making
# it an absolute path.
# value returned in "$func_normal_abspath_result"
func_normal_abspath ()
{
    # Start from root dir and reassemble the path.
    func_normal_abspath_result=
    func_normal_abspath_tpath=$1
    func_normal_abspath_altnamespace=
    case $func_normal_abspath_tpath in
        "")

```

```

# Empty path, that just means $cwd.
func_stripname '' '/' "`pwd`"
func_normal_abspath_result=$func_stripname_result
return

;;
# The next three entries are used to spot a run of precisely
# two leading slashes without using negated character classes;
# we take advantage of case's first-match behaviour.
/**)
# Unusual form of absolute path, do nothing.
;;
/**)
# Not necessarily an ordinary path; POSIX reserves leading '/'
# and for example Cygwin uses it to access remote file shares
# over CIFS/SMB, so we conserve a leading double slash if found.
func_normal_abspath_altnamespace=/
;;
/*)
# Absolute path, do nothing.
;;
*)
# Relative path, prepend $cwd.
func_normal_abspath_tpath=`pwd`/$func_normal_abspath_tpath
;;
esac
# Cancel out all the simple stuff to save iterations. We also want
# the path to end with a slash for ease of parsing, so make sure
# there is one (and only one) here.
func_normal_abspath_tpath=`$ECHO "$func_normal_abspath_tpath" | $SED
\
    -e "$removedotparts" -e "$collapseslashes" -e "$finalslash"`
while ;; do
# Processed it all yet?
if test "$func_normal_abspath_tpath" = / ; then
# If we ascended to the root using ".." the result may be empty
now.
if test -z "$func_normal_abspath_result" ; then
func_normal_abspath_result=/
fi
break
fi
func_normal_abspath_tcomponent=`$ECHO "$func_normal_abspath_tpath"
| $SED \
    -e "$pathcar"`
func_normal_abspath_tpath=`$ECHO "$func_normal_abspath_tpath" |
$SED \
    -e "$pathcdr"`
# Figure out what to do with it
case $func_normal_abspath_tcomponent in
"")
# Trailing empty path component, ignore it.
;;

```

```

..)
# Parent dir; strip last assembled component from result.
func_dirname "$func_normal_abspath_result"
func_normal_abspath_result=$func_dirname_result
;;
*)
# Actual path component, append it.

func_normal_abspath_result=$func_normal_abspath_result/$func_normal_ab
spath_tcomponent
;;
esac
done
# Restore leading double-slash if one was found on entry.

func_normal_abspath_result=$func_normal_abspath_altnamespace$func_norm
al_abspath_result
}

# func_relative_path SRCDIR DSTDIR
# generates a relative path from SRCDIR to DSTDIR, with a trailing
# slash if non-empty, suitable for immediately appending a filename
# without needing to append a separator.
# value returned in "$func_relative_path_result"
func_relative_path ()
{
func_relative_path_result=
func_normal_abspath "$1"
func_relative_path_tlibdir=$func_normal_abspath_result
func_normal_abspath "$2"
func_relative_path_tbindir=$func_normal_abspath_result

# Ascend the tree starting from libdir
while ;; do
# check if we have found a prefix of bindir
case $func_relative_path_tbindir in
$func_relative_path_tlibdir)
# found an exact match
func_relative_path_tcancelled=
break
;;
$func_relative_path_tlibdir*)
# found a matching prefix
func_stripname "$func_relative_path_tlibdir" ''
"$func_relative_path_tbindir"
func_relative_path_tcancelled=$func_stripname_result
if test -z "$func_relative_path_result"; then
func_relative_path_result=.
fi
break
;;
*)

```

```

    func_dirname $func_relative_path_tlibdir
    func_relative_path_tlibdir=${func_dirname_result}
    if test "x$func_relative_path_tlibdir" = x ; then
        # Have to descend all the way to the root!
        func_relative_path_result=../$func_relative_path_result
        func_relative_path_tcancelled=$func_relative_path_tbindir
        break
    fi
    func_relative_path_result=../$func_relative_path_result
    ;;
esac
done

# Now calculate path; take care to avoid doubling-up slashes.
func_stripname ' ' '/' "$func_relative_path_result"
func_relative_path_result=$func_stripname_result
func_stripname '/' '/' "$func_relative_path_tcancelled"
if test "x$func_stripname_result" != x ; then

func_relative_path_result=${func_relative_path_result}/${func_stripnam
e_result}
fi

# Normalisation. If bindir is libdir, return empty string,
# else relative path ending with a slash; either way, target
# file name can be directly appended.
if test ! -z "$func_relative_path_result"; then
    func_stripname './' '' "$func_relative_path_result/"
    func_relative_path_result=$func_stripname_result
fi
}

# The name of this program:
func_dirname_and_basename "$progpdir"
progname=$func_basename_result

# Make sure we have an absolute path for reexecution:
case $progpdir in
    [\\/]*)
        ;;
    *)
        progdir=$(cd "$progpdir" && pwd)
        progpath="$progdir/$progname"
        ;;
*)
    save_IFS="$IFS"
    IFS=${PATH_SEPARATOR:-;}
    for progdir in $PATH; do
        IFS="$save_IFS"
        test -x "$progdir/$progname" && break
    done
    IFS="$save_IFS"

```

```

        test -n "$progdir" || progdir=`pwd`
        progpath="$progdir/$progname"
    ;;
esac

# Sed substitution that helps us do robust quoting.  It backslashifies
# metacharacters that are still active within double-quoted strings.
Xsed="${SED}" -e 1s/^X//
sed_quote_subst='s/\(["$\\])/\|/g'

# Same as above, but do not quote variable references.
double_quote_subst='s/\(["\\])/\|/g'

# Sed substitution that turns a string into a regex matching for the
# string literally.
sed_make_literal_regex='s,[].[^$\\*\/],\|&,g'

# Sed substitution that converts a w32 file name or path
# which contains forward slashes, into one that contains
# (escaped) backslashes.  A very naive implementation.
lt_sed_naive_backslashify='s|/|\\|g;s|\\|/|g;s|/|/|g;s|\\|\\|g'

# Re-`\' parameter expansions in output of double_quote_subst that
# were
# `\'-ed in input to the same.  If an odd number of `\' preceded a '$'
# in input to double_quote_subst, that '$' was protected from
# expansion.
# Since each input `\' is now two `\'s, look for any number of runs of
# four `\'s followed by two `\'s and then a '$'.  `\' that '$'.
bs=`\'
bs2=`\|
bs4=`\|
dollar=`\$'
sed_double_backslash="\
s/$bs4/&
/g
s/^$bs2$dollar/$bs&/
s/\\([^\$bs]\\)$bs2$dollar/\\1$bs2$bs$dollar/g
s/\\n//g"

# Standard options:
opt_dry_run=false
opt_help=false
opt_quiet=false
opt_verbose=false
opt_warning=:

# func_echo arg...
# Echo program name prefixed message, along with the current mode
# name if it has been set yet.
func_echo ()
{

```

```

    $ECHO "$progname: ${opt_mode+$opt_mode: }$*"
}

# func_verbose arg...
# Echo program name prefixed message in verbose mode only.
func_verbose ()
{
    $opt_verbose && func_echo "${1+"$@"}

    # A bug in bash halts the script if the last line of a function
    # fails when set -e is in force, so we need another command to
    # work around that:
    :
}

# func_echo_all arg...
# Invoke $ECHO with all args, space-separated.
func_echo_all ()
{
    $ECHO "$*"
}

# func_error arg...
# Echo program name prefixed message to standard error.
func_error ()
{
    $ECHO "$progname: ${opt_mode+$opt_mode: }"${1+"$@"} 1>&2
}

# func_warning arg...
# Echo program name prefixed warning message to standard error.
func_warning ()
{
    $opt_warning && $ECHO "$progname: ${opt_mode+$opt_mode: }warning:
"${1+"$@"} 1>&2

    # bash bug again:
    :
}

# func_fatal_error arg...
# Echo program name prefixed message to standard error, and exit.
func_fatal_error ()
{
    func_error "${1+"$@"}
    exit $EXIT_FAILURE
}

# func_fatal_help arg...
# Echo program name prefixed message to standard error, followed by
# a help hint, and exit.
func_fatal_help ()

```

```

{
    func_error ${1+"$@"}
    func_fatal_error "$help"
}
help="Try \`${progname} --help' for more information." ## default

# func_grep expression filename
# Check whether EXPRESSION matches any line of FILENAME, without
output.
func_grep ()
{
    $GREP "$1" "$2" >/dev/null 2>&1
}

# func_mkdir_p directory-path
# Make sure the entire path to DIRECTORY-PATH is available.
func_mkdir_p ()
{
    my_directory_path="$1"
    my_dir_list=

    if test -n "$my_directory_path" && test "$opt_dry_run" != ":";
then
        # Protect directory names starting with '-'
        case $my_directory_path in
            -*) my_directory_path="./$my_directory_path" ;;
        esac

        # While some portion of DIR does not yet exist...
        while test ! -d "$my_directory_path"; do
            # ...make a list in topmost first order. Use a colon
delimited
            # list incase some portion of path contains whitespace.
            my_dir_list="$my_directory_path:$my_dir_list"

            # If the last portion added has no slash in it, the list is
done
            case $my_directory_path in */*) ;; *) break ;; esac

            # ...otherwise throw away the child directory and loop
            my_directory_path=`$ECHO "$my_directory_path" | $SED -e
"$dirname" `
            done
            my_dir_list=`$ECHO "$my_dir_list" | $SED 's,:*$,,' `

            save_mkdir_p_IFS="$IFS"; IFS=':'
            for my_dir in $my_dir_list; do
                IFS="$save_mkdir_p_IFS"
                # mkdir can fail with a `File exist' error if two processes

```

```

        # try to create one of the directories concurrently.  Don't
        # stop in that case!
        $MKDIR "$my_dir" 2>/dev/null || :
done
IFS="$save_mkdir_p_IFS"

# Bail out if we (or some other process) failed to create a
directory.
test -d "$my_directory_path" || \
    func_fatal_error "Failed to create \`$1'"
fi
}

# func_mktempdir [string]
# Make a temporary directory that won't clash with other running
# libtool processes, and avoids race conditions if possible.  If
# given, STRING is the basename for that directory.
func_mktempdir ()
{
    my_template="${TMPDIR-/tmp}/${1-$progname}"

    if test "$opt_dry_run" = ":"; then
        # Return a directory name, but don't create it in dry-run mode
        my_tmpdir="${my_template}--$$"
    else

        # If mktemp works, use that first and foremost
        my_tmpdir=`mktemp -d "${my_template}-XXXXXXXX" 2>/dev/null`

        if test ! -d "$my_tmpdir"; then
            # Failing that, at least try and use $RANDOM to avoid a race
            my_tmpdir="${my_template}-${RANDOM-0}$$"

            save_mktempdir_umask=`umask`
            umask 0077
            $MKDIR "$my_tmpdir"
            umask $save_mktempdir_umask
        fi

        # If we're not in dry-run mode, bomb out on failure
        test -d "$my_tmpdir" || \
            func_fatal_error "cannot create temporary directory
\`$my_tmpdir'"
        fi

        $ECHO "$my_tmpdir"
    }

# func_quote_for_eval arg
# Aesthetically quote ARG to be evaled later.

```



```

# This function returns two values: FUNC_QUOTE_FOR_EVAL_RESULT
# is double-quoted, suitable for a subsequent eval, whereas
# FUNC_QUOTE_FOR_EVAL_UNQUOTED_RESULT has merely all characters
# which are still active within double quotes backslashified.
func_quote_for_eval ()
{
    case $1 in
        *[\`\"$]*)
            func_quote_for_eval_unquoted_result=`$ECHO "$1" | $SED
"$sed_quote_subst"` ;;
        *)
            func_quote_for_eval_unquoted_result="$1" ;;
    esac

    case $func_quote_for_eval_unquoted_result in
        # Double-quote args containing shell metacharacters to delay
        # word splitting, command substitution and and variable
        # expansion for a subsequent eval.
        # Many Bourne shells cannot handle close brackets correctly
        # in scan sets, so we specify it separately.
        *[[\~\#\^\&\*\(\)\{\}\|\|;\<\>\?\'\ \ ]*|*|*"")
            func_quote_for_eval_result="\"$func_quote_for_eval_unquoted_result\"";;
        *)
            func_quote_for_eval_result="$func_quote_for_eval_unquoted_result"
            esac
    }

# func_quote_for_expand arg
# Aesthetically quote ARG to be eval'd later; same as above,
# but do not quote variable references.
func_quote_for_expand ()
{
    case $1 in
        *[\`\"$]*)
            my_arg=`$ECHO "$1" | $SED \
            -e "$double_quote_subst" -e "$sed_double_backslash"` ;;
        *)
            my_arg="$1" ;;
    esac

    case $my_arg in
        # Double-quote args containing shell metacharacters to delay
        # word splitting and command substitution for a subsequent eval.
        # Many Bourne shells cannot handle close brackets correctly
        # in scan sets, so we specify it separately.
        *[[\~\#\^\&\*\(\)\{\}\|\|;\<\>\?\'\ \ ]*|*|*"")
            my_arg="\"$my_arg\"";;
    esac
}

```

```

    esac

    func_quote_for_expand_result="$my_arg"
}

# func_show_eval cmd [fail_exp]
# Unless opt_silent is true, then output CMD. Then, if opt_dryrun is
# not true, evaluate CMD. If the evaluation of CMD fails, and
FAIL_EXP
# is given, then evaluate it.
func_show_eval ()
{
    my_cmd="$1"
    my_fail_exp="${2-:}"

    ${opt_silent-false} || {
        func_quote_for_expand "$my_cmd"
        eval "func_echo $func_quote_for_expand_result"
    }

    if ${opt_dry_run-false}; then ;; else
        eval "$my_cmd"
        my_status=$?
        if test "$my_status" -eq 0; then ;; else
            eval "(exit $my_status); $my_fail_exp"
        fi
    fi
}

# func_show_eval_locale cmd [fail_exp]
# Unless opt_silent is true, then output CMD. Then, if opt_dryrun is
# not true, evaluate CMD. If the evaluation of CMD fails, and
FAIL_EXP
# is given, then evaluate it. Use the saved locale for evaluation.
func_show_eval_locale ()
{
    my_cmd="$1"
    my_fail_exp="${2-:}"

    ${opt_silent-false} || {
        func_quote_for_expand "$my_cmd"
        eval "func_echo $func_quote_for_expand_result"
    }

    if ${opt_dry_run-false}; then ;; else
        eval "$lt_user_locale
            $my_cmd"
        my_status=$?
        eval "$lt_safe_locale"
        if test "$my_status" -eq 0; then ;; else

```

```

        eval "(exit $my_status); $my_fail_exp"
    fi
fi
}

# func_tr_sh
# Turn $1 into a string suitable for a shell variable name.
# Result is stored in $func_tr_sh_result. All characters
# not in the set a-zA-Z0-9_ are replaced with '_'. Further,
# if $1 begins with a digit, a '_' is prepended as well.
func_tr_sh ()
{
    case $1 in
        [0-9]* | *[^a-zA-Z0-9_]*)
            func_tr_sh_result=`$ECHO "$1" | $SED 's/^\([0-9]\)/_\1/; s/[^a-zA-Z0-9_]/_/g'`
            ;;
        * )
            func_tr_sh_result=$1
            ;;
    esac
}

# func_version
# Echo version message to standard output and exit.
func_version ()
{
    $opt_debug

    $SED -n '/(C)/!b go
:more
/\./!{
    N
    s/\n# / /
    b more
}
:go
/^# '$PROGRAM' (GNU /,/# warranty; / {
    s/^# //
s/^# *$//
    s/\((C)\) [ 0-9,-]*\ ( [1-9][0-9]*\)/\1\2/
    p
}' < "$progpath"
exit $?
}

# func_usage
# Echo short help message to standard output and exit.
func_usage ()
{
    $opt_debug

```

```

$SED -n '/^# Usage:/,/^# *.*--help/ {
    s/^# //
    s/^# *$//
    s/\$progname/'$progname'/
    p
}' < "$progpath"
echo
$ECHO "run \"\$progname --help | more' for full usage"
exit $?
}

# func_help [NOEXIT]
# Echo long help message to standard output and exit,
# unless 'noexit' is passed as argument.
func_help ()
{
    $opt_debug

    $SED -n '/^# Usage:/,/# Report bugs to/ {
        :print
        s/^# //
        s/^# *$//
        s*\$progname*\$progname'*
        s*\$host*'"$host"'*
        s*\$SHELL*'"$SHELL"'*
        s*\$LTCC*'"$LTCC"'*
        s*\$LTCFLAGS*'"$LTCFLAGS"'*
        s*\$LD*'"$LD"'*
        s/\$with_gnu_ld/'"$with_gnu_ld"'/
        s/\$automake_version/'"$({AUTOMAKE-automake} --version)
2>/dev/null |$SED 1q`"/
        s/\$autoconf_version/'"$({AUTOCONF-autoconf} --version)
2>/dev/null |$SED 1q`"/
        p
        d
    }
    /^# .* home page:/b print
    /^# General help using/b print
    ' < "$progpath"
    ret=$?
    if test -z "$1"; then
        exit $ret
    fi
}

# func_missing_arg argname
# Echo program name prefixed message to standard error and set global
# exit_cmd.
func_missing_arg ()
{
    $opt_debug

```

```

    func_error "missing argument for $1."
    exit_cmd=exit
}

# func_split_short_opt shortopt
# Set func_split_short_opt_name and func_split_short_opt_arg shell
# variables after splitting SHORTOPT after the 2nd character.
func_split_short_opt ()
{
    func_split_short_opt_arg=${1#??}
    func_split_short_opt_name=${1%"$func_split_short_opt_arg"}
} # Extended-shell func_split_short_opt implementation

# func_split_long_opt longopt
# Set func_split_long_opt_name and func_split_long_opt_arg shell
# variables after splitting LONGOPT at the '=' sign.
func_split_long_opt ()
{
    func_split_long_opt_name=${1%=*}
    func_split_long_opt_arg=${1#*=}
} # Extended-shell func_split_long_opt implementation

exit_cmd=:

magic="%%MAGIC variable%%"
magic_exe="%%MAGIC EXE variable%%"

# Global variables.
nonopt=
preserve_args=
lo2o="s/\\.lo\\$/.${objext}/"
o2lo="s/\\.\\${objext}\\$/.lo/"
extracted_archives=
extracted_serial=0

# If this variable is set in any of the actions, the command in it
# will be execed at the end. This prevents here-documents from being
# left over by shells.
exec_cmd=

# func_append var value
# Append VALUE to the end of shell variable VAR.
func_append ()
{
    eval "${1}+=\${2}"
}

```

```

} # Extended-shell func_append implementation

# func_append_quoted var value
# Quote VALUE and append to the end of shell variable VAR, separated
# by a space.
func_append_quoted ()
{
    func_quote_for_eval "${2}"
    eval "${1}+=\\`\\`\\$func_quote_for_eval_result"
} # Extended-shell func_append_quoted implementation

# func_arith arithmetic-term...
func_arith ()
{
    func_arith_result=$(( $* ))
} # Extended-shell func_arith implementation

# func_len string
# STRING may not start with a hyphen.
func_len ()
{
    func_len_result=${#1}
} # Extended-shell func_len implementation

# func_lo2o object
func_lo2o ()
{
    case ${1} in
        *.lo) func_lo2o_result=${1%.lo}.${objext} ;;
        *)    func_lo2o_result=${1} ;;
    esac
} # Extended-shell func_lo2o implementation

# func_xform libobj-or-source
func_xform ()
{
    func_xform_result=${1%.*}.lo
} # Extended-shell func_xform implementation

# func_fatal_configuration arg...
# Echo program name prefixed message to standard error, followed by
# a configuration failure hint, and exit.
func_fatal_configuration ()
{
    func_error ${1+"$@"}
    func_error "See the $PACKAGE documentation for more information."
    func_fatal_error "Fatal configuration error."
}

```

```

}

# func_config
# Display the configuration for all the tags in this script.
func_config ()
{
    re_begincf='^# ### BEGIN LIBTOOL'
    re_endcf='^# ### END LIBTOOL'

    # Default configuration.
    $SED "1,/$re_begincf CONFIG/d;/$re_endcf CONFIG/,\`$d" <
"$proppath"

    # Now print the configurations for the tags.
    for tagname in $taglist; do
        $SED -n "/$re_begincf TAG CONFIG: $tagname\`$/,$re_endcf TAG
CONFIG: $tagname\`$/p" < "$proppath"
    done

    exit $?
}

# func_features
# Display the features supported by this script.
func_features ()
{
    echo "host: $host"
    if test "$build_libtool_libs" = yes; then
        echo "enable shared libraries"
    else
        echo "disable shared libraries"
    fi
    if test "$build_old_libs" = yes; then
        echo "enable static libraries"
    else
        echo "disable static libraries"
    fi

    exit $?
}

# func_enable_tag tagname
# Verify that TAGNAME is valid, and either flag an error and exit, or
# enable the TAGNAME tag. We also add TAGNAME to the global $taglist
# variable here.
func_enable_tag ()
{
    # Global variable:
    tagname="$1"

    re_begincf="^# ### BEGIN LIBTOOL TAG CONFIG: $tagname\`$"

```

```

re_endcf="^# ### END LIBTOOL TAG CONFIG: $tagname\$"
sed_extractcf="/$re_begincf/,/$re_endcf/p"

# Validate tagname.
case $tagname in
  *[_A-Za-z0-9,/]*)
    func_fatal_error "invalid tag name: $tagname"
    ;;
esac

# Don't test for the "default" C tag, as we know it's
# there but not specially marked.
case $tagname in
  CC) ;;
  *)
    if $GREP "$re_begincf" "$progp" >/dev/null 2>&1; then
      taglist="$taglist $tagname"

      # Evaluate the configuration.  Be careful to quote the path
      # and the sed script, to avoid splitting on whitespace, but
      # also don't use non-portable quotes within backquotes within
      # quotes we have to do it in 2 steps:
      extractedcf=`$SED -n -e "$sed_extractcf" < "$progp" `
      eval "$extractedcf"
    else
      func_error "ignoring unknown tag $tagname"
    fi
    ;;
esac
}

# func_check_version_match
# Ensure that we are using m4 macros, and libtool script from the same
# release of libtool.
func_check_version_match ()
{
  if test "$package_revision" != "$macro_revision"; then
    if test "$VERSION" != "$macro_version"; then
      if test -z "$macro_version"; then
        cat >&2 <<_LT_EOF
$progname: Version mismatch error.  This is $PACKAGE $VERSION, but the
$progname: definition of this LT_INIT comes from an older release.
$progname: You should recreate aclocal.m4 with macros from $PACKAGE
$VERSION
$progname: and run autoconf again.
_LT_EOF
      else
        cat >&2 <<_LT_EOF
$progname: Version mismatch error.  This is $PACKAGE $VERSION, but the
$progname: definition of this LT_INIT comes from $PACKAGE
$macro_version.

```



```
$progname: You should recreate aclocal.m4 with macros from $PACKAGE
$VERSION
```

```
$progname: and run autoconf again.
```

```
_LT_EOF
```

```
fi
```

```
else
```

```
cat >&2 <<_LT_EOF
```

```
$progname: Version mismatch error. This is $PACKAGE $VERSION,
revision $package_revision,
```

```
$progname: but the definition of this LT_INIT comes from revision
$macro_revision.
```

```
$progname: You should recreate aclocal.m4 with macros from revision
$package_revision
```

```
$progname: of $PACKAGE $VERSION and run autoconf again.
```

```
_LT_EOF
```

```
fi
```

```
exit $EXIT_MISMATCH
```

```
fi
```

```
}
```

```
# Shorthand for --mode=foo, only valid as the first argument
case $1 in
```

```
clean|clea|cle|cl)
```

```
shift; set dummy --mode clean ${1+"$@"}; shift
```

```
;;
```

```
compile|compil|compi|comp|com|co|c)
```

```
shift; set dummy --mode compile ${1+"$@"}; shift
```

```
;;
```

```
execute|execut|execu|exec|exe|ex|e)
```

```
shift; set dummy --mode execute ${1+"$@"}; shift
```

```
;;
```

```
finish|finis|fini|fin|fi|f)
```

```
shift; set dummy --mode finish ${1+"$@"}; shift
```

```
;;
```

```
install|instal|insta|inst|ins|in|i)
```

```
shift; set dummy --mode install ${1+"$@"}; shift
```

```
;;
```

```
link|lin|li|l)
```

```
shift; set dummy --mode link ${1+"$@"}; shift
```

```
;;
```

```
uninstall|uninstal|uninsta|uninst|unins|unin|uni|un|u)
```

```
shift; set dummy --mode uninstall ${1+"$@"}; shift
```

```
;;
```

```
esac
```

```
# Option defaults:
```

```
opt_debug=:
```

```
opt_dry_run=false
```

```

opt_config=false
opt_preserve_dup_deps=false
opt_features=false
opt_finish=false
opt_help=false
opt_help_all=false
opt_silent=:
opt_warning=:
opt_verbose=:
opt_silent=false
opt_verbose=false

# Parse options once, thoroughly.  This comes as soon as possible in
the
# script to make things like `--version' happen as quickly as we can.
{
  # this just eases exit handling
  while test $# -gt 0; do
    opt="$1"
    shift
    case $opt in
      --debug|-x)      opt_debug='set -x'
                      func_echo "enabling shell trace mode"
                      $opt_debug
                      ;;
      --dry-run|--dryrun|-n)
                      opt_dry_run=:
                      ;;
      --config)
                      opt_config=:
func_config
                      ;;
      --dlopen|-dlopen)
                      optarg="$1"
                      opt_dlopen="${opt_dlopen+$opt_dlopen
}$optarg"
                      shift
                      ;;
      --preserve-dup-deps)
                      opt_preserve_dup_deps=:
                      ;;
      --features)
                      opt_features=:
func_features
                      ;;
      --finish)
                      opt_finish=:
set dummy --mode finish ${1+"$@"}; shift
                      ;;
      --help)
                      opt_help=:

```

```

        ;;
    --help-all)
        opt_help_all=:
opt_help=': help-all'
        ;;
    --mode)
        test $# = 0 && func_missing_arg $opt && break
        optarg="$1"
        opt_mode="$optarg"
case $optarg in
    # Valid mode arguments:
    clean|compile|execute|finish|install|link|relink|uninstall) ;;

    # Catch anything else as an error
    *) func_error "invalid argument for $opt"
        exit_cmd=exit
        break
        ;;
esac

        shift
        ;;
    --no-silent|--no-quiet)
        opt_silent=false
preserve_args+=" $opt"
        ;;
    --no-warning|--no-warn)
        opt_warning=false
preserve_args+=" $opt"
        ;;
    --no-verbose)
        opt_verbose=false
preserve_args+=" $opt"
        ;;
    --silent|--quiet)
        opt_silent=:
preserve_args+=" $opt"
        opt_verbose=false
        ;;
    --verbose|-v)
        opt_verbose=:
preserve_args+=" $opt"
opt_silent=false
        ;;
    --tag)
        test $# = 0 && func_missing_arg $opt && break
        optarg="$1"
        opt_tag="$optarg"
preserve_args+=" $opt $optarg"
func_enable_tag "$optarg"
        shift
        ;;

```

```

-\?|-h)          func_usage          ;;
--help)          func_help           ;;
--version) func_version              ;;

# Separate optargs to long options:
--*=*)
    func_split_long_opt "$opt"
    set dummy "$func_split_long_opt_name"
"$func_split_long_opt_arg" ${1+"$@"}
    shift
    ;;

# Separate non-argument short options:
-\?*|-h*|-n*|-v*)
    func_split_short_opt "$opt"
    set dummy "$func_split_short_opt_name" "-"
"$func_split_short_opt_arg" ${1+"$@"}
    shift
    ;;

--)          break          ;;
-*)          func_fatal_help "unrecognized option \`${opt}'" ;;
*)          set dummy "$opt" ${1+"$@"}; shift; break ;;
esac
done

# Validate options:

# save first non-option argument
if test "$#" -gt 0; then
    nonopt="$opt"
    shift
fi

# preserve --debug
test "$opt_debug" = : || preserve_args+=" --debug"

case $host in
    *cygwin* | *mingw* | *pw32* | *cegcc*)
        # don't eliminate duplications in $postdeps and $predeps
        opt_duplicate_compiler_generated_deps=:
        ;;
    *)
        opt_duplicate_compiler_generated_deps=$opt_preserve_dup_deps
        ;;
esac

$opt_help || {
    # Sanity checks first:
    func_check_version_match

```

```

    if test "$build_libtool_libs" != yes && test "$build_old_libs" !=
yes; then
        func_fatal_configuration "not configured to build any kind of
library"
    fi

    # Darwin sucks
    eval std_shrext="\$shrext_cmds\"

    # Only execute mode is allowed to have -dlopen flags.
    if test -n "$opt_dlopen" && test "$opt_mode" != execute; then
        func_error "unrecognized option \`-dlopen'"
        $ECHO "$help" 1>&2
        exit $EXIT_FAILURE
    fi

    # Change the help message to a mode-specific one.
    generic_help="$help"
    help="Try \`$progname --help --mode=$opt_mode' for more
information."
}

# Bail if the options were screwed
$exit_cmd $EXIT_FAILURE
}

## ----- ##
##   Main.   ##
## ----- ##

# func_lalib_p file
# True iff FILE is a libtool `.la' library or `.lo' object file.
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_lalib_p ()
{
    test -f "$1" &&
        $SED -e 4q "$1" 2>/dev/null \
            | $GREP "^# Generated by .*$PACKAGE" > /dev/null 2>&1
}

# func_lalib_unsafe_p file
# True iff FILE is a libtool `.la' library or `.lo' object file.
# This function implements the same check as func_lalib_p without
# resorting to external programs. To this end, it redirects stdin and
# closes it afterwards, without saving the original file descriptor.
# As a safety measure, use it only where a negative result would be
# fatal anyway. Works if `file' does not exist.

```

```

func_lalib_unsafe_p ()
{
    lalib_p=no
    if test -f "$1" && test -r "$1" && exec 5<&0 <"$1"; then
        for lalib_p_1 in 1 2 3 4
        do
            read lalib_p_line
            case "$lalib_p_line" in
                \#\ Generated\ by\ *$PACKAGE* ) lalib_p=yes; break;;
            esac
        done
        exec 0<&5 5<&-
    fi
    test "$lalib_p" = yes
}

# func_ltwrapper_script_p file
# True iff FILE is a libtool wrapper script
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_script_p ()
{
    func_lalib_p "$1"
}

# func_ltwrapper_executable_p file
# True iff FILE is a libtool wrapper executable
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_executable_p ()
{
    func_ltwrapper_exec_suffix=
    case $1 in
        *.exe) ;;
        *) func_ltwrapper_exec_suffix=.exe ;;
    esac
    $GREP "$magic_exe" "$1$func_ltwrapper_exec_suffix" >/dev/null 2>&1
}

# func_ltwrapper_scriptname file
# Assumes file is an ltwrapper_executable
# uses $file to determine the appropriate filename for a
# temporary ltwrapper_script.
func_ltwrapper_scriptname ()
{
    func_dirname_and_basename "$1" "" "."
    func_stripname ' ' '.exe' "$func_basename_result"

    func_ltwrapper_scriptname_result="$func_dirname_result/$objdir/${func_
stripname_result}_ltshwrapper"
}

```

```

# func_ltwrapper_p file
# True iff FILE is a libtool wrapper script or wrapper executable
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_p ()
{
    func_ltwrapper_script_p "$1" || func_ltwrapper_executable_p "$1"
}

# func_execute_cmds commands fail_cmd
# Execute tilde-delimited COMMANDS.
# If FAIL_CMD is given, eval that upon failure.
# FAIL_CMD may read-access the current command in variable CMD!
func_execute_cmds ()
{
    $opt_debug
    save_ifs=$IFS; IFS='~'
    for cmd in $1; do
        IFS=$save_ifs
        eval cmd="\$cmd\"
        func_show_eval "$cmd" "${2-:}"
    done
    IFS=$save_ifs
}

# func_source file
# Source FILE, adding directory component if necessary.
# Note that it is not necessary on cygwin/mingw to append a dot to
# FILE even if both FILE and FILE.exe exist: automatic-append-.exe
# behavior happens only for exec(3), not for open(2)! Also, sourcing
# `FILE.' does not work on cygwin managed mounts.
func_source ()
{
    $opt_debug
    case $1 in
        /* | *\*) . "$1" ;;
        *) . "./$1" ;;
    esac
}

# func_resolve_sysroot PATH
# Replace a leading = in PATH with a sysroot. Store the result into
# func_resolve_sysroot_result
func_resolve_sysroot ()
{
    func_resolve_sysroot_result=$1
    case $func_resolve_sysroot_result in
    =*)
        func_stripname '=' '' "$func_resolve_sysroot_result"

```

```

    func_resolve_sysroot_result=${lt_sysroot}${func_stripname_result}
    ;;
esac
}

# func_replace_sysroot PATH
# If PATH begins with the sysroot, replace it with = and
# store the result into func_replace_sysroot_result.
func_replace_sysroot ()
{
    case "$lt_sysroot:$1" in
        ?*:"$lt_sysroot"*)
            func_stripname "$lt_sysroot" '' "$1"
            func_replace_sysroot_result="=${func_stripname_result}"
            ;;
        *)
            # Including no sysroot.
            func_replace_sysroot_result=$1
            ;;
    esac
}

# func_infer_tag arg
# Infer tagged configuration to use if any are available and
# if one wasn't chosen via the "--tag" command line option.
# Only attempt this if the compiler in the base compile
# command doesn't match the default compiler.
# arg is usually of the form 'gcc ...'
func_infer_tag ()
{
    $opt_debug
    if test -n "$available_tags" && test -z "$tagname"; then
        CC_quoted=
        for arg in $CC; do
            func_append_quoted CC_quoted "$arg"
        done
        CC_expanded=`func_echo_all $CC`
        CC_quoted_expanded=`func_echo_all $CC_quoted`
        case $@ in
            # Blanks in the command may have been stripped by the calling
shell,
            # but not from the CC environment variable when configure was
run.
            " $CC "*" | "$CC "*" | " $CC_expanded "*" | "$CC_expanded "*" | \
            " $CC_quoted"*" | "$CC_quoted "*" | " $CC_quoted_expanded "*" |
"$CC_quoted_expanded "*" ) ;;
            # Blanks at the start of $base_compile will cause this to fail
            # if we don't check for them as well.
            *)
                for z in $available_tags; do
                    if $GREP "^# ### BEGIN LIBTOOL TAG CONFIG: $z$" < "$progp" >
/dev/null; then

```



```

    # Evaluate the configuration.
    eval "`${SED} -n -e '/^# ### BEGIN LIBTOOL TAG CONFIG:
'$z'$$/,/^# ### END LIBTOOL TAG CONFIG: '$z'$$/p' < $progp`"
    CC_quoted=
    for arg in $CC; do
        # Double-quote args containing other shell metacharacters.
        func_append_quoted CC_quoted "$arg"
    done
    CC_expanded=`func_echo_all $CC`
    CC_quoted_expanded=`func_echo_all $CC_quoted`
    case "$@" in
        "$CC" "*" | "$CC" "*" | " $CC_expanded" "*" | "$CC_expanded" "*" | \
        " $CC_quoted" "*" | "$CC_quoted" "*" | " $CC_quoted_expanded" "*" | \
"$CC_quoted_expanded" *)
        # The compiler in the base compile command matches
        # the one in the tagged configuration.
        # Assume this is the tagged configuration we want.
        tagname=$z
        break
        ;;
    esac
    fi
done
# If $tagname still isn't set, then no tagged configuration
# was found and let the user know that the "--tag" command
# line option must be used.
if test -z "$tagname"; then
    func_echo "unable to infer tagged configuration"
    func_fatal_error "specify a tag with `--tag'"
# else
#     func_verbose "using $tagname tagged configuration"
fi
;;
esac
fi
}

```

```

# func_write_libtool_object output_name pic_name nonpic_name
# Create a libtool object file (analogous to a ".la" file),
# but don't create it if we're doing a dry run.
func_write_libtool_object ()
{
    write_libobj=${1}
    if test "$build_libtool_libs" = yes; then
        write_lobj=`'$2`\'
    else
        write_lobj=none
    fi

    if test "$build_old_libs" = yes; then

```

```

        write_oldobj=\'${3}\'
    else
        write_oldobj=none
    fi

    $opt_dry_run || {
        cat >${write_libobj}T <<EOF
# $write_libobj - a libtool object file
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION
#
# Please DO NOT delete this file!
# It is necessary for linking the library.

# Name of the PIC object.
pic_object=$write_lobj

# Name of the non-PIC object
non_pic_object=$write_oldobj

EOF
        $MV "${write_libobj}T" "${write_libobj}"
    }
}

#####
# FILE NAME AND PATH CONVERSION HELPER FUNCTIONS #
#####

# func_convert_core_file_wine_to_w32 ARG
# Helper function used by file name conversion functions when $build
# is *nix,
# and $host is mingw, cygwin, or some other w32 environment. Relies on
# a
# correctly configured wine environment available, with the winepath
# program
# in $build's $PATH.
#
# ARG is the $build file name to be converted to w32 format.
# Result is available in $func_convert_core_file_wine_to_w32_result,
# and will
# be empty on error (or when ARG is empty)
func_convert_core_file_wine_to_w32 ()
{
    $opt_debug
    func_convert_core_file_wine_to_w32_result="$1"
    if test -n "$1"; then
        # Unfortunately, winepath does not exit with a non-zero error
        # code, so we
        # are forced to check the contents of stdout. On the other hand,
        # if the

```

```

    # command is not found, the shell will set an exit code of 127 and
print
    # *an error message* to stdout. So we must check for both error
code of
    # zero AND non-empty stdout, which explains the odd construction:
    func_convert_core_file_wine_to_w32_tmp=`winepath -w "$1"
2>/dev/null`
    if test "$?" -eq 0 && test -n
"${func_convert_core_file_wine_to_w32_tmp}"; then
        func_convert_core_file_wine_to_w32_result=`$ECHO
"$func_convert_core_file_wine_to_w32_tmp" |
        $SED -e "$lt_sed_naive_backslashify"`
    else
        func_convert_core_file_wine_to_w32_result=
    fi
fi
}
# end: func_convert_core_file_wine_to_w32

# func_convert_core_path_wine_to_w32 ARG
# Helper function used by path conversion functions when $build is
*nix, and
# $host is mingw, cygwin, or some other w32 environment. Relies on a
correctly
# configured wine environment available, with the winepath program in
$build's
# $PATH. Assumes ARG has no leading or trailing path separator
characters.
#
# ARG is path to be converted from $build format to win32.
# Result is available in $func_convert_core_path_wine_to_w32_result.
# Unconvertible file (directory) names in ARG are skipped; if no
directory names
# are convertible, then the result may be empty.
func_convert_core_path_wine_to_w32 ()
{
    $opt_debug
    # unfortunately, winepath doesn't convert paths, only file names
    func_convert_core_path_wine_to_w32_result=""
    if test -n "$1"; then
        oldIFS=$IFS
        IFS=:
        for func_convert_core_path_wine_to_w32_f in $1; do
            IFS=$oldIFS
            func_convert_core_file_wine_to_w32
"$func_convert_core_path_wine_to_w32_f"
            if test -n "$func_convert_core_file_wine_to_w32_result" ; then
                if test -z "$func_convert_core_path_wine_to_w32_result"; then

func_convert_core_path_wine_to_w32_result="$func_convert_core_file_wine_to_w32_result"

```

```

        else
            func_append func_convert_core_path_wine_to_w32_result
";$func_convert_core_file_wine_to_w32_result"
            fi
        fi
    done
    IFS=$oldIFS
fi
}
# end: func_convert_core_path_wine_to_w32

# func_cygpath ARGS...
# Wrapper around calling the cygpath program via LT_CYGPATH. This is
used when
# when (1) $build is *nix and Cygwin is hosted via a wine environment;
or (2)
# $build is MSYS and $host is Cygwin, or (3) $build is Cygwin. In case
(1) or
# (2), returns the Cygwin file name or path in func_cygpath_result
(input
# file name or path is assumed to be in w32 format, as previously
converted
# from $build's *nix or MSYS format). In case (3), returns the w32
file name
# or path in func_cygpath_result (input file name or path is assumed
to be in
# Cygwin format). Returns an empty string on error.
#
# ARGS are passed to cygpath, with the last one being the file name or
path to
# be converted.
#
# Specify the absolute *nix (or w32) name to cygpath in the LT_CYGPATH
# environment variable; do not put it in $PATH.
func_cygpath ()
{
    $opt_debug
    if test -n "$LT_CYGPATH" && test -f "$LT_CYGPATH"; then
        func_cygpath_result=`$LT_CYGPATH "$@" 2>/dev/null`
        if test "$?" -ne 0; then
            # on failure, ensure result is empty
            func_cygpath_result=
        fi
    else
        func_cygpath_result=
        func_error "LT_CYGPATH is empty or specifies non-existent file:
\`$LT_CYGPATH'"
        fi
    }
#end: func_cygpath

```

```

# func_convert_core_msys_to_w32 ARG
# Convert file name or path ARG from MSYS format to w32 format.
Return
# result in func_convert_core_msys_to_w32_result.
func_convert_core_msys_to_w32 ()
{
    $opt_debug
    # awkward: cmd appends spaces to result
    func_convert_core_msys_to_w32_result=`( cmd //c echo "$1" )
2>/dev/null |
    $SED -e 's/[ ]*$//' -e "$!t_sed_naive_backslashify"`
}
#end: func_convert_core_msys_to_w32

# func_convert_file_check ARG1 ARG2
# Verify that ARG1 (a file name in $build format) was converted to
$host
# format in ARG2. Otherwise, emit an error message, but continue
(resetting
# func_to_host_file_result to ARG1).
func_convert_file_check ()
{
    $opt_debug
    if test -z "$2" && test -n "$1" ; then
        func_error "Could not determine host file name corresponding to"
        func_error "  \`$1'"
        func_error "Continuing, but uninstalled executables may not work."
        # Fallback:
        func_to_host_file_result="$1"
    fi
}
# end func_convert_file_check

# func_convert_path_check FROM_PATHSEP TO_PATHSEP FROM_PATH TO_PATH
# Verify that FROM_PATH (a path in $build format) was converted to
$host
# format in TO_PATH. Otherwise, emit an error message, but continue,
resetting
# func_to_host_file_result to a simplistic fallback value (see below).
func_convert_path_check ()
{
    $opt_debug
    if test -z "$4" && test -n "$3"; then
        func_error "Could not determine the host path corresponding to"
        func_error "  \`$3'"
        func_error "Continuing, but uninstalled executables may not work."
        # Fallback. This is a deliberately simplistic "conversion" and
        # should not be "improved". See libtool.info.
        if test "x$1" != "x$2"; then

```

```

        lt_replace_pathsep_chars="s|$1|$2|g"
        func_to_host_path_result=`echo "$3" |
            $SED -e "$lt_replace_pathsep_chars"`
    else
        func_to_host_path_result="$3"
    fi
fi
}
# end func_convert_path_check

# func_convert_path_front_back_pathsep FRONTPAT BACKPAT REPL ORIG
# Modifies func_to_host_path_result by prepending REPL if ORIG matches
FRONTPAT
# and appending REPL if ORIG matches BACKPAT.
func_convert_path_front_back_pathsep ()
{
    $opt_debug
    case $4 in
    $1 ) func_to_host_path_result="$3$func_to_host_path_result"
        ;;
    esac
    case $4 in
    $2 ) func_to_host_path_result+="$3"
        ;;
    esac
}
# end func_convert_path_front_back_pathsep

#####
# $build to $host FILE NAME CONVERSION FUNCTIONS #
#####
# invoked via `to_host_file_cmd ARG'
#
# In each case, ARG is the path to be converted from $build to $host
format.
# Result will be available in $func_to_host_file_result.

# func_to_host_file ARG
# Converts the file name ARG from $build format to $host format.
Return result
# in func_to_host_file_result.
func_to_host_file ()
{
    $opt_debug
    to_host_file_cmd "$1"
}
# end func_to_host_file

```

```

# func_to_tool_file ARG LAZY
# converts the file name ARG from $build format to toolchain format.
Return
# result in func_to_tool_file_result. If the conversion in use is
listed
# in (the comma separated) LAZY, no conversion takes place.
func_to_tool_file ()
{
    $opt_debug
    case , $2, in
        *, "$to_tool_file_cmd", *)
            func_to_tool_file_result=$1
            ;;
        *)
            $to_tool_file_cmd "$1"
            func_to_tool_file_result=$func_to_host_file_result
            ;;
    esac
}
# end func_to_tool_file

# func_convert_file_noop ARG
# Copy ARG to func_to_host_file_result.
func_convert_file_noop ()
{
    func_to_host_file_result="$1"
}
# end func_convert_file_noop

# func_convert_file_msys_to_w32 ARG
# Convert file name ARG from (mingw) MSYS to (mingw) w32 format;
automatic
# conversion to w32 is not available inside the cwrapper. Returns
result in
# func_to_host_file_result.
func_convert_file_msys_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_msys_to_w32 "$1"
        func_to_host_file_result="$func_convert_core_msys_to_w32_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_msys_to_w32

# func_convert_file_cygwin_to_w32 ARG
# Convert file name ARG from Cygwin to w32 format. Returns result in

```

```

# func_to_host_file_result.
func_convert_file_cygwin_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        # because $build is cygwin, we call "the" cygpath in $PATH; no
        # need to use
        # LT_CYGPATH in this case.
        func_to_host_file_result=`cygpath -m "$1"`
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_cygwin_to_w32

# func_convert_file_nix_to_w32 ARG
# Convert file name ARG from *nix to w32 format. Requires a wine
# environment
# and a working winepath. Returns result in func_to_host_file_result.
func_convert_file_nix_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_file_wine_to_w32 "$1"

        func_to_host_file_result="$func_convert_core_file_wine_to_w32_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_nix_to_w32

# func_convert_file_msys_to_cygwin ARG
# Convert file name ARG from MSYS to Cygwin format. Requires
# LT_CYGPATH set.
# Returns result in func_to_host_file_result.
func_convert_file_msys_to_cygwin ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_msys_to_w32 "$1"
        func_cygpath -u "$func_convert_core_msys_to_w32_result"
        func_to_host_file_result="$func_cygpath_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_msys_to_cygwin

```



```

# func_convert_file_nix_to_cygwin ARG
# Convert file name ARG from *nix to Cygwin format.  Requires Cygwin
installed
# in a wine environment, working winepath, and LT_CYGPATH set.
Returns result
# in func_to_host_file_result.
func_convert_file_nix_to_cygwin ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        # convert from *nix to w32, then use cygpath to convert from w32
to cygwin.
        func_convert_core_file_wine_to_w32 "$1"
        func_cygpath -u "$func_convert_core_file_wine_to_w32_result"
        func_to_host_file_result="$func_cygpath_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_nix_to_cygwin

#####
# $build to $host PATH CONVERSION FUNCTIONS #
#####
# invoked via `sto_host_path_cmd ARG'
#
# In each case, ARG is the path to be converted from $build to $host
format.
# The result will be available in $func_to_host_path_result.
#
# Path separators are also converted from $build format to $host
format.  If
# ARG begins or ends with a path separator character, it is preserved
(but
# converted to $host format) on output.
#
# All path conversion functions are named using the following
convention:
#   file name conversion function      : func_convert_file_X_to_Y ()
#   path conversion function          : func_convert_path_X_to_Y ()
# where, for any given $build/$host combination the 'X_to_Y' value is
the
# same.  If conversion functions are added for new $build/$host
combinations,
# the two new functions must follow this pattern, or
func_init_to_host_path_cmd
# will break.

# func_init_to_host_path_cmd

```

```

# Ensures that function "pointer" variable $to_host_path_cmd is set to
the
# appropriate value, based on the value of $to_host_file_cmd.
to_host_path_cmd=
func_init_to_host_path_cmd ()
{
    $opt_debug
    if test -z "$to_host_path_cmd"; then
        func_stripname 'func_convert_file_' '' "$to_host_file_cmd"
        to_host_path_cmd="func_convert_path_${func_stripname_result}"
    fi
}

# func_to_host_path ARG
# Converts the path ARG from $build format to $host format. Return
result
# in func_to_host_path_result.
func_to_host_path ()
{
    $opt_debug
    func_init_to_host_path_cmd
    $to_host_path_cmd "$1"
}
# end func_to_host_path

# func_convert_path_noop ARG
# Copy ARG to func_to_host_path_result.
func_convert_path_noop ()
{
    func_to_host_path_result="$1"
}
# end func_convert_path_noop

# func_convert_path_msys_to_w32 ARG
# Convert path ARG from (mingw) MSYS to (mingw) w32 format; automatic
# conversion to w32 is not available inside the cwrapper. Returns
result in
# func_to_host_path_result.
func_convert_path_msys_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # Remove leading and trailing path separator characters from ARG.
MSYS
        # behavior is inconsistent here; cygpath turns them into '.' and
';.';
        # and winepath ignores them completely.
        func_stripname : : "$1"
    fi
}

```

```

func_to_host_path_tmpl=$func_stripname_result
func_convert_core_msys_to_w32 "$func_to_host_path_tmpl"
func_to_host_path_result="$func_convert_core_msys_to_w32_result"
func_convert_path_check : ";" \
    "$func_to_host_path_tmpl" "$func_to_host_path_result"
func_convert_path_front_back_pathsep ":*" "*:;" ";" "$1"
fi
}
# end func_convert_path_msys_to_w32

# func_convert_path_cygwin_to_w32 ARG
# Convert path ARG from Cygwin to w32 format. Returns result in
# func_to_host_file_result.
func_convert_path_cygwin_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmpl=$func_stripname_result
        func_to_host_path_result=`cygpath -m -p "$func_to_host_path_tmpl"`
        func_convert_path_check : ";" \
            "$func_to_host_path_tmpl" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*:;" ";" "$1"
    fi
}
# end func_convert_path_cygwin_to_w32

# func_convert_path_nix_to_w32 ARG
# Convert path ARG from *nix to w32 format. Requires a wine
environment and
# a working winepath. Returns result in func_to_host_file_result.
func_convert_path_nix_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmpl=$func_stripname_result
        func_convert_core_path_wine_to_w32 "$func_to_host_path_tmpl"

func_to_host_path_result="$func_convert_core_path_wine_to_w32_result"
        func_convert_path_check : ";" \
            "$func_to_host_path_tmpl" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*:;" ";" "$1"
    fi
}
# end func_convert_path_nix_to_w32

```

```

# func_convert_path_msys_to_cygwin ARG
# Convert path ARG from MSYS to Cygwin format.  Requires LT_CYGPATH
set.
# Returns result in func_to_host_file_result.
func_convert_path_msys_to_cygwin ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmp1=$func_stripname_result
        func_convert_core_msys_to_w32 "$func_to_host_path_tmp1"
        func_cygpath -u -p "$func_convert_core_msys_to_w32_result"
        func_to_host_path_result="$func_cygpath_result"
        func_convert_path_check : : \
            "$func_to_host_path_tmp1" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*::" : "$1"
    fi
}
# end func_convert_path_msys_to_cygwin

# func_convert_path_nix_to_cygwin ARG
# Convert path ARG from *nix to Cygwin format.  Requires Cygwin
installed in a
# a wine environment, working winepath, and LT_CYGPATH set.  Returns
result in
# func_to_host_file_result.
func_convert_path_nix_to_cygwin ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # Remove leading and trailing path separator characters from
        # ARG. msys behavior is inconsistent here, cygpath turns them
        # into ';' and ';.', and winepath ignores them completely.
        func_stripname : : "$1"
        func_to_host_path_tmp1=$func_stripname_result
        func_convert_core_path_wine_to_w32 "$func_to_host_path_tmp1"
        func_cygpath -u -p "$func_convert_core_path_wine_to_w32_result"
        func_to_host_path_result="$func_cygpath_result"
        func_convert_path_check : : \
            "$func_to_host_path_tmp1" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*::" : "$1"
    fi
}
# end func_convert_path_nix_to_cygwin

```

```

# func_mode_compile arg...
func_mode_compile ()
{
    $opt_debug
    # Get the compilation command and the source file.
    base_compile=
    srcfile="$nonopt" # always keep a non-empty value in "srcfile"
    suppress_opt=yes
    suppress_output=
    arg_mode=normal
    libobj=
    later=
    pie_flag=

    for arg
    do
        case $arg_mode in
            arg )
                # do not "continue". Instead, add this to base_compile
                lastarg="$arg"
                arg_mode=normal
                ;;

            target )
                libobj="$arg"
                arg_mode=normal
                continue
                ;;

            normal )
                # Accept any command-line options.
                case $arg in
                    -o)
                        test -n "$libobj" && \
                            func_fatal_error "you cannot specify \`-o' more than once"
                        arg_mode=target
                        continue
                        ;;

                    -pie | -fpie | -fPIE)
                        pie_flag+=" $arg"
                        continue
                        ;;

                    -shared | -static | -prefer-pic | -prefer-non-pic)
                        later+=" $arg"
                        continue
                        ;;

                    -no-suppress)
                        suppress_opt=no
                        continue
                esac
            esac
        done
    done
}

```

```

        ;;

-Xcompiler)
    arg_mode=arg # the next one goes into the "base_compile" arg
list
    continue # The current "srcfile" will either be retained
or
    ;; # replaced later. I would guess that would be a
bug.

-Wc,*)
    func_stripname '-Wc,' '' "$arg"
    args=$func_stripname_result
    lastarg=
    save_ifs="$IFS"; IFS=', '
    for arg in $args; do
        IFS="$save_ifs"
        func_append_quoted lastarg "$arg"
    done
    IFS="$save_ifs"
    func_stripname ' ' '' "$lastarg"
    lastarg=$func_stripname_result

    # Add the arguments to base_compile.
    base_compile+=" $lastarg"
    continue
    ;;

*)
    # Accept the current argument as the source file.
    # The previous "srcfile" becomes the current argument.
    #
    lastarg="$srcfile"
    srcfile="$arg"
    ;;
esac # case $arg
;;
esac # case $arg_mode

# Aesthetically quote the previous argument.
func_append_quoted base_compile "$lastarg"
done # for arg

case $arg_mode in
arg)
    func_fatal_error "you must specify an argument for -Xcompile"
    ;;
target)
    func_fatal_error "you must specify a target with \`-o'"
    ;;
*)
    # Get the name of the library object.

```

```

    test -z "$libobj" && {
    func_basename "$srcfile"
    libobj="$func_basename_result"
    }
    ;;
esac

# Recognize several different file suffixes.
# If the user specifies -o file.o, it is replaced with file.lo
case $libobj in
*. [cCFSifmso] | \
*.ada | *.adb | *.ads | *.asm | \
*.c++ | *.cc | *.ii | *.class | *.cpp | *.cxx | \
*. [fF][09]? | *.for | *.java | *.go | *.obj | *.sx | *.cu | *.cup)
    func_xform "$libobj"
    libobj=$func_xform_result
    ;;
esac

case $libobj in
*.lo) func_lo2o "$libobj"; obj=$func_lo2o_result ;;
*)
    func_fatal_error "cannot determine name of library object from
\`$libobj'"
    ;;
esac

func_infer_tag $base_compile

for arg in $later; do
    case $arg in
    -shared)
        test "$build_libtool_libs" != yes && \
            func_fatal_configuration "can not build a shared library"
        build_old_libs=no
        continue
        ;;

    -static)
        build_libtool_libs=no
        build_old_libs=yes
        continue
        ;;

    -prefer-pic)
        pic_mode=yes
        continue
        ;;

    -prefer-non-pic)
        pic_mode=no
        continue
    )

```

```

;;
esac
done

func_quote_for_eval "$libobj"
test "X$libobj" != "X$func_quote_for_eval_result" \
  && $ECHO "X$libobj" | $GREP '[~#^*{};<>?'"'"' &()|`$[]' \
  && func_warning "libobj name `$libobj' may not contain shell
special characters."
func_dirname_and_basename "$obj" "/" ""
objname="$func_basename_result"
xdir="$func_dirname_result"
lobj=${xdir}$objdir/$objname

test -z "$base_compile" && \
  func_fatal_help "you must specify a compilation command"

# Delete any leftover library objects.
if test "$build_old_libs" = yes; then
  removelist="$obj $lobj $libobj ${libobj}T"
else
  removelist="$lobj $libobj ${libobj}T"
fi

# On Cygwin there's no "real" PIC flag so we must build both
object types
case $host_os in
  cygwin* | mingw* | pw32* | os2* | cegcc*)
    pic_mode=default
    ;;
esac
if test "$pic_mode" = no && test "$deplibs_check_method" !=
pass_all; then
  # non-PIC code in shared libraries is not supported
  pic_mode=default
fi

# Calculate the filename of the output object if compiler does
# not support -o with -c
if test "$compiler_c_o" = no; then
  output_obj=`$ECHO "$srcfile" | $SED 's%^.*/%%;
s%\.[^.]*%%'\`.$objext}
  lockfile="$output_obj.lock"
else
  output_obj=
  need_locks=no
  lockfile=
fi

# Lock this critical section if it is needed
# We use this script file to make the link, it avoids creating a
new file

```



```

if test "$need_locks" = yes; then
    until $opt_dry_run || ln "$proppath" "$lockfile" 2>/dev/null; do
        func_echo "Waiting for $lockfile to be removed"
        sleep 2
    done
elif test "$need_locks" = warn; then
    if test -f "$lockfile"; then
        $ECHO "\
*** ERROR, $lockfile exists and contains:
`cat $lockfile 2>/dev/null`

```

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```

    $opt_dry_run || $RM $removelist
    exit $EXIT_FAILURE
fi
    removelist+=" $output_obj"
    $ECHO "$srcfile" > "$lockfile"
fi

$opt_dry_run || $RM $removelist
removelist+=" $lockfile"
trap '$opt_dry_run || $RM $removelist; exit $EXIT_FAILURE' 1 2 15

func_to_tool_file "$srcfile" func_convert_file_msys_to_w32
srcfile=$func_to_tool_file_result
func_quote_for_eval "$srcfile"
qsrcfile=$func_quote_for_eval_result

# Only build a PIC object if we are building libtool libraries.
if test "$build_libtool_libs" = yes; then
    # Without this assignment, base_compile gets emptied.
    fbsd_hideous_sh_bug=$base_compile

    if test "$pic_mode" != no; then
        command="$base_compile $qsrcfile $pic_flag"
    else
        # Don't build PIC code
        command="$base_compile $qsrcfile"
    fi

    func_mkdir_p "$xdir$objdir"

    if test -z "$output_obj"; then
        # Place PIC objects in $objdir
        command+=" -o $lobj"
    fi
fi

```

```
func_show_eval_locale "$command"\  
    'test -n "$output_obj" && $RM $removelist; exit  
$EXIT_FAILURE'
```

```
    if test "$need_locks" = warn &&  
        test "X`cat $lockfile 2>/dev/null`" != "X$srcfile"; then  
        $ECHO "\n  
*** ERROR, $lockfile contains:  
`cat $lockfile 2>/dev/null`
```

but it should contain:
\$srcfile

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```
    $opt_dry_run || $RM $removelist  
    exit $EXIT_FAILURE  
    fi  
  
    # Just move the object if needed, then go on to compile the next  
one  
    if test -n "$output_obj" && test "X$output_obj" != "X$obj";  
then  
    func_show_eval '$MV "$output_obj" "$obj"' \  
        'error=$?; $opt_dry_run || $RM $removelist; exit $error'  
    fi  
  
    # Allow error messages only from the first compilation.  
    if test "$suppress_opt" = yes; then  
    suppress_output=' >/dev/null 2>&1'  
    fi  
    fi  
  
    # Only build a position-dependent object if we build old  
libraries.  
    if test "$build_old_libs" = yes; then  
        if test "$pic_mode" != yes; then  
            # Don't build PIC code  
            command="$base_compile $qsrcfile$pie_flag"  
        else  
            command="$base_compile $qsrcfile $pic_flag"  
        fi  
        if test "$compiler_c_o" = yes; then  
            command+=" -o $obj"  
        fi  
    fi
```

```

# Suppress compiler output if we already did a PIC compilation.
command+=" $suppress_output"
func_show_eval_locale "$command" \
  '$opt_dry_run || $RM $removelist; exit $EXIT_FAILURE'

if test "$need_locks" = warn &&
  test "X`cat $lockfile 2>/dev/null`" != "X$srcfile"; then
  $ECHO "\
*** ERROR, $lockfile contains:
`cat $lockfile 2>/dev/null`

```

but it should contain:
\$srcfile

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```

$opt_dry_run || $RM $removelist
exit $EXIT_FAILURE
fi

# Just move the object if needed
if test -n "$output_obj" && test "X$output_obj" != "X$obj"; then
func_show_eval '$MV "$output_obj" "$obj"' \
  'error=$?; $opt_dry_run || $RM $removelist; exit $error'
fi
fi

$opt_dry_run || {
  func_write_libtool_object "$libobj" "$objdir/$objname"
"$objname"

  # Unlock the critical section if it was locked
  if test "$need_locks" != no; then
  removelist=$lockfile
    $RM "$lockfile"
  fi
}

exit $EXIT_SUCCESS
}

$opt_help || {
  test "$opt_mode" = compile && func_mode_compile ${1+"$@"}
}

func_mode_help ()
{

```

```

# We need to display help for each of the modes.
case $opt_mode in
  "")
    # Generic help is extracted from the usage comments
    # at the start of this file.
    func_help
    ;;

  clean)
    $ECHO \
"Usage: $progname [OPTION]... --mode=clean RM [RM-OPTION]... FILE...

Remove files from the build directory.

RM is the name of the program to use to delete files associated with
each FILE
(typically \"/bin/rm').  RM-OPTIONS are options (such as \'-f') to be
passed
to RM.

If FILE is a libtool library, object or program, all the files
associated
with it are deleted. Otherwise, only FILE itself is deleted using RM."
    ;;

  compile)
    $ECHO \
"Usage: $progname [OPTION]... --mode=compile COMPILE-COMMAND...
SOURCEFILE

Compile a source file into a libtool library object.

This mode accepts the following additional options:

  -o OUTPUT-FILE      set the output file name to OUTPUT-FILE
  -no-suppress        do not suppress compiler output for multiple
passes
  -prefer-pic         try to build PIC objects only
  -prefer-non-pic     try to build non-PIC objects only
  -shared             do not build a \'.o' file suitable for static
linking
  -static             only build a \'.o' file suitable for static
linking
  -Wc,FLAG           pass FLAG directly to the compiler

COMPILE-COMMAND is a command to be used in creating a \'standard'
object file
from the given SOURCEFILE.

The output file name is determined by removing the directory component
from
SOURCEFILE, then substituting the C source code suffix \'.c' with the

```

```
library object suffix, \`.lo'."
```

```
;;
```

```
execute)
```

```
$ECHO \
```

```
"Usage: $progname [OPTION]... --mode=execute COMMAND [ARGS]..."
```

Automatically set library path, then run a program.

This mode accepts the following additional options:

```
-dlopen FILE      add the directory containing FILE to the library  
path
```

This mode sets the library path environment variable according to \`-dlopen' flags.

If any of the ARGS are libtool executable wrappers, then they are translated into their corresponding uninstalled binary, and any of their required library directories are added to the library path.

Then, COMMAND is executed, with ARGS as arguments."

```
;;
```

```
finish)
```

```
$ECHO \
```

```
"Usage: $progname [OPTION]... --mode=finish [LIBDIR]..."
```

Complete the installation of libtool libraries.

Each LIBDIR is a directory that contains libtool libraries.

The commands that this mode executes may require superuser privileges.

Use

the \`--dry-run' option if you just want to see what would be executed."

```
;;
```

```
install)
```

```
$ECHO \
```

```
"Usage: $progname [OPTION]... --mode=install INSTALL-COMMAND..."
```

Install executables or libraries.

INSTALL-COMMAND is the installation command. The first component should be either the \`install' or \`cp' program.

The following components of INSTALL-COMMAND are treated specially:

`-inst-prefix-dir PREFIX-DIR` Use PREFIX-DIR as a staging area for installation

The rest of the components are interpreted as arguments to that command (only BSD-compatible install options are recognized)."

`;;`

link)

`$ECHO \`

"Usage: \$progname [OPTION]... --mode=link LINK-COMMAND..."

Link object files or libraries together to form another library, or to create an executable program.

LINK-COMMAND is a command using the C compiler that you would use to create a program from several object files.

The following components of LINK-COMMAND are treated specially:

<code>-all-static</code>	do not do any dynamic linking at all
<code>-avoid-version</code>	do not add a version suffix if possible
<code>-bindir BINDIR</code>	specify path to binaries directory (for systems where
	libraries must be found in the PATH setting at
	runtime)
<code>-dlopen FILE</code>	<code>\`-dlpreopen'</code> FILE if it cannot be dlopened at
	runtime
<code>-dlpreopen FILE</code>	link in FILE and add its symbols to
<code>lt_preloaded_symbols</code>	
<code>-export-dynamic</code>	allow symbols from OUTPUT-FILE to be resolved with
	<code>dlsym(3)</code>
<code>-export-symbols SYMFILE</code>	
	try to export only the symbols listed in SYMFILE
<code>-export-symbols-regex REGEX</code>	
	try to export only the symbols matching REGEX
<code>-LLIBDIR</code>	search LIBDIR for required installed libraries
<code>-lname</code>	OUTPUT-FILE requires the installed library libNAME
<code>-module</code>	build a library that can dlopened
<code>-no-fast-install</code>	disable the fast-install mode
<code>-no-install</code>	link a not-installable executable
<code>-no-undefined</code>	declare that a library does not refer to external
	symbols
<code>-o OUTPUT-FILE</code>	create OUTPUT-FILE from the specified objects
<code>-objectlist FILE</code>	Use a list of object files found in FILE to
	specify objects
<code>-precious-files-regex REGEX</code>	
	don't remove output files matching REGEX
<code>-release RELEASE</code>	specify package release information

-rpath LIBDIR the created library will eventually be installed
 in LIBDIR
 -R[]LIBDIR add LIBDIR to the runtime path of programs and
 libraries
 -shared only do dynamic linking of libtool libraries
 -shrext SUFFIX override the standard shared library file
 extension
 -static do not do any dynamic linking of uninstalled
 libtool libraries
 -static-libtool-libs do not do any dynamic linking of libtool libraries
 -version-info CURRENT[:REVISION[:AGE]]
 specify library version info [each variable
 defaults to 0]
 -weak LIBNAME declare that the target provides the LIBNAME
 interface
 -Wc,FLAG pass linker-specific FLAG directly to the compiler
 -Xcompiler FLAG pass linker-specific FLAG directly to the compiler
 -Wl,FLAG pass linker-specific FLAG directly to the linker
 -Xlinker FLAG pass linker-specific FLAG directly to the linker
 -XCc linker FLAG pass link-specific FLAG to the compiler driver
 (CC)

All other options (arguments beginning with \`-') are ignored.

Every other argument is treated as a filename. Files ending in \`.la'
 are
 treated as uninstalled libtool libraries, other files are standard or
 library
 object files.

If the OUTPUT-FILE ends in \`.la', then a libtool library is created,
 only library objects (\`.lo' files) may be specified, and \`-rpath' is
 required, except when creating a convenience library.

If OUTPUT-FILE ends in \`.a' or \`.lib', then a standard library is
 created
 using \`.ar' and \`.ranlib', or on Windows using \`.lib'.

If OUTPUT-FILE ends in \`.lo' or \`\${objext}', then a reloadable
 object file
 is created, otherwise an executable program is created."

```

    ;;

    uninstall)
      $ECHO \
"Usage: $progname [OPTION]... --mode=uninstall RM [RM-OPTION]...
FILE..."
  
```

Remove libraries from an installation directory.

RM is the name of the program to use to delete files associated with each FILE (typically `\`/bin/rm'`). RM-OPTIONS are options (such as `\`-f'`) to be passed to RM.

If FILE is a libtool library, all the files associated with it are deleted.

Otherwise, only FILE itself is deleted using RM."

```
;;

*)
  func_fatal_help "invalid operation mode \`${opt_mode}'"
;;
esac

echo
$ECHO "Try \`${progname} --help' for more information about other
modes."
}
```

Now that we've collected a possible --mode arg, show help if necessary

```
if $opt_help; then
  if test "$opt_help" = :; then
    func_mode_help
  else
    {
      func_help noexit
      for opt_mode in compile link execute install finish uninstall
clean; do
        func_mode_help
        done
    } | sed -n '1p; 2,$s/^Usage:/ or: /p'
    {
      func_help noexit
      for opt_mode in compile link execute install finish uninstall
clean; do
        echo
        func_mode_help
        done
    } |
    sed '1d
/^When reporting/,/^Report/{
H
d
}
$x
/information about other modes/d
/more detailed .*MODE/d
s/^Usage:.*--mode=\([^ ]*\) .*/Description of \1 mode:/'
fi
```



```

    exit $?
fi

# func_mode_execute arg...
func_mode_execute ()
{
    $opt_debug
    # The first argument is the command name.
    cmd="$nonopt"
    test -z "$cmd" && \
        func_fatal_help "you must specify a COMMAND"

    # Handle -dlopen flags immediately.
    for file in $opt_dlopen; do
        test -f "$file" \
            || func_fatal_help "\`$file' is not a file"

        dir=
        case $file in
            *.la)
                func_resolve_sysroot "$file"
                file=$func_resolve_sysroot_result

                # Check to see that this really is a libtool archive.
                func_lalib_unsafe_p "$file" \
                    || func_fatal_help "\`$lib' is not a valid libtool archive"

                # Read the libtool library.
                dlname=
                library_names=
                func_source "$file"

                # Skip this library if it cannot be dlopened.
                if test -z "$dlname"; then
                    # Warn if it was a shared library.
                    test -n "$library_names" && \
                        func_warning "\`$file' was not linked with \`-export-
dynamic'"
                    continue
                fi

                func_dirname "$file" "" "."
                dir="$func_dirname_result"

                if test -f "$dir/$objdir/$dlname"; then
                    dir+="/$objdir"
                else
                    if test ! -f "$dir/$dlname"; then
                        func_fatal_error "cannot find \`$dlname' in \`$dir' or
\`$dir/$objdir'"
                    fi
                fi
            fi
        fi
    fi
}

```

```

fi
;;

*.lo)
# Just add the directory containing the .lo file.
func_dirname "$file" "" "."
dir="$func_dirname_result"
;;

*)
func_warning "\`-dlopen' is ignored for non-libtool libraries and
objects"
continue
;;
esac

# Get the absolute pathname.
absdir=`cd "$dir" && pwd`
test -n "$absdir" && dir="$absdir"

# Now add the directory to shlibpath_var.
if eval "test -z \"\${shlibpath_var}\""; then
eval "shlibpath_var=\"\${shlibpath_var}\${dir}\""
else
eval "shlibpath_var=\"\${shlibpath_var}\${dir}:\${shlibpath_var}\""
fi
done

# This variable tells wrapper scripts just to set shlibpath_var
# rather than running their programs.
libtool_execute_magic="$magic"

# Check if any of the arguments is a wrapper script.
args=
for file
do
case $file in
-* | *.la | *.lo ) ;;
*)
# Do a test to see if this is really a libtool program.
if func_ltwrapper_script_p "$file"; then
func_source "$file"
# Transform arg to wrapped name.
file="$progdir/$program"
elif func_ltwrapper_executable_p "$file"; then
func_ltwrapper_scriptname "$file"
func_source "$func_ltwrapper_scriptname_result"
# Transform arg to wrapped name.
file="$progdir/$program"
fi
;;
;;
esac

```

```

    # Quote arguments (to preserve shell metacharacters).
    func_append_quoted args "$file"
done

if test "X$opt_dry_run" = Xfalse; then
  if test -n "$shlibpath_var"; then
    # Export the shlibpath_var.
    eval "export $shlibpath_var"
  fi

  # Restore saved environment variables
  for lt_var in LANG LANGUAGE LC_ALL LC_CTYPE LC_COLLATE
LC_MESSAGES
  do
    eval "if test \"\${save_$lt_var+set}\" = set; then
          $lt_var=\$save_$lt_var; export $lt_var
        else
          $lt_unset $lt_var
        fi"
  done

  # Now prepare to actually exec the command.
  exec_cmd="\$cmd$args"
else
  # Display what would be done.
  if test -n "$shlibpath_var"; then
    eval "\$ECHO \"\$shlibpath_var=\$$shlibpath_var\""
    echo "export $shlibpath_var"
  fi
  $ECHO "$cmd$args"
  exit $EXIT_SUCCESS
fi
}

test "$opt_mode" = execute && func_mode_execute ${1+"$@"}

# func_mode_finish arg...
func_mode_finish ()
{
  $opt_debug
  libs=
  libdirs=
  admincmds=

  for opt in "$nonopt" ${1+"$@"}
  do
    if test -d "$opt"; then
      libdirs+=" $opt"

      elif test -f "$opt"; then
        if func_lalib_unsafe_p "$opt"; then

```

```

    libs+=" $opt"
else
    func_warning "`$opt' is not a valid libtool archive"
fi

else
func_fatal_error "invalid argument `'$opt'"
fi
done

if test -n "$libs"; then
    if test -n "$lt_sysroot"; then
        sysroot_regex=`$ECHO "$lt_sysroot" | $SED
"$sed_make_literal_regex"`
        sysroot_cmd="s/\([ ']\)$sysroot_regex/\1/g;"
    else
        sysroot_cmd=
    fi

    # Remove sysroot references
    if $opt_dry_run; then
        for lib in $libs; do
            echo "removing references to $lt_sysroot and `=' prefixes
from $lib"
        done
    else
        tmpdir=`func_mkdir`
        for lib in $libs; do
            sed -e "${sysroot_cmd} s/\([ ']-[LR]\)=/\1/g; s/\([ ']\)=/\1/g"
$lib \
            > $tmpdir/tmp-la
            mv -f $tmpdir/tmp-la $lib
        done
        ${RM}r "$tmpdir"
    fi
fi

if test -n "$finish_cmds$finish_eval" && test -n "$libdirs"; then
    for libdir in $libdirs; do
        if test -n "$finish_cmds"; then
            # Do each command in the finish commands.
            func_execute_cmds "$finish_cmds" 'admincmds="$admincmds
'"$cmd"'"
        fi
        if test -n "$finish_eval"; then
            # Do the single finish_eval.
            eval cmds=\"$finish_eval\"
            $opt_dry_run || eval "$cmds" || admincmds+="
$cmds"
        fi
    done
fi

```

```

# Exit here if they wanted silent mode.
$opt_silent && exit $EXIT_SUCCESS

if test -n "$finish_cmds$finish_eval" && test -n "$libdirs"; then
  echo "-----"
  -----"
  echo "Libraries have been installed in:"
  for libdir in $libdirs; do
    $ECHO "  $libdir"
  done
  echo
  echo "If you ever happen to want to link against installed
libraries"
  echo "in a given directory, LIBDIR, you must either use libtool,
and"
  echo "specify the full pathname of the library, or use the \'-
LLIBDIR'"
  echo "flag during linking and do at least one of the following:"
  if test -n "$shlibpath_var"; then
    echo "  - add LIBDIR to the '$shlibpath_var' environment
variable"
    echo "    during execution"
  fi
  if test -n "$runpath_var"; then
    echo "  - add LIBDIR to the '$runpath_var' environment
variable"
    echo "    during linking"
  fi
  if test -n "$hardcode_libdir_flag_spec"; then
    libdir=LIBDIR
    eval flag=\"$hardcode_libdir_flag_spec\"

    $ECHO "  - use the '$flag' linker flag"
  fi
  if test -n "$admincmds"; then
    $ECHO "  - have your system administrator run these
commands:$admincmds"
  fi
  if test -f /etc/ld.so.conf; then
    echo "  - have your system administrator add LIBDIR to
\`/etc/ld.so.conf'"
  fi
  echo

  echo "See any operating system documentation about shared
libraries for"
  case $host in
    solaris2.[6789]|solaris2.1[0-9])
      echo "more information, such as the ld(1), crle(1) and ld.so(8)
manual"
      echo "pages."

```

```

        ;;
    *)
        echo "more information, such as the ld(1) and ld.so(8) manual
pages."
        ;;
    esac
    echo "-----"
-----"
    fi
    exit $EXIT_SUCCESS
}

test "$opt_mode" = finish && func_mode_finish ${1+"$@"}

# func_mode_install arg...
func_mode_install ()
{
    $opt_debug
    # There may be an optional sh(1) argument at the beginning of
    # install_prog (especially on Windows NT).
    if test "$nonopt" = "$SHELL" || test "$nonopt" = /bin/sh ||
        # Allow the use of GNU shtool's install command.
        case $nonopt in *shtool*) ;; *) false;; esac; then
        # Aesthetically quote it.
        func_quote_for_eval "$nonopt"
        install_prog="$func_quote_for_eval_result "
        arg=$1
        shift
    else
        install_prog=
        arg=$nonopt
    fi

    # The real first argument should be the name of the installation
    program.
    # Aesthetically quote it.
    func_quote_for_eval "$arg"
    install_prog+="$func_quote_for_eval_result"
    install_shared_prog=$install_prog
    case " $install_prog " in
        *[\ \ /]cp\ *) install_cp=: ;;
        *) install_cp=false ;;
    esac

    # We need to accept at least all the BSD install flags.
    dest=
    files=
    opts=
    prev=
    install_type=
    isdir=no

```

```

stripme=
no_mode=:
for arg
do
    arg2=
    if test -n "$dest"; then
files+=" $dest"
dest=$arg
continue
    fi

    case $arg in
    -d) isdir=yes ;;
    -f)
if $install_cp; then ;; else
    prev=$arg
    fi
    ;;
    -g | -m | -o)
prev=$arg
    ;;
    -s)
stripme=" -s"
continue
    ;;
    -*)
    ;;
    *)
# If the previous option needed an argument, then skip it.
if test -n "$prev"; then
    if test "x$prev" = x-m && test -n "$install_override_mode";
then
        arg2=$install_override_mode
        no_mode=false
        fi
        prev=
    else
        dest=$arg
        continue
    fi
    ;;
    esac

# Aesthetically quote the argument.
func_quote_for_eval "$arg"
install_prog+=" $func_quote_for_eval_result"
if test -n "$arg2"; then
func_quote_for_eval "$arg2"
    fi
install_shared_prog+=" $func_quote_for_eval_result"
done

```

```

test -z "$install_prog" && \
    func_fatal_help "you must specify an install program"

test -n "$prev" && \
    func_fatal_help "the '\$prev' option requires an argument"

if test -n "$install_override_mode" && $no_mode; then
    if $install_cp; then ;; else
        func_quote_for_eval "$install_override_mode"
        install_shared_prog+=" -m $func_quote_for_eval_result"
    fi
fi

if test -z "$files"; then
    if test -z "$dest"; then
        func_fatal_help "no file or destination specified"
    else
        func_fatal_help "you must specify a destination"
    fi
fi

# Strip any trailing slash from the destination.
func_stripname '' '/' "$dest"
dest=$func_stripname_result

# Check to see that the destination is a directory.
test -d "$dest" && isdir=yes
if test "$isdir" = yes; then
    destdir="$dest"
    destname=
else
    func_dirname_and_basename "$dest" "" "."
    destdir="$func_dirname_result"
    destname="$func_basename_result"

    # Not a directory, so check to see that there is only one file
    specified.
    set dummy $files; shift
    test "$#" -gt 1 && \
        func_fatal_help "\`$dest' is not a directory"
fi
case $destdir in
[\\/*]* | [A-Za-z]:[\\/*]*) ;;
*)
    for file in $files; do
        case $file in
*.lo) ;;
*)
            func_fatal_help "\`$destdir' must be an absolute directory
name"
            ;;
        esac
    done
esac

```



```

done
;;
esac

# This variable tells wrapper scripts just to set variables rather
# than running their programs.
libtool_install_magic="$magic"

staticlibs=
future_libdirs=
current_libdirs=
for file in $files; do

    # Do each installation.
    case $file in
        *.$libext)
            # Do the static libraries later.
            staticlibs+=" $file"
            ;;

        *.la)
            func_resolve_sysroot "$file"
            file=$func_resolve_sysroot_result

            # Check to see that this really is a libtool archive.
            func_lalib_unsafe_p "$file" \
                || func_fatal_help "`$file' is not a valid libtool archive"

            library_names=
            old_library=
            relink_command=
            func_source "$file"

            # Add the libdir to current_libdirs if it is the destination.
            if test "X$destdir" = "X$libdir"; then
                case "$current_libdirs " in
                    *" $libdir "*) ;;
                    *) current_libdirs+=" $libdir" ;;
                esac
            else
                # Note the libdir as a future libdir.
                case "$future_libdirs " in
                    *" $libdir "*) ;;
                    *) future_libdirs+=" $libdir" ;;
                esac
            fi

            func_dirname "$file" "/" ""
            dir="$func_dirname_result"
            dir+="$objdir"

            if test "$fast_install" = no && test -n "$relink_command"; then

```

```

# Strip any trailing slash from the destination.
func_stripname '' '/' "$libdir"
destlibdir=$func_stripname_result

func_stripname '' '/' "$destdir"
s_destdir=$func_stripname_result

# Determine the prefix the user has applied to our future dir.
inst_prefix_dir=`$ECHO "X$s_destdir" | $Xsed -e
"s%$destlibdir\$%"`

# Don't allow the user to place us outside of our expected
# location b/c this prevents finding dependent libraries that
# are installed to the same prefix.
# At present, this check doesn't affect windows .dll's that
# are installed into $libdir/../bin (currently, that works
fine)
# but it's something to keep an eye on.
test "$inst_prefix_dir" = "$destdir" && \
func_fatal_error "error: cannot install `'$file'` to a
directory not ending in $libdir"

if test -n "$inst_prefix_dir"; then
# Stick the inst_prefix_dir data into the link command.
relink_command=`$ECHO "$relink_command" | $SED
"s%@inst_prefix_dir@%-inst-prefix-dir $inst_prefix_dir%"`
else
relink_command=`$ECHO "$relink_command" | $SED
"s%@inst_prefix_dir@%"`
fi

func_warning "relinking `'$file'`"
func_show_eval "$relink_command" \
'func_fatal_error "error: relink `'$file'`\' with the above
command before installing it"'
fi

# See the names of the shared library.
set dummy $library_names; shift
if test -n "$1"; then
realname="$1"
shift

srcname="$realname"
test "$fast_install" = no && test -n "$relink_command" &&
srcname="$realname"

# Install the shared library and build the symlinks.
func_show_eval "$install_shared_prog $dir/$srcname
$destdir/$realname" \
'exit $?'
tstripme="$stripme"

```

```

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
    case $realname in
        *.dll.a)
            tstripme=""
            ;;
        esac
        ;;
    esac
if test -n "$tstripme" && test -n "$striplib"; then
    func_show_eval "$striplib $destdir/$realname" 'exit $?'
fi

if test "$#" -gt 0; then
    # Delete the old symlinks, and create new ones.
    # Try `ln -sf' first, because the `ln' binary might depend on
    # the symlink we replace! Solaris /bin/ln does not
understand -f,
    # so we also need to try rm && ln -s.
    for linkname
    do
        test "$linkname" != "$realname" \
            && func_show_eval "(cd $destdir && { $LN_S -f $realname
$linkname || { $RM $linkname && $LN_S $realname $linkname; }; })"
    done
fi

    # Do each command in the postinstall commands.
    lib="$destdir/$realname"
    func_execute_cmds "$postinstall_cmds" 'exit $?'
fi

# Install the pseudo-library for information purposes.
func_basename "$file"
name="$func_basename_result"
instname="$dir/$name"i
func_show_eval "$install_prog $instname $destdir/$name" 'exit $?'

# Maybe install the static library, too.
test -n "$old_library" && staticlibs+=" $dir/$old_library"
;;

*.lo)
# Install (i.e. copy) a libtool object.

# Figure out destination file name, if it wasn't already
specified.
if test -n "$destname"; then
    destfile="$destdir/$destname"
else
    func_basename "$file"
    destfile="$func_basename_result"

```

```

    destfile="$destdir/$destfile"
fi

# Deduce the name of the destination old-style object file.
case $destfile in
*.lo)
    func_lo2o "$destfile"
    staticdest=$func_lo2o_result
    ;;
*.$objext)
    staticdest="$destfile"
    destfile=
    ;;
*)
    func_fatal_help "cannot copy a libtool object to ``$destfile'"
    ;;
esac

# Install the libtool object if requested.
test -n "$destfile" && \
    func_show_eval "$install_prog $file $destfile" 'exit $?'

# Install the old object if enabled.
if test "$build_old_libs" = yes; then
    # Deduce the name of the old-style object file.
    func_lo2o "$file"
    staticobj=$func_lo2o_result
    func_show_eval "$install_prog \ $staticobj \ $staticdest" 'exit
$?'
fi
exit $EXIT_SUCCESS
;;

*)
# Figure out destination file name, if it wasn't already
specified.
if test -n "$destname"; then
    destfile="$destdir/$destname"
else
    func_basename "$file"
    destfile="$func_basename_result"
    destfile="$destdir/$destfile"
fi

# If the file is missing, and there is a .exe on the end, strip
it
# because it is most likely a libtool script we actually want to
# install
stripped_ext=""
case $file in
*.exe)
    if test ! -f "$file"; then

```

```

        func_stripname ' ' '.exe' "$file"
        file=${func_stripname_result}
        stripped_ext=".exe"
    fi
    ;;
esac

# Do a test to see if this is really a libtool program.
case $host in
*cygwin* | *mingw*)
    if func_ltwrapper_executable_p "$file"; then
        func_ltwrapper_scriptname "$file"
        wrapper=${func_ltwrapper_scriptname_result}
    else
        func_stripname ' ' '.exe' "$file"
        wrapper=${func_stripname_result}
    fi
    ;;
*)
    wrapper=$file
    ;;
esac
if func_ltwrapper_script_p "$wrapper"; then
    notinst_deplibs=
    relink_command=

    func_source "$wrapper"

    # Check the variables that should have been set.
    test -z "$generated_by_libtool_version" && \
        func_fatal_error "invalid libtool wrapper script \`$wrapper'"

    finalize=yes
    for lib in $notinst_deplibs; do
        # Check to see that each library is installed.
        libdir=
        if test -f "$lib"; then
            func_source "$lib"
        fi
        libfile="$libdir/"`$ECHO "$lib" | $SED 's%^.*/%%g'`###
    testsuite: skip nested quoting test
        if test -n "$libdir" && test ! -f "$libfile"; then
            func_warning "`$lib' has not been installed in \`$libdir'"
            finalize=no
        fi
    done

    relink_command=
    func_source "$wrapper"

    outputname=
    if test "$fast_install" = no && test -n "$relink_command"; then

```

```

$opt_dry_run || {
    if test "$finalize" = yes; then
        tmpdir=`func_mkdir`
        func_basename "$file$stripped_ext"
        file="$func_basename_result"
        outputname="$tmpdir/$file"
        # Replace the output file specification.
        relink_command=`$ECHO "$relink_command" | $SED
's%@OUTPUT@%' "$outputname"'%g'`

        $opt_silent || {
            func_quote_for_expand "$relink_command"
            eval "func_echo $func_quote_for_expand_result"
        }
        if eval "$relink_command"; then :
        else
            func_error "error: relink \`${file}' with the above command
before installing it"
            $opt_dry_run || ${RM}r "$tmpdir"
            continue
        fi
        file="$outputname"
    else
        func_warning "cannot relink \`${file}'"
    fi
}
else
    # Install the binary that we compiled earlier.
    file=`$ECHO "$file$stripped_ext" | $SED
"s%\([^/]*\)%%$objdir/\1%"`
    fi
fi

# remove .exe since cygwin /usr/bin/install will append another
# one anyway
case $install_prog,$host in
*/usr/bin/install*,*cygwin*)
    case $file:$destfile in
*.exe:*.exe)
        # this is ok
        ;;
*.exe:*)
        destfile=$destfile.exe
        ;;
*:*.*.exe)
        func_stripname '' '.exe' "$destfile"
        destfile=$func_stripname_result
        ;;
    esac
;;
esac

```

```

func_show_eval "$install_prog\$stripme \$file \$destfile" 'exit
$?'
$opt_dry_run || if test -n "$outputname"; then
    ${RM}r "$tmpdir"
fi
;;
esac
done

for file in $staticlibs; do
    func_basename "$file"
    name="$func_basename_result"

    # Set up the ranlib parameters.
    oldlib="$destdir/$name"
    func_to_tool_file "$oldlib" func_convert_file_msys_to_w32
    tool_oldlib=$func_to_tool_file_result

    func_show_eval "$install_prog \$file \$oldlib" 'exit $?'

    if test -n "$stripme" && test -n "$old_striplib"; then
        func_show_eval "$old_striplib $tool_oldlib" 'exit $?'
    fi

    # Do each command in the postinstall commands.
    func_execute_cmds "$old_postinstall_cmds" 'exit $?'
done

test -n "$future_libdirs" && \
    func_warning "remember to run \"\$progname --
finish$future_libdirs'"

if test -n "$current_libdirs"; then
    # Maybe just do a dry run.
    $opt_dry_run && current_libdirs=" -n$current_libdirs"
    exec_cmd='$SHELL $proppath $preserve_args --
finish$current_libdirs'
else
    exit $EXIT_SUCCESS
fi
}

test "$opt_mode" = install && func_mode_install ${1+"$@"}

# func_generate_dlsyms outputname originator pic_p
# Extract symbols from dlprefiles and create ${outputname}.S.o with
# a dlpreopen symbol table.
func_generate_dlsyms ()
{
    $opt_debug
    my_outputname="$1"

```

```

my_originator="$2"
my_pic_p="{3-no}"
my_prefix=`$ECHO "$my_originator" | sed 's%[^a-zA-Z0-9]%%g'`
my_dlsyms=

if test -n "$dlfiles$dlprefiles" || test "$dlself" != no; then
  if test -n "$NM" && test -n "$global_symbol_pipe"; then
    my_dlsyms="{my_outputname}S.c"
  else
    func_error "not configured to extract global symbols from
dlpreopened files"
  fi
fi

if test -n "$my_dlsyms"; then
  case $my_dlsyms in
    "") ;;
    *.c)
      # Discover the nlist of each of the dlfiles.
      nlist="$output_objdir/{my_outputname}.nm"

      func_show_eval "$RM $nlist ${nlist}S ${nlist}T"

      # Parse the name list into a source file.
      func_verbose "creating $output_objdir/$my_dlsyms"

      $opt_dry_run || $ECHO > "$output_objdir/$my_dlsyms" "\
/* $my_dlsyms - symbol resolution table for `'$my_outputname'` dlsym
emulation. */
/* Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION */

#ifdef __cplusplus
extern \"C\" {
#endif

#ifdef __GNUC__ && (((__GNUC__ == 4) && (__GNUC_MINOR__ >= 4))
|| (__GNUC__ > 4))
#pragma GCC diagnostic ignored \"-Wstrict-prototypes\"
#endif

/* Keep this code in sync between libtool.m4, ltmain, lt_system.h, and
tests. */
#ifdef __WIN32 || defined(__CYGWIN__) || defined(_WIN32_WCE)
/* DATA imports from DLLs on WIN32 can't be const, because runtime
relocations are performed -- see ld's documentation on pseudo-
relocs. */
# define LT_DLSYM_CONST
#elif defined(__osf__)
/* This system does not cope well with relocations in const data. */
# define LT_DLSYM_CONST
#else
# define LT_DLSYM_CONST const

```



```

#endif

/* External symbol declarations for the compiler. */\
"

    if test "$dlself" = yes; then
        func_verbose "generating symbol list for \`${output}'"

        $opt_dry_run || echo ': @PROGRAM@ ' > "$nlist"

        # Add our own program objects to the symbol list.
        progfiles=`$ECHO "$objs$old_deplibs" | $SP2NL | $SED "$lo2o" |
$NL2SP`
        for progfile in $progfiles; do
            func_to_tool_file "$progfile" func_convert_file_msys_to_w32
            func_verbose "extracting global C symbols from
\`${func_to_tool_file_result}'"
            $opt_dry_run || eval "$NM $func_to_tool_file_result |
$global_symbol_pipe >> '$nlist'"
            done

            if test -n "$exclude_expsyms"; then
                $opt_dry_run || {
                    eval '$EGREP -v " ($exclude_expsyms)$" "$nlist" >
"$nlist.T"
                    eval '$MV "$nlist.T" "$nlist"'
                }
            fi

            if test -n "$export_symbols_regex"; then
                $opt_dry_run || {
                    eval '$EGREP -e "$export_symbols_regex" "$nlist" >
"$nlist.T"
                    eval '$MV "$nlist.T" "$nlist"'
                }
            fi

            # Prepare the list of exported symbols
            if test -z "$export_symbols"; then
                export_symbols="$output_objdir/$outputname.exp"
                $opt_dry_run || {
                    $RM $export_symbols
                    eval "${SED} -n -e '/^: @PROGRAM@ \$/d' -e 's/^.*
\(.*\)\$/\1/p' "' < "$nlist" > "$export_symbols"'
                    case $host in
                        *cygwin* | *mingw* | *cegcc* )
                            eval "echo EXPORTS "' >
"$output_objdir/$outputname.def"'
                            eval 'cat "$export_symbols" >>
"$output_objdir/$outputname.def"'
                            ;;
                    esac
                }
            fi
        }
    fi

```

```

    }
    else
        $opt_dry_run || {
            eval "${SED} -e 's/\([[:.*^$]\)/\\\\\\\\1/g' -e 's/^/ /' -e
's/\/$/'" < "$export_symbols" > "$output_objdir/$outputname.exp"
            eval '$GREP -f "$output_objdir/$outputname.exp" < "$nlist"
> "$nlist"T'
            eval '$MV "$nlist"T "$nlist"'
            case $host in
                *cygwin* | *mingw* | *cegcc* )
                    eval "echo EXPORTS "'>
"$output_objdir/$outputname.def"'
                    eval 'cat "$nlist" >> "$output_objdir/$outputname.def"'
                    ;;
            esac
        }
    fi
fi

for dlprefile in $dlprefiles; do
    func_verbose "extracting global C symbols from `"$dlprefile`"
    func_basename "$dlprefile"
    name="$func_basename_result"
    case $host in
        *cygwin* | *mingw* | *cegcc* )
            # if an import library, we need to obtain dlname
            if func_win32_import_lib_p "$dlprefile"; then
                func_tr_sh "$dlprefile"
                eval "curr_lafile=\$libfile_$func_tr_sh_result"
                dlprefile_dlbasename=""
                if test -n "$curr_lafile" && func_lalib_p "$curr_lafile";
then
                    # Use subshell, to avoid clobbering current variable
values
                    dlprefile_dlname=`source "$curr_lafile" && echo
"$dlname"`
                    if test -n "$dlprefile_dlname" ; then
                        func_basename "$dlprefile_dlname"
                        dlprefile_dlbasename="$func_basename_result"
                    else
                        # no lafile. user explicitly requested -dlpreopen
<import library>.
                        $sharedlib_from_linklib_cmd "$dlprefile"
                        dlprefile_dlbasename=$sharedlib_from_linklib_result
                    fi
                fi
            fi
            $opt_dry_run || {
                if test -n "$dlprefile_dlbasename" ; then
                    eval '$ECHO ": $dlprefile_dlbasename" >> "$nlist"'
                else
                    func_warning "Could not compute DLL name from $name"
                    eval '$ECHO ": $name " >> "$nlist"'
                fi
            }
        }
    }
fi

```

```

        fi
        func_to_tool_file "$dlpfile"
func_convert_file_msys_to_w32
        eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe |
        $SED -e '/I __imp/d' -e 's/I __nm_/D /;s/_nm_/' >>
'$nlist'"
    }
    else # not an import lib
        $opt_dry_run || {
            eval '$ECHO ": $name " >> "$nlist"'
            func_to_tool_file "$dlpfile"
func_convert_file_msys_to_w32
            eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe >> '$nlist'"
        }
    fi
;;
*)
    $opt_dry_run || {
        eval '$ECHO ": $name " >> "$nlist"'
        func_to_tool_file "$dlpfile"
func_convert_file_msys_to_w32
        eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe >> '$nlist'"
    }
;;
esac
done

$opt_dry_run || {
# Make sure we have at least an empty file.
test -f "$nlist" || : > "$nlist"

if test -n "$exclude_expsyms"; then
    $EGREP -v " ($exclude_expsyms)$" "$nlist" > "$nlist"
    $MV "$nlist" "$nlist"
fi

# Try sorting and uniquifying the output.
if $GREP -v "^: " < "$nlist" |
    if sort -k 3 </dev/null >/dev/null 2>&1; then
        sort -k 3
    else
        sort +2
    fi |
    uniq > "$nlist"; then
:
else
    $GREP -v "^: " < "$nlist" > "$nlist"
fi

```

```

        if test -f "$nlist"S; then
            eval "$global_symbol_to_cdecl" < "$nlist"S >>
"$output_objdir/$my_dlsyms"
        else
            echo '/* NONE */' >> "$output_objdir/$my_dlsyms"
        fi

        echo >> "$output_objdir/$my_dlsyms" "\

/* The mapping between symbol names and symbols. */
typedef struct {
    const char *name;
    void *address;
} lt_dlsymlist;
extern LT_DLSYM_CONST lt_dlsymlist
lt_${my_prefix}_LTX_preloaded_symbols[];
LT_DLSYM_CONST lt_dlsymlist
lt_${my_prefix}_LTX_preloaded_symbols[] =
{\
    { \ "$my_originator\ ", (void *) 0 },"

        case $need_lib_prefix in
        no)
            eval "$global_symbol_to_c_name_address" < "$nlist" >>
"$output_objdir/$my_dlsyms"
            ;;
        *)
            eval "$global_symbol_to_c_name_address_lib_prefix" < "$nlist"
>> "$output_objdir/$my_dlsyms"
            ;;
        esac
        echo >> "$output_objdir/$my_dlsyms" "\
{0, (void *) 0}
};

/* This works around a problem in FreeBSD linker */
#ifdef FREEBSD_WORKAROUND
static const void *lt_preloaded_setup() {
    return lt_${my_prefix}_LTX_preloaded_symbols;
}
#endif

#ifdef __cplusplus
}
#endif\
"

    } # !$opt_dry_run

    pic_flag_for_syntable=
    case "$compile_command " in
    *" -static ") ;;
    *)

```

```

case $host in
# compiling the symbol table file with pic_flag works around
# a FreeBSD bug that causes programs to crash when -lm is
# linked before any other PIC object.  But we must not use
# pic_flag when linking with -static.  The problem exists in
# FreeBSD 2.2.6 and is fixed in FreeBSD 3.1.
*-*-freebsd2.*|*-*-freebsd3.0*|*-*-freebsdelf3.0*)
    pic_flag_for_syntable=" $pic_flag -DFREEBSD_WORKAROUND" ;;
*-*-hpux*)
    pic_flag_for_syntable=" $pic_flag" ;;
*)
    if test "X$my_pic_p" != Xno; then
        pic_flag_for_syntable=" $pic_flag"
    fi
    ;;
esac
;;
esac
symtab_cflags=
for arg in $LTCFLAGS; do
    case $arg in
        -pie | -fpie | -fPIE) ;;
        *) symtab_cflags+=" $arg" ;;
    esac
done

# Now compile the dynamic symbol file.
func_show_eval '(cd $output_objdir && $LTCC$symtab_cflags -
c$no_builtin_flag$pic_flag_for_syntable "$my_dlsyms")' 'exit $?'

# Clean up the generated files.
func_show_eval '$RM "$output_objdir/$my_dlsyms" "$nlist"
"${nlist}S" "${nlist}T"'

# Transform the symbol file into the correct name.
symfileobj="$output_objdir/${my_outputname}S.$objext"
case $host in
*cygwin* | *mingw* | *cegcc* )
    if test -f "$output_objdir/$my_outputname.def"; then
        compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$output_objdir/$my_outputname.def $symfileobj%"`
        finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$output_objdir/$my_outputname.def $symfileobj%"`
    else
        compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
        finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
    fi
    ;;
*)

```

```

        compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
        finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
        ;;
    esac
    ;;
    *)
    func_fatal_error "unknown suffix for \`$my_dlsyms'"
    ;;
    esac
else
    # We keep going just in case the user didn't refer to
    # lt_preloaded_symbols.  The linker will fail if
global_symbol_pipe
    # really was required.

    # Nullify the symbol file.
    compile_command=`$ECHO "$compile_command" | $SED "s%
@SYMFIL@%%"`
    finalize_command=`$ECHO "$finalize_command" | $SED "s%
@SYMFIL@%%"`
    fi
}

# func_win32_libid arg
# return the library type of file 'arg'
#
# Need a lot of goo to handle *both* DLLs and import libs
# Has to be a shell function in order to 'eat' the argument
# that is supplied when $file_magic_command is called.
# Despite the name, also deal with 64 bit binaries.
func_win32_libid ()
{
    $opt_debug
    win32_libid_type="unknown"
    win32_fileres=`file -L $1 2>/dev/null`
    case $win32_fileres in
    *ar\ archive\ import\ library*) # definitely import
        win32_libid_type="x86 archive import"
        ;;
    *ar\ archive*) # could be an import, or static
        # Keep the egrep pattern in sync with the one in
_LT_CHECK_MAGIC_METHOD.
        if eval $OBJDUMP -f $1 | $SED -e '10q' 2>/dev/null |
            $EGREP 'file format (pei*-i386(.?architecture: i386)?|pe-arm-
wince|pe-x86-64)' >/dev/null; then
            func_to_tool_file "$1" func_convert_file_msys_to_w32
            win32_nmres=`eval $NM -f posix -A \`${func_to_tool_file_result}`"
            |
            $SED -n -e '
                1,100{

```

```

        / I /{
            s,.*,import,
            p
            q
        }
    }'`
    case $win32_nmres in
import*)  win32_libid_type="x86 archive import";;
*)       win32_libid_type="x86 archive static";;
esac
fi
;;
*DLL*)
    win32_libid_type="x86 DLL"
    ;;
*executable*) # but shell scripts are "executable" too...
    case $win32_fileres in
*MS\ Windows\ PE\ Intel*)
        win32_libid_type="x86 DLL"
        ;;
    esac
    ;;
esac
$ECHO "$win32_libid_type"
}

# func_cygming_dll_for_implib ARG
#
# Platform-specific function to extract the
# name of the DLL associated with the specified
# import library ARG.
# Invoked by eval'ing the libtool variable
#   $sharedlib_from_linklib_cmd
# Result is available in the variable
#   $sharedlib_from_linklib_result
func_cygming_dll_for_implib ()
{
    $opt_debug
    sharedlib_from_linklib_result=`$DLLTOOL --identify-strict --identify
"$1"`
}

# func_cygming_dll_for_implib_fallback_core SECTION_NAME LIBNAMEs
#
# This is the core of a fallback implementation of a
# platform-specific function to extract the name of the
# DLL associated with the specified import library LIBNAME.
#
# SECTION_NAME is either .idata$6 or .idata$7, depending
# on the platform and compiler that created the implib.
#
# Echoes the name of the DLL associated with the

```

```

# specified import library.
func_cygmimg_dll_for_implib_fallback_core ()
{
    $opt_debug
    match_literal=`$ECHO "$1" | $SED "$sed_make_literal_regex"`
    $OBJDUMP -s --section "$1" "$2" 2>/dev/null |
        $SED '/^Contents of section '"$match_literal"'/{
            # Place marker at beginning of archive member dllname section
            s/./====MARK====/
            p
            d
        }
        # These lines can sometimes be longer than 43 characters, but
        # are always uninteresting
        /:[ ]*file format pe[i]\{,1\}-/d
        /^In archive [^:]*:/d
        # Ensure marker is printed
        /^====MARK====/p
        # Remove all lines with less than 43 characters
        /^.\{43\}/!d
        # From remaining lines, remove first 43 characters
        s/^.\{43\}//' |
        $SED -n '
            # Join marker and all lines until next marker into a single line
            /^====MARK====/ b para
            H
            $ b para
            b
            :para
            x
            s/\n//g
            # Remove the marker
            s/^====MARK====//
            # Remove trailing dots and whitespace
            s/[\. \t]*$//
            # Print
            ./p' |
        # we now have a list, one entry per line, of the stringified
        # contents of the appropriate section of all members of the
        # archive which possess that section. Heuristic: eliminate
        # all those which have a first or second character that is
        # a '.' (that is, objdump's representation of an unprintable
        # character.) This should work for all archives with less than
        # 0x302f exports -- but will fail for DLLs whose name actually
        # begins with a literal '.' or a single character followed by
        # a '.'.
        #
        # Of those that remain, print the first one.
        $SED -e '/^\./d;/^\./d;q'
    }

# func_cygmimg_gnu_implib_p ARG

```



```

# This predicate returns with zero status (TRUE) if
# ARG is a GNU/binutils-style import library. Returns
# with nonzero status (FALSE) otherwise.
func_cygming_gnu_implib_p ()
{
    $opt_debug
    func_to_tool_file "$1" func_convert_file_msys_to_w32
    func_cygming_gnu_implib_tmp=`$NM "$func_to_tool_file_result" | eval
"$global_symbol_pipe" | $EGREP ' (_head_[A-Za-z0-9_]+_[ad]l*|[A-Za-z0-
9_]+_[ad]l*_iname)$'`
    test -n "$func_cygming_gnu_implib_tmp"
}

# func_cygming_ms_implib_p ARG
# This predicate returns with zero status (TRUE) if
# ARG is an MS-style import library. Returns
# with nonzero status (FALSE) otherwise.
func_cygming_ms_implib_p ()
{
    $opt_debug
    func_to_tool_file "$1" func_convert_file_msys_to_w32
    func_cygming_ms_implib_tmp=`$NM "$func_to_tool_file_result" | eval
"$global_symbol_pipe" | $GREP '_NULL_IMPORT_DESCRIPTOR'`
    test -n "$func_cygming_ms_implib_tmp"
}

# func_cygming_dll_for_implib_fallback ARG
# Platform-specific function to extract the
# name of the DLL associated with the specified
# import library ARG.
#
# This fallback implementation is for use when $DLLTOOL
# does not support the --identify-strict option.
# Invoked by eval'ing the libtool variable
#   $sharedlib_from_linklib_cmd
# Result is available in the variable
#   $sharedlib_from_linklib_result
func_cygming_dll_for_implib_fallback ()
{
    $opt_debug
    if func_cygming_gnu_implib_p "$1" ; then
        # binutils import library

sharedlib_from_linklib_result=`func_cygming_dll_for_implib_fallback_co
re '.idata$7' "$1"`
        elif func_cygming_ms_implib_p "$1" ; then
            # ms-generated import library

sharedlib_from_linklib_result=`func_cygming_dll_for_implib_fallback_co
re '.idata$6' "$1"`
        else
            # unknown

```

```

        sharedlib_from_linklib_result=""
    fi
}

# func_extract_an_archive dir oldlib
func_extract_an_archive ()
{
    $opt_debug
    f_ex_an_ar_dir="$1"; shift
    f_ex_an_ar_oldlib="$1"
    if test "$lock_old_archive_extraction" = yes; then
        lockfile=$f_ex_an_ar_oldlib.lock
        until $opt_dry_run || ln "$proppath" "$lockfile" 2>/dev/null; do
            func_echo "Waiting for $lockfile to be removed"
            sleep 2
        done
    fi
    func_show_eval "(cd \"$f_ex_an_ar_dir\" && $AR x
\"$f_ex_an_ar_oldlib\")" \
        'stat=?; rm -f "$lockfile"; exit $stat'
    if test "$lock_old_archive_extraction" = yes; then
        $opt_dry_run || rm -f "$lockfile"
    fi
    if ($AR t "$f_ex_an_ar_oldlib" | sort | sort -uc >/dev/null 2>&1);
then
    :
else
    func_fatal_error "object name conflicts in archive:
$f_ex_an_ar_dir/$f_ex_an_ar_oldlib"
fi
}

# func_extract_archives gentop oldlib ...
func_extract_archives ()
{
    $opt_debug
    my_gentop="$1"; shift
    my_oldlibs=${1+"$@"}
    my_oldobjs=""
    my_xlib=""
    my_xabs=""
    my_xdir=""

    for my_xlib in $my_oldlibs; do
        # Extract the objects.
        case $my_xlib in
            [\\/] * | [A-Za-z]:[\\/] *) my_xabs="$my_xlib" ;;
            *) my_xabs=`pwd`"/$my_xlib" ;;
        esac
        func_basename "$my_xlib"
    done
}

```

```

my_xlib="$func_basename_result"
my_xlib_u=$my_xlib
while ;; do
    case " $extracted_archives " in
*) $my_xlib_u *)
    func_arith $extracted_serial + 1
    extracted_serial=$func_arith_result
    my_xlib_u=lt$extracted_serial-$my_xlib ;;
*) break ;;
esac
done
extracted_archives="$extracted_archives $my_xlib_u"
my_xdir="$my_gentop/$my_xlib_u"

func_mkdir_p "$my_xdir"

case $host in
*-darwin*)
func_verbose "Extracting $my_xabs"
# Do not bother doing anything if just a dry run
$opt_dry_run || {
    darwin_orig_dir=`pwd`
    cd $my_xdir || exit $?
    darwin_archive=$my_xabs
    darwin_curdir=`pwd`
    darwin_base_archive=`basename "$darwin_archive"`
    darwin_arches=`$LIPO -info "$darwin_archive" 2>/dev/null |
$GREP Architectures 2>/dev/null || true`
    if test -n "$darwin_arches"; then
        darwin_arches=`$ECHO "$darwin_arches" | $SED -e 's/.*are:/'`
        darwin_arch=
        func_verbose "$darwin_base_archive has multiple architectures
$darwin_arches"
        for darwin_arch in $darwin_arches ; do
            func_mkdir_p "unfat-$$/${darwin_base_archive}-
${darwin_arch}"
            $LIPO -thin $darwin_arch -output "unfat-
$$/${darwin_base_archive}-${darwin_arch}/${darwin_base_archive}"
            "${darwin_archive}"
            cd "unfat-$$/${darwin_base_archive}-${darwin_arch}"
            func_extract_an_archive "`pwd`" "${darwin_base_archive}"
            cd "$darwin_curdir"
            $RM "unfat-$$/${darwin_base_archive}-
${darwin_arch}/${darwin_base_archive}"
        done # $darwin_arches
        ## Okay now we've a bunch of thin objects, gotta fatten
them up :)
        darwin_filelist=`find unfat-$$ -type f -name \*.o -print -o -
name \*.lo -print | $SED -e "$basename" | sort -u`
        darwin_file=
        darwin_files=
        for darwin_file in $darwin_filelist; do

```

```

        darwin_files=`find unfat-$$ -name $darwin_file -print |
sort | $NL2SP`
        $LIPO -create -output "$darwin_file" $darwin_files
done # $darwin_filelist
$RM -rf unfat-$$
cd "$darwin_orig_dir"
else
cd $darwin_orig_dir
func_extract_an_archive "$my_xdir" "$my_xabs"
fi # $darwin_arches
} # !$opt_dry_run
;;
*)
func_extract_an_archive "$my_xdir" "$my_xabs"
;;
esac
my_oldobjs="$my_oldobjs "`find $my_xdir -name \*.so -print
-o -name \*.lo -print | sort | $NL2SP`
done

func_extract_archives_result="$my_oldobjs"
}

```

```

# func_emit_wrapper [arg=no]
#
# Emit a libtool wrapper script on stdout.
# Don't directly open a file because we may want to
# incorporate the script contents within a cygwin/mingw
# wrapper executable. Must ONLY be called from within
# func_mode_link because it depends on a number of variables
# set therein.
#
# ARG is the value that the WRAPPER_SCRIPT_BELONGS_IN_OBJDIR
# variable will take. If 'yes', then the emitted script
# will assume that the directory in which it is stored is
# the $objdir directory. This is a cygwin/mingw-specific
# behavior.
func_emit_wrapper ()
{
    func_emit_wrapper_arg1=${1-no}

    $ECHO "\
#! $SHELL

# $output - temporary wrapper script for $objdir/$outputname
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION
#
# The $output program cannot be directly executed until all the
libtool
# libraries that it depends on are installed.
#

```

```

# This wrapper script should never be moved out of the build
directory.
# If it is, it will not operate correctly.

# Sed substitution that helps us do robust quoting. It backslashifies
# metacharacters that are still active within double-quoted strings.
sed_quote_subst='$sed_quote_subst'

# Be Bourne compatible
if test -n \"\${ZSH_VERSION+set}\" && (emulate sh) >/dev/null 2>&1;
then
  emulate sh
  NULLCMD=:
  # Zsh 3.x and 4.x performs word splitting on \"${1+\"$@\"}\", which
  # is contrary to our usage. Disable this feature.
  alias -g \"\${1+\"$@\"}\"='\"$@\"'
  setopt NO_GLOB_SUBST
else
  case \"(set -o) 2>/dev/null\" in *posix*) set -o posix;; esac
fi
BIN_SH=xpg4; export BIN_SH # for Tru64
DUALCASE=1; export DUALCASE # for MKS sh

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

relink_command=\"\${relink_command}\"

# This environment variable determines our operation mode.
if test \"\${libtool_install_magic}\" = \"\${magic}\"; then
  # install mode needs the following variables:
  generated_by_libtool_version='\${macro_version}'
  notinst_deplibs='\${notinst_deplibs}'
else
  # When we are sourced in execute mode, \"file\" and \"ECHO\" are already
  set.
  if test \"\${libtool_execute_magic}\" != \"\${magic}\"; then
    file=\"\${0}\"

    qECHO=`ECHO \"$ECHO\" | SED \"$sed_quote_subst\"`
    ECHO \"\

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
  eval 'cat <<_LTECHO_EOF'
  \"\$1
  _LTECHO_EOF'
}
  ECHO=\"\${qECHO}\"
fi

```

```

# Very basic option parsing. These options are (a) specific to
# the libtool wrapper, (b) are identical between the wrapper
# /script/ and the wrapper /executable/ which is used only on
# windows platforms, and (c) all begin with the string "--lt-"
# (application programs are unlikely to have options which match
# this pattern).
#
# There are only two supported options: --lt-debug and
# --lt-dump-script. There is, deliberately, no --lt-help.
#
# The first argument to this parsing function should be the
# script's $0 value, followed by "$@".
lt_option_debug=
func_parse_lt_options ()
{
    lt_script_arg0=\$0
    shift
    for lt_opt
    do
        case "\$lt_opt" in
            --lt-debug) lt_option_debug=1 ;;
            --lt-dump-script)
                lt_dump_D=\`\$ECHO \$X\$lt_script_arg0\` | \$SED -e 's/^X//' -e
                's%/[^\^]*$%%'\`
                test "\$X\$lt_dump_D\` = "\$X\$lt_script_arg0\` && lt_dump_D=.
                lt_dump_F=\`\$ECHO \$X\$lt_script_arg0\` | \$SED -e 's/^X//' -e
                's%^.*/%%\`'\`
                cat "\$lt_dump_D/\$lt_dump_F\`
                exit 0
                ;;
            --lt-*)
                \$ECHO \$"Unrecognized --lt- option: '\$lt_opt'\` 1>&2
                exit 1
                ;;
            esac
        done

        # Print the debug banner immediately:
        if test -n "\$lt_option_debug"; then
            echo "\${outputname}:\${output}:\${LINENO}: libtool wrapper (GNU
            \$PACKAGE$TIMESTAMP) $VERSION\` 1>&2
        fi
    }

# Used when --lt-debug. Prints its arguments to stdout
# (redirection is the responsibility of the caller)
func_lt_dump_args ()
{
    lt_dump_args_N=1;
    for lt_arg
    do

```

```

        \${ECHO} \("${outputname}:${output}:\${LINENO}):
newargv[\${lt_dump_args_N}]: \${lt_arg}\
        lt_dump_args_N=\`expr \${lt_dump_args_N} + 1\`
    done
}

# Core function for launching the target application
func_exec_program_core ()
{
"
    case $host in
    # Backslashes separate directories on plain windows
    *-*-mingw | *-*-os2* | *-*cegcc*)
        \${ECHO} "\
            if test -n \("${lt_option_debug}"; then
                \${ECHO} \("${outputname}:${output}:\${LINENO}): newargv[0]:
\${progdir}\${lt_program}" 1>&2
                func_lt_dump_args \${1+\ "${lt_arg}" } 1>&2
            fi
            exec \("${progdir}\${lt_program}" \${1+\ "${lt_arg}" }
"
        ;;

    *)
        \${ECHO} "\
            if test -n \("${lt_option_debug}"; then
                \${ECHO} \("${outputname}:${output}:\${LINENO}): newargv[0]:
\${progdir}/\${lt_program}" 1>&2
                func_lt_dump_args \${1+\ "${lt_arg}" } 1>&2
            fi
            exec \("${progdir}/\${lt_program}" \${1+\ "${lt_arg}" }
"
        ;;
    esac
    \${ECHO} "\
        \${ECHO} \("${0}: cannot exec \${lt_program} \${*}" 1>&2
        exit 1
    }

# A function to encapsulate launching the target application
# Strips options in the --lt-* namespace from \${lt_arg} and
# launches target application with the remaining arguments.
func_exec_program ()
{
    case \ "${lt_arg} " in
    *\\ --lt-*)
        for lt_wr_arg
        do
            case \${lt_wr_arg} in
            --lt-*) ;;
            *) set x \ "${lt_arg}" \ "${lt_wr_arg}"; shift;;
            esac
        done
    esac
}

```

```

        shift
    done ;;
esac
func_exec_program_core \${1+"\$@"}
}

# Parse options
func_parse_lt_options "\$0" \${1+"\$@"}

# Find the directory that this script lives in.
thisdir=\\$ECHO "\\$file" | $SED 's%/[^/]*$%%'\`
test "x$thisdir" = "x$file" && thisdir=.

# Follow symbolic links until we get to the real thisdir.
file=\\ls -ld "\\$file" | $SED -n 's/.*-> //p'\`
while test -n "\\$file"; do
    destdir=\\$ECHO "\\$file" | $SED 's%/[^/]*$%%'\`

    # If there was a directory component, then change thisdir.
    if test "x$destdir" != "x$file"; then
        case "\\$destdir" in
            [\\\\/]* | [A-Za-z]:[\\\\/]* ) thisdir= "\\$destdir" ;;
            *) thisdir= "\\$thisdir/$destdir" ;;
        esac
    fi

    file=\\$ECHO "\\$file" | $SED 's%^.*/%%%'\`
    file=\\ls -ld "\\$thisdir/$file" | $SED -n 's/.*-> //p'\`
done

# Usually 'no', except on cygwin/mingw when embedded into
# the cwrapper.
WRAPPER_SCRIPT_BELONGS_IN_OBJDIR=$func_emit_wrapper_arg1
if test "\\$WRAPPER_SCRIPT_BELONGS_IN_OBJDIR" = "yes"; then
    # special case for '.'
    if test "\\$thisdir" = "."; then
        thisdir=\\pwd\`
    fi
    # remove .libs from thisdir
    case "\\$thisdir" in
        *[[\\\\/]]$objdir ) thisdir=\\$ECHO "\\$thisdir" | $SED
's%[[\\\\/]][^\\\\/]*$%%'\` ;;
        $objdir ) thisdir=. ;;
    esac
fi

# Try to get the absolute directory name.
absdir=\\cd "\\$thisdir" && pwd\`
test -n "\\$absdir" && thisdir= "\\$absdir"
"

    if test "$fast_install" = yes; then

```



```

        $ECHO "\
program=lt-'$outputname'$exeext
progdir=\"\$thisdir/$objdir\"

if test ! -f \"\$progdir/\$program\" ||
  { file=`ls -ldt \"\$progdir/\$program\"
\"\$progdir/./\$program\" 2>/dev/null | ${SED} 1q\`; \
  test \"X$file\" != \"X\$progdir/\$program\"; }; then

file=\"\$\$-\$program\"

if test ! -d \"\$progdir\"; then
  $MKDIR \"\$progdir\"
else
  $RM \"\$progdir/\$file\"
fi

        $ECHO "\

# relink executable if necessary
if test -n \"\$relink_command\"; then
  if relink_command_output=`eval \$relink_command 2>&1\`; then :
  else
    $ECHO \"\$relink_command_output\" >&2
    $RM \"\$progdir/\$file\"
    exit 1
  fi
fi

$MV \"\$progdir/\$file\" \"\$progdir/\$program\" 2>/dev/null ||
{ $RM \"\$progdir/\$program\";
  $MV \"\$progdir/\$file\" \"\$progdir/\$program\"; }
$RM \"\$progdir/\$file\"
fi"
else
  $ECHO "\
program='$outputname'
progdir=\"\$thisdir/$objdir\"
"

  fi

  $ECHO "\

if test -f \"\$progdir/\$program\"; then"

  # fixup the dll searchpath if we need to.
  #
  # Fix the DLL searchpath if we need to. Do this before
prepending
  # to shlibpath, because on Windows, both are PATH and uninstalled
  # libraries must come first.
  if test -n "$dllsearchpath"; then

```

```

    $ECHO "\
# Add the dll search path components to the executable PATH
PATH=$dllsearchpath:\$PATH
"
    fi

    # Export our shlibpath_var if we have one.
    if test "$shlibpath_overrides_runpath" = yes && test -n
"$shlibpath_var" && test -n "$temp_rpath"; then
        $ECHO "\
# Add our own library path to $shlibpath_var
shlibpath_var=\"$temp_rpath\$$shlibpath_var\"

# Some systems cannot cope with colon-terminated $shlibpath_var
# The second colon is a workaround for a bug in BeOS R4 sed
shlibpath_var=\"\${ECHO \"\$$shlibpath_var\" | $SED 's/::*\$//'\`

export $shlibpath_var
"
    fi

    $ECHO "\
if test \"\$libtool_execute_magic\" != \"\$magic\"; then
    # Run the actual program with our arguments.
    func_exec_program \${1+\\"$@"}
fi
else
    # The program doesn't exist.
    \${ECHO \"\${0}: error: \\\"\\$progdir/\$program' does not exist\"
1>&2
    \${ECHO \"This script is just a wrapper for \$program.\" 1>&2
    \${ECHO \"See the $PACKAGE documentation for more information.\"
1>&2
    exit 1
fi
fi\
"
}

# func_emit_cwrappersource
# emit the source code for a wrapper executable on stdout
# Must ONLY be called from within func_mode_link because
# it depends on a number of variable set therein.
func_emit_cwrappersource ()
{
    cat <<EOF

/* $cwrappersource - temporary wrapper executable for
$objdir/$outputname
Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION

```

The \$output program cannot be directly executed until all the libtool libraries that it depends on are installed.

This wrapper executable should never be moved out of the build directory.

If it is, it will not operate correctly.

*/

EOF

```
        cat <<"EOF"
#ifdef _MSC_VER
# define _CRT_SECURE_NO_DEPRECATED 1
#endif
#include <stdio.h>
#include <stdlib.h>
#ifdef _MSC_VER
# include <direct.h>
# include <process.h>
# include <io.h>
#else
# include <unistd.h>
# include <stdint.h>
# ifdef __CYGWIN__
# include <io.h>
# endif
#endif
#include <malloc.h>
#include <stdarg.h>
#include <assert.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/stat.h>

/* declarations of non-ANSI functions */
#ifdef __MINGW32__
# ifdef __STRICT_ANSI__
int _putenv (const char *);
# endif
#elif defined(__CYGWIN__)
# ifdef __STRICT_ANSI__
char *realpath (const char *, char *);
int putenv (char *);
int setenv (const char *, const char *, int);
# endif
/* #elif defined (other platforms) ... */
#endif

/* portability defines, excluding path handling macros */
#ifdef _MSC_VER
# define setmode _setmode
```

```

# define stat      _stat
# define chmod    _chmod
# define getcwd   _getcwd
# define putenv   _putenv
# define S_IXUSR  _S_IXEXEC
# ifndef _INTPTR_T_DEFINED
#   define _INTPTR_T_DEFINED
#   define intptr_t int
# endif
#elif defined(__MINGW32__)
# define setmode  _setmode
# define stat     _stat
# define chmod    _chmod
# define getcwd   _getcwd
# define putenv   _putenv
#elif defined(__CYGWIN__)
# define HAVE_SETENV
# define FOPEN_WB "wb"
/* #elif defined (other platforms) ... */
#endif

#if defined(PATH_MAX)
# define LT_PATHMAX PATH_MAX
#elif defined(MAXPATHLEN)
# define LT_PATHMAX MAXPATHLEN
#else
# define LT_PATHMAX 1024
#endif

#ifndef S_IXOTH
# define S_IXOTH 0
#endif
#ifndef S_IXGRP
# define S_IXGRP 0
#endif

/* path handling portability macros */
#ifndef DIR_SEPARATOR
# define DIR_SEPARATOR '/'
# define PATH_SEPARATOR ':'
#endif

#if defined (_WIN32) || defined (__MSDOS__) || defined (__DJGPP__) ||
\
  defined (__OS2__)
# define HAVE_DOS_BASED_FILE_SYSTEM
# define FOPEN_WB "wb"
# ifndef DIR_SEPARATOR_2
#   define DIR_SEPARATOR_2 '\\\
# endif
# ifndef PATH_SEPARATOR_2
#   define PATH_SEPARATOR_2 ';'

```

```

# endif
#endif

#ifndef DIR_SEPARATOR_2
# define IS_DIR_SEPARATOR(ch) ((ch) == DIR_SEPARATOR)
#else /* DIR_SEPARATOR_2 */
# define IS_DIR_SEPARATOR(ch) \
    (((ch) == DIR_SEPARATOR) || ((ch) == DIR_SEPARATOR_2))
#endif /* DIR_SEPARATOR_2 */

#ifndef PATH_SEPARATOR_2
# define IS_PATH_SEPARATOR(ch) ((ch) == PATH_SEPARATOR)
#else /* PATH_SEPARATOR_2 */
# define IS_PATH_SEPARATOR(ch) ((ch) == PATH_SEPARATOR_2)
#endif /* PATH_SEPARATOR_2 */

#ifndef FOPEN_WB
# define FOPEN_WB "w"
#endif
#ifndef _O_BINARY
# define _O_BINARY 0
#endif

#define XMMALLOC(type, num) ((type *) xmalloc ((num) *
sizeof(type)))
#define XFREE(stale) do { \
    if (stale) { free ((void *) stale); stale = 0; } \
} while (0)

#ifdef LT_DEBUGWRAPPER
static int lt_debug = 1;
#else
static int lt_debug = 0;
#endif

const char *program_name = "libtool-wrapper"; /* in case xstrdup fails
*/

void *xmalloc (size_t num);
char *xstrdup (const char *string);
const char *base_name (const char *name);
char *find_executable (const char *wrapper);
char *chase_symlinks (const char *pathspec);
int make_executable (const char *path);
int check_executable (const char *path);
char *strendzap (char *str, const char *pat);
void lt_debugprintf (const char *file, int line, const char *fmt,
...);
void lt_fatal (const char *file, int line, const char *message, ...);
static const char *nonnull (const char *s);
static const char *nonempty (const char *s);
void lt_setenv (const char *name, const char *value);

```

```

char *lt_extend_str (const char *orig_value, const char *add, int
to_end);
void lt_update_exe_path (const char *name, const char *value);
void lt_update_lib_path (const char *name, const char *value);
char **prepare_spawn (char **argv);
void lt_dump_script (FILE *f);
EOF

        cat <<EOF
volatile const char * MAGIC_EXE = "$magic_exe";
const char * LIB_PATH_VARNAME = "$shlibpath_var";
EOF

        if test "$shlibpath_overrides_runpath" = yes && test -n
"$shlibpath_var" && test -n "$temp_rpath"; then
            func_to_host_path "$temp_rpath"
            cat <<EOF
const char * LIB_PATH_VALUE      = "$func_to_host_path_result";
EOF
        else
            cat <<"EOF"
const char * LIB_PATH_VALUE      = "";
EOF
        fi

        if test -n "$dllsearchpath"; then
            func_to_host_path "$dllsearchpath:"
            cat <<EOF
const char * EXE_PATH_VARNAME = "PATH";
const char * EXE_PATH_VALUE  = "$func_to_host_path_result";
EOF
        else
            cat <<"EOF"
const char * EXE_PATH_VARNAME = "";
const char * EXE_PATH_VALUE  = "";
EOF
        fi

        if test "$fast_install" = yes; then
            cat <<EOF
const char * TARGET_PROGRAM_NAME = "lt-$outputname"; /* hopefully, no
.exe */
EOF
        else
            cat <<EOF
const char * TARGET_PROGRAM_NAME = "$outputname"; /* hopefully, no
.exe */
EOF
        fi

        cat <<"EOF"

```

```

#define LTWRAPPER_OPTION_PREFIX          "--lt-"

static const char *ltwrapper_option_prefix = LTWRAPPER_OPTION_PREFIX;
static const char *dumpscrip_opt         = LTWRAPPER_OPTION_PREFIX
"dump-script";
static const char *debug_opt             = LTWRAPPER_OPTION_PREFIX
"debug";

int
main (int argc, char *argv[])
{
    char **newargz;
    int newargc;
    char *tmp_pathspec;
    char *actual_cwrapper_path;
    char *actual_cwrapper_name;
    char *target_name;
    char *lt_argv_zero;
    intptr_t rval = 127;

    int i;

    program_name = (char *) xstrdup (base_name (argv[0]));
    newargz = XMALLOC (char *, argc + 1);

    /* very simple arg parsing; don't want to rely on getopt
     * also, copy all non cwrapper options to newargz, except
     * argz[0], which is handled differently
     */
    newargc=0;
    for (i = 1; i < argc; i++)
    {
        if (strcmp (argv[i], dumpscrip_opt) == 0)
        {
EOF
            case "$host" in
                *mingw* | *cygwin* )
                    # make stdout use "unix" line endings
                    echo "          setmode(1, _O_BINARY);"
                    ;;
                esac

            cat <<"EOF"
            lt_dump_script (stdout);
            return 0;
        }
        if (strcmp (argv[i], debug_opt) == 0)
        {
            lt_debug = 1;
            continue;
        }
    }
}

```

```

        if (strcmp (argv[i], ltwrapper_option_prefix) == 0)
            {
                /* however, if there is an option in the
LTWRAPPER_OPTION_PREFIX
                namespace, but it is not one of the ones we know about
and
                have already dealt with, above (including dump-script),
then
                report an error. Otherwise, targets might begin to
believe
                they are allowed to use options in the
LTWRAPPER_OPTION_PREFIX
                namespace. The first time any user complains about this,
we'll
                need to make LTWRAPPER_OPTION_PREFIX a configure-time
option
                or a configure.ac-settable value.
                */
                lt_fatal (__FILE__, __LINE__,
                        "unrecognized %s option: '%s'",
                        ltwrapper_option_prefix, argv[i]);
            }
        /* otherwise ... */
        newargz[++newargc] = xstrdup (argv[i]);
    }
    newargz[++newargc] = NULL;

EOF
        cat <<EOF
        /* The GNU banner must be the first non-error debug message */
        lt_debugprintf (__FILE__, __LINE__, "libtool wrapper (GNU
$PACKAGE$TIMESTAMP) $VERSION\n");
EOF
        cat <<"EOF"
        lt_debugprintf (__FILE__, __LINE__, "(main) argv[0]: %s\n",
argv[0]);
        lt_debugprintf (__FILE__, __LINE__, "(main) program_name: %s\n",
program_name);

        tmp_pathspec = find_executable (argv[0]);
        if (tmp_pathspec == NULL)
            lt_fatal (__FILE__, __LINE__, "couldn't find %s", argv[0]);
        lt_debugprintf (__FILE__, __LINE__,
                        "(main) found exe (before symlink chase) at: %s\n",
                        tmp_pathspec);

        actual_cwrapper_path = chase_symlinks (tmp_pathspec);
        lt_debugprintf (__FILE__, __LINE__,
                        "(main) found exe (after symlink chase) at: %s\n",
                        actual_cwrapper_path);
        XFREE (tmp_pathspec);

```



```

actual_cwrapper_name = xstrdup (base_name (actual_cwrapper_path));
strendzap (actual_cwrapper_path, actual_cwrapper_name);

/* wrapper name transforms */
strendzap (actual_cwrapper_name, ".exe");
tmp_pathspec = lt_extend_str (actual_cwrapper_name, ".exe", 1);
XFREE (actual_cwrapper_name);
actual_cwrapper_name = tmp_pathspec;
tmp_pathspec = 0;

/* target_name transforms -- use actual target program name; might
have lt- prefix */
target_name = xstrdup (base_name (TARGET_PROGRAM_NAME));
strendzap (target_name, ".exe");
tmp_pathspec = lt_extend_str (target_name, ".exe", 1);
XFREE (target_name);
target_name = tmp_pathspec;
tmp_pathspec = 0;

lt_debugprintf (__FILE__, __LINE__,
                "(main) libtool target name: %s\n",
                target_name);
EOF

        cat <<EOF
newargz[0] =
    XMALLOC (char, (strlen (actual_cwrapper_path) +
                    strlen ("${objdir}") + 1 + strlen (actual_cwrapper_name)
+ 1));
strcpy (newargz[0], actual_cwrapper_path);
strcat (newargz[0], "${objdir}");
strcat (newargz[0], "/");
EOF

        cat <<"EOF"
/* stop here, and copy so we don't have to do this twice */
tmp_pathspec = xstrdup (newargz[0]);

/* do NOT want the lt- prefix here, so use actual_cwrapper_name */
strcat (newargz[0], actual_cwrapper_name);

/* DO want the lt- prefix here if it exists, so use target_name */
lt_argv_zero = lt_extend_str (tmp_pathspec, target_name, 1);
XFREE (tmp_pathspec);
tmp_pathspec = NULL;
EOF

        case $host_os in
            mingw*)
                cat <<"EOF"
{
    char* p;

```

```

while ((p = strchr (newargz[0], '\\')) != NULL)
    {
    *p = '/';
    }
while ((p = strchr (lt_argv_zero, '\\')) != NULL)
    {
    *p = '/';
    }
}
EOF

;;
esac

cat <<"EOF"
XFREE (target_name);
XFREE (actual_cwrapper_path);
XFREE (actual_cwrapper_name);

lt_setenv ("BIN_SH", "xpg4"); /* for Tru64 */
lt_setenv ("DUALCASE", "1"); /* for MSK sh */
/* Update the DLL searchpath.  EXE_PATH_VALUE ($dllsearchpath) must
   be prepended before (that is, appear after) LIB_PATH_VALUE
($temp_rpath)
   because on Windows, both *_VARNAMES are PATH but uninstalled
   libraries must come first. */
lt_update_exe_path (EXE_PATH_VARNAME, EXE_PATH_VALUE);
lt_update_lib_path (LIB_PATH_VARNAME, LIB_PATH_VALUE);

lt_debugprintf (__FILE__, __LINE__, "(main) lt_argv_zero: %s\n",
                nonnull (lt_argv_zero));
for (i = 0; i < newargc; i++)
    {
    lt_debugprintf (__FILE__, __LINE__, "(main) newargz[%d]: %s\n",
                    i, nonnull (newargz[i]));
    }
}
EOF

case $host_os in
mingw*)
cat <<"EOF"
/* execv doesn't actually work on mingw as expected on unix */
newargz = prepare_spawn (newargz);
rval = _spawnv (_P_WAIT, lt_argv_zero, (const char * const *)
newargz);
if (rval == -1)
    {
    /* failed to start process */
    lt_debugprintf (__FILE__, __LINE__,
                    "(main) failed to launch target \"%s\": %s\n",
                    lt_argv_zero, nonnull (strerror (errno)));
    return 127;
}

```

```

    }
    return rval;
EOF
    ;;
    *)
    cat <<"EOF"
    execv (lt_argv_zero, newargz);
    return rval; /* =127, but avoids unused variable warning */
EOF
    ;;
    esac

    cat <<"EOF"
}

void *
xmalloc (size_t num)
{
    void *p = (void *) malloc (num);
    if (!p)
        lt_fatal (__FILE__, __LINE__, "memory exhausted");

    return p;
}

char *
xstrdup (const char *string)
{
    return string ? strcpy ((char *) xmalloc (strlen (string) + 1),
        string) : NULL;
}

const char *
base_name (const char *name)
{
    const char *base;

#ifdef HAVE_DOS_BASED_FILE_SYSTEM
    /* Skip over the disk name in MSDOS pathnames. */
    if (isalpha ((unsigned char) name[0]) && name[1] == ':')
        name += 2;
#endif

    for (base = name; *name; name++)
        if (IS_DIR_SEPARATOR (*name))
            base = name + 1;
    return base;
}

int
check_executable (const char *path)
{

```

```

struct stat st;

lt_debugprintf (__FILE__, __LINE__, "(check_executable): %s\n",
                nonempty (path));
if ((!path) || (!*path))
    return 0;

if ((stat (path, &st) >= 0)
    && (st.st_mode & (S_IXUSR | S_IXGRP | S_IXOTH)))
    return 1;
else
    return 0;
}

int
make_executable (const char *path)
{
    int rval = 0;
    struct stat st;

    lt_debugprintf (__FILE__, __LINE__, "(make_executable): %s\n",
                    nonempty (path));
    if ((!path) || (!*path))
        return 0;

    if (stat (path, &st) >= 0)
        {
            rval = chmod (path, st.st_mode | S_IXOTH | S_IXGRP | S_IXUSR);
        }
    return rval;
}

/* Searches for the full path of the wrapper. Returns
   newly allocated full path name if found, NULL otherwise
   Does not chase symlinks, even on platforms that support them.
*/
char *
find_executable (const char *wrapper)
{
    int has_slash = 0;
    const char *p;
    const char *p_next;
    /* static buffer for getcwd */
    char tmp[LT_PATHMAX + 1];
    int tmp_len;
    char *concat_name;

    lt_debugprintf (__FILE__, __LINE__, "(find_executable): %s\n",
                    nonempty (wrapper));

    if ((wrapper == NULL) || (*wrapper == '\\0'))
        return NULL;

```

```

/* Absolute path? */
#if defined (HAVE_DOS_BASED_FILE_SYSTEM)
    if (isalpha ((unsigned char) wrapper[0]) && wrapper[1] == ':')
    {
        concat_name = xstrdup (wrapper);
        if (check_executable (concat_name))
            return concat_name;
        XFREE (concat_name);
    }
    else
    {
#endif
        if (IS_DIR_SEPARATOR (wrapper[0]))
        {
            concat_name = xstrdup (wrapper);
            if (check_executable (concat_name))
                return concat_name;
            XFREE (concat_name);
        }
#endif
#if defined (HAVE_DOS_BASED_FILE_SYSTEM)
    }
#endif

for (p = wrapper; *p; p++)
    if (*p == '/')
    {
        has_slash = 1;
        break;
    }
if (!has_slash)
{
    /* no slashes; search PATH */
    const char *path = getenv ("PATH");
    if (path != NULL)
    {
        for (p = path; *p; p = p_next)
        {
            const char *q;
            size_t p_len;
            for (q = p; *q; q++)
                if (IS_PATH_SEPARATOR (*q))
                    break;
            p_len = q - p;
            p_next = (*q == '\\0' ? q : q + 1);
            if (p_len == 0)
            {
                /* empty path: current directory */
                if (getcwd (tmp, LT_PATHMAX) == NULL)
                    lt_fatal (__FILE__, __LINE__, "getcwd failed: %s",
                                nonnull (strerror (errno)));
                tmp_len = strlen (tmp);
            }
        }
    }
}

```

```

        concat_name =
            XMALLOC (char, tmp_len + 1 + strlen (wrapper) + 1);
        memcpy (concat_name, tmp, tmp_len);
        concat_name[tmp_len] = '/';
        strcpy (concat_name + tmp_len + 1, wrapper);
    }
    else
    {
        concat_name =
            XMALLOC (char, p_len + 1 + strlen (wrapper) + 1);
        memcpy (concat_name, p, p_len);
        concat_name[p_len] = '/';
        strcpy (concat_name + p_len + 1, wrapper);
    }
    if (check_executable (concat_name))
        return concat_name;
    XFREE (concat_name);
}
}
/* not found in PATH; assume curdir */
}
/* Relative path | not found in path: prepend cwd */
if (getcwd (tmp, LT_PATHMAX) == NULL)
    lt_fatal (__FILE__, __LINE__, "getcwd failed: %s",
              nonnull (strerror (errno)));
tmp_len = strlen (tmp);
concat_name = XMALLOC (char, tmp_len + 1 + strlen (wrapper) + 1);
memcpy (concat_name, tmp, tmp_len);
concat_name[tmp_len] = '/';
strcpy (concat_name + tmp_len + 1, wrapper);

if (check_executable (concat_name))
    return concat_name;
XFREE (concat_name);
return NULL;
}

char *
chase_symlinks (const char *pathspec)
{
#ifdef S_ISLNK
    return xstrdup (pathspec);
#else
    char buf[LT_PATHMAX];
    struct stat s;
    char *tmp_pathspec = xstrdup (pathspec);
    char *p;
    int has_symlinks = 0;
    while (strlen (tmp_pathspec) && !has_symlinks)
    {
        lt_debugprintf (__FILE__, __LINE__,
                        "checking path component for symlinks: %s\n",

```

```

        tmp_pathspec);
if (lstat (tmp_pathspec, &s) == 0)
{
    if (S_ISLNK (s.st_mode) != 0)
        {
            has_symlinks = 1;
            break;
        }

    /* search backwards for last DIR_SEPARATOR */
    p = tmp_pathspec + strlen (tmp_pathspec) - 1;
    while ((p > tmp_pathspec) && (!IS_DIR_SEPARATOR (*p)))
        p--;
    if ((p == tmp_pathspec) && (!IS_DIR_SEPARATOR (*p)))
        {
            /* no more DIR_SEPARATORS left */
            break;
        }
    *p = '\\0';
}
else
{
    lt_fatal (__FILE__, __LINE__,
              "error accessing file \"%s\": %s",
              tmp_pathspec, nonnull (strerror (errno)));
}
}
XFREE (tmp_pathspec);

if (!has_symlinks)
{
    return xstrdup (pathspec);
}

tmp_pathspec = realpath (pathspec, buf);
if (tmp_pathspec == 0)
{
    lt_fatal (__FILE__, __LINE__,
              "could not follow symlinks for %s", pathspec);
}
return xstrdup (tmp_pathspec);
#endif
}

char *
strendzap (char *str, const char *pat)
{
    size_t len, patlen;

    assert (str != NULL);
    assert (pat != NULL);

```

```

len = strlen (str);
patlen = strlen (pat);

if (patlen <= len)
{
    str += len - patlen;
    if (strcmp (str, pat) == 0)
        *str = '\0';
}
return str;
}

void
lt_debugprintf (const char *file, int line, const char *fmt, ...)
{
    va_list args;
    if (lt_debug)
    {
        (void) fprintf (stderr, "%s:%s:%d: ", program_name, file, line);
        va_start (args, fmt);
        (void) vfprintf (stderr, fmt, args);
        va_end (args);
    }
}

static void
lt_error_core (int exit_status, const char *file,
               int line, const char *mode,
               const char *message, va_list ap)
{
    fprintf (stderr, "%s:%s:%d: %s: ", program_name, file, line, mode);
    vfprintf (stderr, message, ap);
    fprintf (stderr, ".\n");

    if (exit_status >= 0)
        exit (exit_status);
}

void
lt_fatal (const char *file, int line, const char *message, ...)
{
    va_list ap;
    va_start (ap, message);
    lt_error_core (EXIT_FAILURE, file, line, "FATAL", message, ap);
    va_end (ap);
}

static const char *
nonnull (const char *s)
{
    return s ? s : "(null)";
}

```



```

static const char *
nonempty (const char *s)
{
    return (s && !*s) ? "(empty)" : nonnull (s);
}

void
lt_setenv (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
                    "(lt_setenv) setting '%s' to '%s'\n",
                    nonnull (name), nonnull (value));
    {
#ifdef HAVE_SETENV
        /* always make a copy, for consistency with !HAVE_SETENV */
        char *str = xstrdup (value);
        setenv (name, str, 1);
#else
        int len = strlen (name) + 1 + strlen (value) + 1;
        char *str = XMALLOC (char, len);
        sprintf (str, "%s=%s", name, value);
        if (putenv (str) != EXIT_SUCCESS)
            {
                XFREE (str);
            }
#endif
    }
}

char *
lt_extend_str (const char *orig_value, const char *add, int to_end)
{
    char *new_value;
    if (orig_value && *orig_value)
        {
            int orig_value_len = strlen (orig_value);
            int add_len = strlen (add);
            new_value = XMALLOC (char, add_len + orig_value_len + 1);
            if (to_end)
                {
                    strcpy (new_value, orig_value);
                    strcpy (new_value + orig_value_len, add);
                }
            else
                {
                    strcpy (new_value, add);
                    strcpy (new_value + add_len, orig_value);
                }
        }
    else
        {

```

```

        new_value = xstrdup (add);
    }
    return new_value;
}

void
lt_update_exe_path (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
        "(lt_update_exe_path) modifying '%s' by prepending
'%s'\n",
        nonnull (name), nonnull (value));

    if (name && *name && value && *value)
    {
        char *new_value = lt_extend_str (getenv (name), value, 0);
        /* some systems can't cope with a ':'-terminated path '#' */
        int len = strlen (new_value);
        while (((len = strlen (new_value)) > 0) && IS_PATH_SEPARATOR
(new_value[len-1]))
        {
            new_value[len-1] = '\0';
        }
        lt_setenv (name, new_value);
        XFREE (new_value);
    }
}

void
lt_update_lib_path (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
        "(lt_update_lib_path) modifying '%s' by prepending
'%s'\n",
        nonnull (name), nonnull (value));

    if (name && *name && value && *value)
    {
        char *new_value = lt_extend_str (getenv (name), value, 0);
        lt_setenv (name, new_value);
        XFREE (new_value);
    }
}

EOF

    case $host_os in
        mingw*)
            cat <<"EOF"

/* Prepares an argument vector before calling spawn().
Note that spawn() does not by itself call the command interpreter
(getenv ("COMSPEC") != NULL ? getenv ("COMSPEC") :
```

```

    ({ OSVERSIONINFO v; v.dwOSVersionInfoSize =
sizeof(OSVERSIONINFO);
    GetVersionEx(&v);
    v.dwPlatformId == VER_PLATFORM_WIN32_NT;
    }) ? "cmd.exe" : "command.com").
    Instead it simply concatenates the arguments, separated by ' ', and
calls
    CreateProcess(). We must quote the arguments since Win32
CreateProcess()
    interprets characters like ' ', '\t', '\\', '"' (but not '<' and
'>') in a
    special way:
    - Space and tab are interpreted as delimiters. They are not treated
as
    delimiters if they are surrounded by double quotes: "...".
    - Unescaped double quotes are removed from the input. Their only
effect is
    that within double quotes, space and tab are treated like normal
characters.
    - Backslashes not followed by double quotes are not special.
    - But 2*n+1 backslashes followed by a double quote become
n backslashes followed by a double quote (n >= 0):
    \" -> "
    \\\" -> \"
    \\\\" -> \\\"
    \\\\\" -> \\\\"
*/
#define SHELL_SPECIAL_CHARS "\"\\
\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017\020\021\0
22\023\024\025\026\027\030\031\032\033\034\035\036\037"
#define SHELL_SPACE_CHARS "
\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017\020\021\0
22\023\024\025\026\027\030\031\032\033\034\035\036\037"
char **
prepare_spawn (char **argv)
{
    size_t argc;
    char **new_argv;
    size_t i;

    /* Count number of arguments. */
    for (argc = 0; argv[argc] != NULL; argc++)
        ;

    /* Allocate new argument vector. */
    new_argv = XMALLOC (char *, argc + 1);

    /* Put quoted arguments into the new argument vector. */
    for (i = 0; i < argc; i++)
    {
        const char *string = argv[i];

        if (string[0] == '\0')

```

```

new_argv[i] = xstrdup ("\"");
else if (strpbrk (string, SHELL_SPECIAL_CHARS) != NULL)
{
int quote_around = (strpbrk (string, SHELL_SPACE_CHARS) !=
NULL);
size_t length;
unsigned int backslashes;
const char *s;
char *quoted_string;
char *p;

length = 0;
backslashes = 0;
if (quote_around)
length++;
for (s = string; *s != '\\0'; s++)
{
char c = *s;
if (c == '"')
length += backslashes + 1;
length++;
if (c == '\\')
backslashes++;
else
backslashes = 0;
}
if (quote_around)
length += backslashes + 1;

quoted_string = XMALLOC (char, length + 1);

p = quoted_string;
backslashes = 0;
if (quote_around)
*p++ = '"';
for (s = string; *s != '\\0'; s++)
{
char c = *s;
if (c == '"')
{
unsigned int j;
for (j = backslashes + 1; j > 0; j--)
*p++ = '\\';
}
*p++ = c;
if (c == '\\')
backslashes++;
else
backslashes = 0;
}
if (quote_around)
{

```

```

        unsigned int j;
        for (j = backslashes; j > 0; j--)
            *p++ = '\\';
            *p++ = '"';
        }
        *p = '\\0';

        new_argv[i] = quoted_string;
    }
    else
        new_argv[i] = (char *) string;
    }
new_argv[argc] = NULL;

return new_argv;
}
EOF

        ;;
    esac

        cat <<"EOF"
void lt_dump_script (FILE* f)
{
EOF
        func_emit_wrapper yes |
        $SED -n -e '
s/^\(.\{79\}\)\(.*\)/\1\
\2/
h
s/\(\\\"\\)\(.*\)/\1/g
s/\$/\n/
s/\(^[^n]*\).*/ fputs ("\1", f);/p
g
D'
        cat <<"EOF"
}
EOF
}
# end: func_emit_cwrappers_src

# func_win32_import_lib_p ARG
# True if ARG is an import lib, as indicated by $file_magic_cmd
func_win32_import_lib_p ()
{
    $opt_debug
    case `eval $file_magic_cmd "\$1" 2>/dev/null | $SED -e 10q` in
        *import*) : ;;
        *) false ;;
    esac
}

# func_mode_link arg...

```

```

func_mode_link ()
{
    $opt_debug
    case $host in
        *-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-cegcc*)
            # It is impossible to link a dll without this setting, and
            # we shouldn't force the makefile maintainer to figure out
            # which system we are compiling for in order to pass an extra
            # flag for every libtool invocation.
            # allow_undefined=no

            # FIXME: Unfortunately, there are problems with the above when
trying
            # to make a dll which has undefined symbols, in which case not
            # even a static library is built. For now, we need to specify
            # -no-undefined on the libtool link line when we can be certain
            # that all symbols are satisfied, otherwise we get a static
library.
            allow_undefined=yes
            ;;
        *)
            allow_undefined=yes
            ;;
    esac
    libtool_args=$nonopt
    base_compile="$nonopt $@"
    compile_command=$nonopt
    finalize_command=$nonopt

    compile_rpath=
    finalize_rpath=
    compile_shlibpath=
    finalize_shlibpath=
    convenience=
    old_convenience=
    deplibs=
    old_deplibs=
    compiler_flags=
    linker_flags=
    dllsearchpath=
    lib_search_path=`pwd`
    inst_prefix_dir=
    new_inherited_linker_flags=

    avoid_version=no
    bindir=
    dlfiles=
    dlprefiles=
    dlself=no
    export_dynamic=no
    export_symbols=
    export_symbols_regex=

```

```

generated=
libobjs=
ltlibs=
module=no
no_install=no
objs=
non_pic_objects=
precious_files_regex=
prefer_static_libs=no
preload=no
prev=
prevarg=
release=
rpath=
xrpath=
perm_rpath=
temp_rpath=
thread_safe=no
vinfo=
vinfo_number=no
weak_libs=
single_module="{wl}-single_module"
func_infer_tag $base_compile

# We need to know -static, to get the right output filenames.
for arg
do
    case $arg in
        -shared)
            test "$build_libtool_libs" != yes && \
                func_fatal_configuration "can not build a shared library"
            build_old_libs=no
            break
        ;;
        -all-static | -static | -static-libtool-libs)
            case $arg in
                -all-static)
                    if test "$build_libtool_libs" = yes && test -z
"$link_static_flag"; then
                        func_warning "complete static linking is impossible in this
configuration"
                    fi
                    if test -n "$link_static_flag"; then
                        dlopen_self=$dlopen_self_static
                    fi
                    prefer_static_libs=yes
                ;;
                -static)
                    if test -z "$pic_flag" && test -n "$link_static_flag"; then
                        dlopen_self=$dlopen_self_static
                    fi
                    prefer_static_libs=built
            esac
        ;;
    esac
done

```

```

    ;;
-static-libtool-libs)
    if test -z "$pic_flag" && test -n "$link_static_flag"; then
        dlopen_self=$dlopen_self_static
    fi
    prefer_static_libs=yes
    ;;
esac
build_libtool_libs=no
build_old_libs=yes
break
;;
esac
done

# See if our shared archives depend on static archives.
test -n "$old_archive_from_new_cmds" && build_old_libs=yes

# Go through the arguments, transforming them on the way.
while test "$#" -gt 0; do
    arg="$1"
    shift
    func_quote_for_eval "$arg"
    qarg=$func_quote_for_eval_unquoted_result
    libtool_args+=" $func_quote_for_eval_result"

    # If the previous option needs an argument, assign it.
    if test -n "$prev"; then
    case $prev in
output)
        compile_command+=" @OUTPUT@"
        finalize_command+=" @OUTPUT@"
        ;;
esac

case $prev in
bindir)
    bindir="$arg"
    prev=
    continue
    ;;
dlfiles|dlprefiles)
    if test "$preload" = no; then
        # Add the symbol object into the linking commands.
        compile_command+=" @SYMFIL@"
        finalize_command+=" @SYMFIL@"
        preload=yes
    fi
    case $arg in
*.la | *.lo) ;; # We handle these cases below.
force)
        if test "$dlsself" = no; then

```



```

        dlself=needless
        export_dynamic=yes
    fi
    prev=
    continue
    ;;
self)
    if test "$prev" = dlprefiles; then
        dlself=yes
    elif test "$prev" = dlfiles && test "$dlopen_self" != yes;
then
        dlself=yes
    else
        dlself=needless
        export_dynamic=yes
    fi
    prev=
    continue
    ;;
*)
    if test "$prev" = dlfiles; then
        dlfiles+=" $arg"
    else
        dlprefiles+=" $arg"
    fi
    prev=
    continue
    ;;
esac
;;
expsyms)
    export_symbols="$arg"
    test -f "$arg" \
        || func_fatal_error "symbol file \`$arg' does not exist"
    prev=
    continue
    ;;
expsyms_regex)
    export_symbols_regex="$arg"
    prev=
    continue
    ;;
framework)
    case $host in
        *-*-darwin*)
            case "$deplibs " in
                *" $qarg.ltframework ") ;;
                *) deplibs+=" $qarg.ltframework" # this is fixed later
                    ;;
            esac
        ;;
    esac
    ;;
esac

```

```

    prev=
    continue
    ;;
inst_prefix)
    inst_prefix_dir="$arg"
    prev=
    continue
    ;;
objectlist)
    if test -f "$arg"; then
        save_arg=$arg
        moreargs=
        for fil in `cat "$save_arg"`
        do
#           moreargs+=" $fil"
            arg=$fil
            # A libtool-controlled object.

            # Check to see that this really is a libtool object.
            if func_lalib_unsafe_p "$arg"; then
                pic_object=
                non_pic_object=

                # Read the .lo file
                func_source "$arg"

                if test -z "$pic_object" ||
                    test -z "$non_pic_object" ||
                    test "$pic_object" = none &&
                    test "$non_pic_object" = none; then
                    func_fatal_error "cannot find name of object for `'$arg'"
                fi

                # Extract subdirectory from the argument.
                func_dirname "$arg" "/" ""
                xdir="$func_dirname_result"

                if test "$pic_object" != none; then
                    # Prepend the subdirectory the object is found in.
                    pic_object="$xdir$pic_object"

                    if test "$prev" = dlfiles; then
                        if test "$build_libtool_libs" = yes && test
"$dlopen_support" = yes; then
                            dlfiles+=" $pic_object"
                            prev=
                            continue
                        else
                            # If libtool objects are unsupported, then we need to
preload.
                                prev=dlprefiles
                            fi
                        fi
                    fi
                fi
            fi
        done
    fi

```

```

fi

# CHECK ME: I think I busted this. -Ossama
if test "$prev" = dlprefiles; then
    # Preload the old-style object.
    dlprefiles+=" $pic_object"
    prev=
fi

# A PIC object.
libobjs+=" $pic_object"
arg="$pic_object"
fi

# Non-PIC object.
if test "$non_pic_object" != none; then
    # Prepend the subdirectory the object is found in.
    non_pic_object="$xdir$non_pic_object"

    # A standard non-PIC object
    non_pic_objects+=" $non_pic_object"
    if test -z "$pic_object" || test "$pic_object" = none ;
then
        arg="$non_pic_object"
    fi
else
    # If the PIC object exists, use it instead.
    # $xdir was prepended to $pic_object above.
    non_pic_object="$pic_object"
    non_pic_objects+=" $non_pic_object"
fi
else
# Only an error if not doing a dry-run.
if $opt_dry_run; then
    # Extract subdirectory from the argument.
    func_dirname "$arg" "/" ""
    xdir="$func_dirname_result"

    func_lo2o "$arg"
    pic_object=$xdir$objdir/$func_lo2o_result
    non_pic_object=$xdir$func_lo2o_result
    libobjs+=" $pic_object"
    non_pic_objects+=" $non_pic_object"
    else
        func_fatal_error "\`$arg' is not a valid libtool object"
    fi
fi
done
else
    func_fatal_error "link input file \`$arg' does not exist"
fi
arg=$save_arg

```

```

    prev=
    continue
    ;;
precious_regex)
    precious_files_regex="$arg"
    prev=
    continue
    ;;
release)
    release="- $arg"
    prev=
    continue
    ;;
rpath | xrpath)
    # We need an absolute path.
    case $arg in
    [\\/* | [A-Za-z]:[\\/*]*) ;;
    *)
        func_fatal_error "only absolute run-paths are allowed"
        ;;
    esac
    if test "$prev" = rpath; then
        case "$rpath " in
        * " $arg "*) ;;
        *) rpath+=" $arg" ;;
        esac
    else
        case "$xrpath " in
        * " $arg "*) ;;
        *) xrpath+=" $arg" ;;
        esac
    fi
    prev=
    continue
    ;;
shrext)
    shrext_cmds="$arg"
    prev=
    continue
    ;;
weak)
    weak_libs+=" $arg"
    prev=
    continue
    ;;
xcclinker)
    linker_flags+=" $qarg"
    compiler_flags+=" $qarg"
    prev=
    compile_command+=" $qarg"
    finalize_command+=" $qarg"
    continue

```

```

    ;;
xcompiler)
    compiler_flags+=" $qarg"
    prev=
    compile_command+=" $qarg"
    finalize_command+=" $qarg"
    continue
    ;;
xlinker)
    linker_flags+=" $qarg"
    compiler_flags+=" $wl$qarg"
    prev=
    compile_command+=" $wl$qarg"
    finalize_command+=" $wl$qarg"
    continue
    ;;
*)
    eval "$prev=\"\$arg\""
    prev=
    continue
    ;;
esac
fi # test -n "$prev"

prevarg="$arg"

case $arg in
-all-static)
if test -n "$link_static_flag"; then
    # See comment for -static flag below, for more details.
    compile_command+=" $link_static_flag"
    finalize_command+=" $link_static_flag"
fi
continue
;;

-allow-undefined)
# FIXME: remove this flag sometime in the future.
func_fatal_error "\`-allow-undefined' must not be used because it
is the default"
;;

-avoid-version)
avoid_version=yes
continue
;;

-bindir)
prev=bindir
continue
;;

```

```

    -dlopen)
prev=dlfiles
continue
;;

    -dlpreopen)
prev=dlprefiles
continue
;;

    -export-dynamic)
export_dynamic=yes
continue
;;

    -export-symbols | -export-symbols-regex)
if test -n "$export_symbols" || test -n "$export_symbols_regex";
then
    func_fatal_error "more than one -exported-symbols argument is
not allowed"
fi
if test "X$arg" = "X-export-symbols"; then
    prev=expsyms
else
    prev=expsyms_regex
fi
continue
;;

    -framework)
prev=framework
continue
;;

    -inst-prefix-dir)
prev=inst_prefix
continue
;;

    # The native IRIX linker understands -LANG:*, -LIST:* and -LNO:*
    # so, if we see these flags be careful not to treat them like -L
    -L[A-Z][A-Z]*:*)
case $with_gcc/$host in
no/*-*-irix* | /*-*-irix*)
    compile_command+=" $arg"
    finalize_command+=" $arg"
    ;;
esac
continue
;;

    -L*)

```

```

func_stripname "-L" '' "$arg"
if test -z "$func_stripname_result"; then
  if test "$#" -gt 0; then
    func_fatal_error "require no space between \`-L' and \`$1'"
  else
    func_fatal_error "need path for \`-L' option"
  fi
fi
func_resolve_sysroot "$func_stripname_result"
dir=$func_resolve_sysroot_result
# We need an absolute path.
case $dir in
  [\\/* | [A-Za-z]:[\\/*]*) ;;
*)
  absdir=`cd "$dir" && pwd`
  test -z "$absdir" && \
    func_fatal_error "cannot determine absolute directory name of
\`$dir'"
  dir="$absdir"
  ;;
esac
case "$deplibs " in
*" -L$dir "*" | "*" $arg "*)
  # Will only happen for absolute or sysroot arguments
  ;;
*)
  # Preserve sysroot, but never include relative directories
  case $dir in
    [\\/* | [A-Za-z]:[\\/*] | =*) deplibs+=" $arg" ;;
    *) deplibs+=" -L$dir" ;;
  esac
  lib_search_path+=" $dir"
  ;;
esac
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-cegcc*)
  testbindir=`$ECHO "$dir" | $SED 's*/lib$*/bin*'`
  case :$dllsearchpath: in
    *:$dir:*) ;;
    :*) dllsearchpath=$dir;;
    *) dllsearchpath+=":$dir";;
  esac
  case :$dllsearchpath: in
    *:$testbindir:*) ;;
    :*) dllsearchpath=$testbindir;;
    *) dllsearchpath+=":$testbindir";;
  esac
  ;;
;;
esac
continue
;;

```

```

-l*)
if test "X$arg" = "X-lc" || test "X$arg" = "X-lm"; then
  case $host in
    *--cygwin* | *--mingw* | *--pw32* | *--beos* | *--cegcc* |
*--haiku*)
      # These systems don't actually have a C or math library (as
such)
      continue
      ;;
    *--os2*)
      # These systems don't actually have a C library (as such)
      test "X$arg" = "X-lc" && continue
      ;;
    *--openbsd* | *--freebsd* | *--dragonfly*)
      # Do not include libc due to us having libc/libc_r.
      test "X$arg" = "X-lc" && continue
      ;;
    *--rhapsody* | *--darwin1.[012])
      # Rhapsody C and math libraries are in the System framework
      deplibs+=" System.ltframework"
      continue
      ;;
    *--sco3.2v5* | *--sco5v6*)
      # Causes problems with __ctype
      test "X$arg" = "X-lc" && continue
      ;;
    *--sysv4.2uw2* | *--sysv5* | *--unixware* | *--OpenUNIX*)
      # Compiler inserts libc in the correct place for threads to
work
      test "X$arg" = "X-lc" && continue
      ;;
  esac
elif test "X$arg" = "X-lc_r"; then
  case $host in
    *--openbsd* | *--freebsd* | *--dragonfly*)
      # Do not include libc_r directly, use -pthread flag.
      continue
      ;;
  esac
fi
deplibs+=" $arg"
continue
;;

-module)
module=yes
continue
;;

# Tru64 UNIX uses -model [arg] to determine the layout of C++
# classes, name mangling, and exception handling.
# Darwin uses the -arch flag to determine output architecture.

```



```

    -model|-arch|-isysroot|--sysroot)
compiler_flags+=" $arg"
compile_command+=" $arg"
finalize_command+=" $arg"
prev=xcompiler
continue
;;

    -mt|-mthreads|-kthread|-Kthread|-pthread|-pthreads|--thread-safe
\
|-threads|-fopenmp|-openmp|-mp|-xopenmp|-omp|-qsmp=*)
compiler_flags+=" $arg"
compile_command+=" $arg"
finalize_command+=" $arg"
case "$new_inherited_linker_flags " in
    *" $arg ") ;;
    * ) new_inherited_linker_flags+=" $arg" ;;
esac
continue
;;

    -multi_module)
single_module="{wl}-multi_module"
continue
;;

    -no-fast-install)
fast_install=no
continue
;;

    -no-install)
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-darwin* |
*-cegcc*)
    # The PATH hackery in wrapper scripts is required on Windows
    # and Darwin in order for the loader to find any dlls it needs.
    func_warning "\`-no-install' is ignored for $host"
    func_warning "assuming \`-no-fast-install' instead"
    fast_install=no
    ;;
*) no_install=yes ;;
esac
continue
;;

    -no-undefined)
allow_undefined=no
continue
;;

    -objectlist)

```

```

prev=objectlist
continue
;;

-o) prev=output ;;

-precious-files-regex)
prev=precious_regex
continue
;;

-release)
prev=release
continue
;;

-rpath)
prev=rpath
continue
;;

-R)
prev=xrpath
continue
;;

-R*)
func_stripname '-R' '' "$arg"
dir=${func_stripname_result}
# We need an absolute path.
case $dir in
[\\/] * | [A-Za-z]:[\\/] *) ;;
=*)
    func_stripname '=' '' "$dir"
    dir=${lt_sysroot}${func_stripname_result}
    ;;
*)
    func_fatal_error "only absolute run-paths are allowed"
    ;;
esac
case "$xrpath " in
*" $dir "*) ;;
*) xrpath+=" $dir" ;;
esac
continue
;;

-shared)
# The effects of -shared are defined in a previous loop.
continue
;;

```

```

-shrext)
prev=shrext
continue
;;

-static | -static-libtool-libs)
# The effects of -static are defined in a previous loop.
# We used to do the same as -all-static on platforms that
# didn't have a PIC flag, but the assumption that the effects
# would be equivalent was wrong. It would break on at least
# Digital Unix and AIX.
continue
;;

-thread-safe)
thread_safe=yes
continue
;;

-version-info)
prev=vinfo
continue
;;

-version-number)
prev=vinfo
vinfo_number=yes
continue
;;

-weak)
prev=weak
continue
;;

-Wc,*)
func_stripname '-Wc,' '' "$arg"
args=$func_stripname_result
arg=
save_ifs="$IFS"; IFS=', '
for flag in $args; do
IFS="$save_ifs"
func_quote_for_eval "$flag"
arg+=" $func_quote_for_eval_result"
compiler_flags+=" $func_quote_for_eval_result"
done
IFS="$save_ifs"
func_stripname ' ' '' "$arg"
arg=$func_stripname_result
;;

-Wl,*)

```

```

func_stripname '-Wl,' '' "$arg"
args=$func_stripname_result
arg=
save_ifs="$IFS"; IFS=', '
for flag in $args; do
    IFS="$save_ifs"
    func_quote_for_eval "$flag"
    arg+=" $wl$func_quote_for_eval_result"
    compiler_flags+=" $wl$func_quote_for_eval_result"
    linker_flags+=" $func_quote_for_eval_result"
done
IFS="$save_ifs"
func_stripname ' ' '' "$arg"
arg=$func_stripname_result
;;

-Xcompiler)
prev=xcompiler
continue
;;

-Xlinker)
prev=xlinker
continue
;;

-XCCLinker)
prev=xcclinker
continue
;;

# -msg_* for osf cc
-msg_*)
func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
;;

# Flags to be passed through unchanged, with rationale:
# -64, -mips[0-9]      enable 64-bit mode for the SGI compiler
# -r[0-9][0-9]*       specify processor for the SGI compiler
# -xarch=*, -xtarget=* enable 64-bit mode for the Sun compiler
# +DA*, +DD*          enable 64-bit mode for the HP compiler
# -q*                 compiler args for the IBM compiler
# -m*, -t[45]*, -txscale* architecture-specific flags for GCC
# -F/path             path to uninstalled frameworks, gcc on
darwin
# -p, -pg, --coverage, -fprofile-* profiling flags for GCC
# -fstack-protector*  stack protector flags for GCC
# @file              GCC response files
# -tp=*              Portland pgcc target processor selection
# --sysroot=*        for sysroot support

```

```

        # -O*, -flto*, -fwhopr*, -fuse-linker-plugin GCC link-time
optimization
        -64|-mips[0-9]|-r[0-9][0-9]*|-xarch=*|-xtarget=*|+DA*|+DD*|-q*|-
m*| \
        -t[45]*|-txscale*|-p|-pg|--coverage|-fprofile-*|-F*|@*|-tp=*|--
sysroot=*| \
        -O*|-flto*|-fwhopr*|-fuse-linker-plugin|-fstack-protector*)
        func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
        compile_command+=" $arg"
        finalize_command+=" $arg"
        compiler_flags+=" $arg"
        continue
        ;;

# Some other compiler flag.
-* | +*)
        func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
        ;;

*.$objext)
# A standard object.
objs+=" $arg"
        ;;

*.lo)
# A libtool-controlled object.

# Check to see that this really is a libtool object.
if func_lalib_unsafe_p "$arg"; then
        pic_object=
        non_pic_object=

# Read the .lo file
func_source "$arg"

        if test -z "$pic_object" ||
            test -z "$non_pic_object" ||
            test "$pic_object" = none &&
            test "$non_pic_object" = none; then
                func_fatal_error "cannot find name of object for `\$arg'"
        fi

# Extract subdirectory from the argument.
func_dirname "$arg" "/" ""
xdir="$func_dirname_result"

        if test "$pic_object" != none; then
            # Prepend the subdirectory the object is found in.
            pic_object="$xdir$pic_object"

```

```

        if test "$prev" = dlfiles; then
            if test "$build_libtool_libs" = yes && test
"$dlopen_support" = yes; then
                dlfiles+=" $pic_object"
                prev=
                continue
            else
                # If libtool objects are unsupported, then we need to
preload.
                prev=dlprefiles
            fi
        fi

        # CHECK ME: I think I busted this. -Ossama
        if test "$prev" = dlprefiles; then
            # Preload the old-style object.
            dlprefiles+=" $pic_object"
            prev=
        fi

        # A PIC object.
        libobjs+=" $pic_object"
        arg="$pic_object"
    fi

    # Non-PIC object.
    if test "$non_pic_object" != none; then
        # Prepend the subdirectory the object is found in.
        non_pic_object="$xdir$non_pic_object"

        # A standard non-PIC object
        non_pic_objects+=" $non_pic_object"
        if test -z "$pic_object" || test "$pic_object" = none ; then
            arg="$non_pic_object"
        fi
    else
        # If the PIC object exists, use it instead.
        # $xdir was prepended to $pic_object above.
        non_pic_object="$pic_object"
        non_pic_objects+=" $non_pic_object"
    fi
else
    # Only an error if not doing a dry-run.
    if $opt_dry_run; then
        # Extract subdirectory from the argument.
        func_dirname "$arg" "/" ""
        xdir="$func_dirname_result"

        func_lo2o "$arg"
        pic_object=$xdir$objdir/$func_lo2o_result
        non_pic_object=$xdir$func_lo2o_result
        libobjs+=" $pic_object"
    fi
fi

```

```

        non_pic_objects+=" $non_pic_object"
    else
        func_fatal_error "\`$arg' is not a valid libtool object"
    fi
fi
;;

*.$libext)
# An archive.
deplibs+=" $arg"
old_deplibs+=" $arg"
continue
;;

*.la)
# A libtool-controlled library.

func_resolve_sysroot "$arg"
if test "$prev" = dlfiles; then
    # This library was specified with -dlopen.
    dlfiles+=" $func_resolve_sysroot_result"
    prev=
elif test "$prev" = dlprefiles; then
    # The library was specified with -dlpreopen.
    dlprefiles+=" $func_resolve_sysroot_result"
    prev=
else
    deplibs+=" $func_resolve_sysroot_result"
fi
continue
;;

# Some other compiler argument.
*)
# Unknown arguments in both finalize_command and compile_command
need
# to be aesthetically quoted because they are eval'd later.
func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
;;
esac # arg

# Now actually substitute the argument into the commands.
if test -n "$arg"; then
    compile_command+=" $arg"
    finalize_command+=" $arg"
fi
done # argument parsing loop

test -n "$prev" && \
    func_fatal_help "the \`$prevarg' option requires an argument"

```

```

    if test "$export_dynamic" = yes && test -n
"$export_dynamic_flag_spec"; then
        eval arg="\$export_dynamic_flag_spec\"
        compile_command+=" $arg"
        finalize_command+=" $arg"
    fi

oldlibs=
# calculate the name of the file, without its directory
func_basename "$output"
outputname="$func_basename_result"
libobjs_save="$libobjs"

if test -n "$shlibpath_var"; then
    # get the directories listed in $shlibpath_var
    eval shlib_search_path=\`$ECHO "\${$shlibpath_var}" \| \${SED
\'s/:/ /g\|\'`
else
    shlib_search_path=
fi
eval sys_lib_search_path="\$sys_lib_search_path_spec\"
eval sys_lib_dlsearch_path="\$sys_lib_dlsearch_path_spec\"

func_dirname "$output" "/" ""
output_objdir="$func_dirname_result$objdir"
func_to_tool_file "$output_objdir/"
tool_output_objdir=$func_to_tool_file_result
# Create the object directory.
func_mkdir_p "$output_objdir"

# Determine the type of output
case $output in
"")
    func_fatal_help "you must specify an output file"
    ;;
*.$libext) linkmode=oldlib ;;
*.lo | *.$objext) linkmode=obj ;;
*.la) linkmode=lib ;;
*) linkmode=prog ;; # Anything else should be a program.
esac

specialdeplibs=

libs=
# Find all interdependent deplibs by searching for libraries
# that are linked more than once (e.g. -la -lb -la)
for deplib in $deplibs; do
    if $opt_preserve_dup_deps ; then
        case "$libs " in
*" $deplib ") specialdeplibs+=" $deplib" ;;
        esac
    fi
fi

```



```

    libs+=" $deplib"
done

if test "$linkmode" = lib; then
    libs="$predeps $libs $compiler_lib_search_path $postdeps"

    # Compute libraries that are listed more than once in $predeps
    # $postdeps and mark them as special (i.e., whose duplicates are
    # not to be eliminated).
    pre_post_deps=
    if $opt_duplicate_compiler_generated_deps; then
    for pre_post_dep in $predeps $postdeps; do
        case "$pre_post_deps " in
            *) $pre_post_dep *) specialdeplibs+=" $pre_post_deps" ;;
        esac
        pre_post_deps+=" $pre_post_dep"
    done
    fi
    pre_post_deps=
fi

deplibs=
newdependency_libs=
newlib_search_path=
need_relink=no # whether we're linking any uninstalled libtool
libraries
notinst_deplibs= # not-installed libtool libraries
notinst_path= # paths that contain not-installed libtool libraries

case $linkmode in
lib)
    passes="conv dlpreopen link"
    for file in $dlfiles $dlprefiles; do
        case $file in
            *.la) ;;
            *)
                func_fatal_help "libraries can \`-dlopen' only libtool
libraries: $file"
                ;;
        esac
    done
    ;;
prog)
    compile_deplibs=
    finalize_deplibs=
    alldeplibs=no
    newdlfiles=
    newdlprefiles=
    passes="conv scan dlopen dlpreopen link"
    ;;
*) passes="conv"
    ;;

```

```

esac

for pass in $passes; do
# The preopen pass in lib mode reverses $deplibs; put it back
here
# so that -L comes before libs that need it for instance...
if test "$linkmode,$pass" = "lib,link"; then
## FIXME: Find the place where the list is rebuilt in the wrong
##          order, and fix it there properly
tmp_deplibs=
for deplib in $deplibs; do
tmp_deplibs="$deplib $tmp_deplibs"
done
deplibs="$tmp_deplibs"
fi

if test "$linkmode,$pass" = "lib,link" ||
test "$linkmode,$pass" = "prog,scan"; then
libs="$deplibs"
deplibs=
fi
if test "$linkmode" = prog; then
case $pass in
dlopen) libs="$dlfiles" ;;
dlpreopen) libs="$dlprefiles" ;;
link) libs="$deplibs %DEPLIBS% $dependency_libs" ;;
esac
fi
if test "$linkmode,$pass" = "lib,dlpreopen"; then
# Collect and forward deplibs of preopened libtool libs
for lib in $dlprefiles; do
# Ignore non-libtool-libs
dependency_libs=
func_resolve_sysroot "$lib"
case $lib in
*.la) func_source "$func_resolve_sysroot_result" ;;
esac

# Collect preopened libtool deplibs, except any this library
# has declared as weak libs
for deplib in $dependency_libs; do
func_basename "$deplib"
deplib_base=$func_basename_result
case " $weak_libs " in
*" $deplib_base ") ;;
*) deplibs+=" $deplib" ;;
esac
done
done
libs="$dlprefiles"
fi
if test "$pass" = dlopen; then

```

```

# Collect dlpreopened libraries
save_deplibs="$deplibs"
deplibs=
fi

for deplib in $libs; do
lib=
found=no
case $deplib in
-mt|-mthreads|-kthread|-Kthread|-pthread|-pthreads|--thread-safe
\
|-threads|-fopenmp|-openmp|-mp|-xopenmp|-omp|-qsmplib*)
if test "$linkmode,$pass" = "prog,link"; then
compile_deplibs="$deplib $compile_deplibs"
finalize_deplibs="$deplib $finalize_deplibs"
else
compiler_flags+="$ deplib"
if test "$linkmode" = lib ; then
case "$new_inherited_linker_flags " in
*" $deplib ") ;;
* ) new_inherited_linker_flags+="$ deplib" ;;
esac
fi
fi
continue
;;
-l*)
if test "$linkmode" != lib && test "$linkmode" != prog; then
func_warning "`-l' is ignored for archives/objects"
continue
fi
func_stripname '-l' '' "$deplib"
name=$func_stripname_result
if test "$linkmode" = lib; then
searchdirs="$newlib_search_path $lib_search_path
$compiler_lib_search_dirs $sys_lib_search_path $shlib_search_path"
else
searchdirs="$newlib_search_path $lib_search_path
$sys_lib_search_path $shlib_search_path"
fi
for searchdir in $searchdirs; do
for search_ext in .la $std_shrext .so .a; do
# Search the libtool library
lib="$searchdir/lib${name}${search_ext}"
if test -f "$lib"; then
if test "$search_ext" = ".la"; then
found=yes
else
found=no
fi
break 2
fi
fi

```

```

done
done
if test "$found" != yes; then
  # deplib doesn't seem to be a libtool library
  if test "$linkmode,$pass" = "prog,link"; then
    compile_deplibs="$deplib $compile_deplibs"
    finalize_deplibs="$deplib $finalize_deplibs"
  else
    deplibs="$deplib $deplibs"
    test "$linkmode" = lib && newdependency_libs="$deplib
$newdependency_libs"
  fi
  continue
else # deplib is a libtool library
  # If $allow_libtool_libs_with_static_runtimes && $deplib is a
stdlib,
  # We need to do some special things here, and not later.
  if test "X$allow_libtool_libs_with_static_runtimes" = "Xyes"
; then
    case " $predeps $postdeps " in
    *" $deplib ")
      if func_lalib_p "$lib"; then
        library_names=
        old_library=
        func_source "$lib"
        for l in $old_library $library_names; do
          ll="$l"
        done
        if test "X$ll" = "X$old_library" ; then # only static
version available
          found=no
          func_dirname "$lib" "" "."
          ladir="$func_dirname_result"
          lib=$ladir/$old_library
          if test "$linkmode,$pass" = "prog,link"; then
            compile_deplibs="$deplib $compile_deplibs"
            finalize_deplibs="$deplib $finalize_deplibs"
          else
            deplibs="$deplib $deplibs"
            test "$linkmode" = lib && newdependency_libs="$deplib
$newdependency_libs"
          fi
          continue
        fi
      fi
    ;;
    *) ;;
  esac
fi
fi
;; # -l
*.ltframework)

```

```

if test "$linkmode,$pass" = "prog,link"; then
    compile_deplibs="$deplib $compile_deplibs"
    finalize_deplibs="$deplib $finalize_deplibs"
else
    deplibs="$deplib $deplibs"
    if test "$linkmode" = lib ; then
        case "$new_inherited_linker_flags " in
            *) $deplib *) ;;
            * ) new_inherited_linker_flags+="$ deplib" ;;
        esac
    fi
fi
continue
;;
-L*)
case $linkmode in
lib)
    deplibs="$deplib $deplibs"
    test "$pass" = conv && continue
    newdependency_libs="$deplib $newdependency_libs"
    func_stripname '-L' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    newlib_search_path+="$ func_resolve_sysroot_result"
    ;;
prog)
    if test "$pass" = conv; then
        deplibs="$deplib $deplibs"
        continue
    fi
    if test "$pass" = scan; then
        deplibs="$deplib $deplibs"
    else
        compile_deplibs="$deplib $compile_deplibs"
        finalize_deplibs="$deplib $finalize_deplibs"
    fi
    func_stripname '-L' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    newlib_search_path+="$ func_resolve_sysroot_result"
    ;;
*)
    func_warning "\`-L' is ignored for archives/objects"
    ;;
esac # linkmode
continue
;; # -L
-R*)
if test "$pass" = link; then
    func_stripname '-R' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    dir=$func_resolve_sysroot_result
    # Make sure the xrpath contains only unique directories.
    case "$xrpath " in

```

```

        *) $dir *) ;;
        *) xrp+= " $dir" ;;
    esac
fi
deplibs="$deplib $deplibs"
continue
;;
*.la)
    func_resolve_sysroot "$deplib"
    lib=$func_resolve_sysroot_result
    ;;
*.$libext)
    if test "$pass" = conv; then
        deplibs="$deplib $deplibs"
        continue
    fi
    case $linkmode in
    lib)
        # Linking convenience modules into shared libraries is
allowed,
        # but linking other static libraries is non-portable.
        case " $dlpreconveniencelibs " in
        *) $deplib *) ;;
        *)
            valid_a_lib=no
            case $deplibs_check_method in
            match_pattern*)
                set dummy $deplibs_check_method; shift
                match_pattern_regex=`expr "$deplibs_check_method" : "$1
\(.*\)"`
                if eval "\$ECHO \"\$deplib\" 2>/dev/null | $SED 10q \
                    | $EGREP \"$match_pattern_regex\" > /dev/null; then
                    valid_a_lib=yes
                fi
                ;;
            pass_all)
                valid_a_lib=yes
                ;;
            esac
            if test "$valid_a_lib" != yes; then
                echo
                $ECHO "*** Warning: Trying to link with static lib archive
$deplib."
                echo "*** I have the capability to make that library
automatically link in when"
                echo "*** you link to this library. But I can only do this
if you have a"
                echo "*** shared version of the library, which you do not
appear to have"
                echo "*** because the file extensions .$libext of this
argument makes me believe"

```

```

        echo "*** that it is just a static archive that I should
not use here."
        else
        echo
        $ECHO "*** Warning: Linking the shared library $output
against the"
        $ECHO "*** static library $deplib is not portable!"
        deplibs="$deplib $deplibs"
        fi
        ;;
    esac
    continue
    ;;
prog)
    if test "$pass" != link; then
        deplibs="$deplib $deplibs"
    else
        compile_deplibs="$deplib $compile_deplibs"
        finalize_deplibs="$deplib $finalize_deplibs"
    fi
    continue
    ;;
esac # linkmode
;; # *.$libext
*.lo | *.$objext)
    if test "$pass" = conv; then
        deplibs="$deplib $deplibs"
    elif test "$linkmode" = prog; then
        if test "$pass" = dlpreopen || test "$dlopen_support" != yes
|| test "$build_libtool_libs" = no; then
            # If there is no dlopen support or we're linking
statically,
            # we need to preload.
            newdlprefiles+=" $deplib"
            compile_deplibs="$deplib $compile_deplibs"
            finalize_deplibs="$deplib $finalize_deplibs"
        else
            newdlfiles+=" $deplib"
        fi
    fi
    continue
    ;;
%DEPLIBS%)
    alldeplibs=yes
    continue
    ;;
esac # case $deplib

if test "$found" = yes || test -f "$lib"; then :
else
    func_fatal_error "cannot find the library `$lib' or unhandled
argument ``$deplib'"

```

```

fi

# Check to see that this really is a libtool archive.
func_lalib_unsafe_p "$lib" \
  || func_fatal_error "\`$lib' is not a valid libtool archive"

func_dirname "$lib" "" "."
ladir="$func_dirname_result"

dlname=
dlopen=
dlpreopen=
libdir=
library_names=
old_library=
inherited_linker_flags=
# If the library was installed with an old release of libtool,
# it will not redefine variables installed, or shouldnotlink
installed=yes
shouldnotlink=no
avoidtempdirpath=

# Read the .la file
func_source "$lib"

# Convert "-framework foo" to "foo.lframework"
if test -n "$inherited_linker_flags"; then
  tmp_inherited_linker_flags=`$ECHO "$inherited_linker_flags" |
$SED 's/-framework \([^ $]*\)\/\1.lframework/g'\`
  for tmp_inherited_linker_flag in $tmp_inherited_linker_flags;
do
  case " $new_inherited_linker_flags " in
    * $tmp_inherited_linker_flag *) ;;
    *) new_inherited_linker_flags+="$
$tmp_inherited_linker_flag";;
  esac
done
fi
dependency_libs=`$ECHO " $dependency_libs" | $SED 's% \([^
$]*\)\.lframework% -framework \1%g'\`
if test "$linkmode,$pass" = "lib,link" ||
  test "$linkmode,$pass" = "prog,scan" ||
  { test "$linkmode" != prog && test "$linkmode" != lib; }; then
  test -n "$dlopen" && dlfiles+="$dlopen"
  test -n "$dlpreopen" && dlprefiles+="$dlpreopen"
fi

if test "$pass" = conv; then
  # Only check for convenience libraries
  deplibs="$lib $deplibs"
  if test -z "$libdir"; then

```



```

        if test -z "$old_library"; then
            func_fatal_error "cannot find name of link library for
\`$lib'"
        fi
        # It is a libtool convenience library, so add in its objects.
        convenience+=" $ladir/$objdir/$old_library"
        old_convenience+=" $ladir/$objdir/$old_library"
    elif test "$linkmode" != prog && test "$linkmode" != lib; then
        func_fatal_error "\`$lib' is not a convenience library"
    fi
    tmp_libs=
    for deplib in $dependency_libs; do
        deplibs="$deplib $deplibs"
        if $opt_preserve_dup_deps ; then
            case "$tmp_libs " in
                *" $deplib "*) specialdeplibs+=" $deplib" ;;
            esac
        fi
        tmp_libs+=" $deplib"
    done
    continue
fi # $pass = conv

# Get the name of the library we link against.
linklib=
if test -n "$old_library" &&
{ test "$prefer_static_libs" = yes ||
  test "$prefer_static_libs,$installed" = "built,no"; }; then
    linklib=$old_library
else
    for l in $old_library $library_names; do
        linklib="$l"
    done
fi
if test -z "$linklib"; then
    func_fatal_error "cannot find name of link library for \`$lib'"
fi

# This library was specified with -dlopen.
if test "$pass" = dlopen; then
    if test -z "$libdir"; then
        func_fatal_error "cannot -dlopen a convenience library:
\`$lib'"
    fi
    if test -z "$dlname" ||
    test "$dlopen_support" != yes ||
    test "$build_libtool_libs" = no; then
        # If there is no dlname, no dlopen support or we're linking
        # statically, we need to preload.  We also need to preload
any
        # dependent libraries so libltdl's deplib preloader doesn't

```

```

        # bomb out in the load deplibs phase.
        dlprefiles+=" $lib $dependency_libs"
    else
        newdlfiles+=" $lib"
    fi
    continue
fi # $pass = dlopen

# We need an absolute path.
case $ladir in
[\\/* | [A-Za-z]:[\\/*]*) abs_ladir="$ladir" ;;
*)
    abs_ladir=`cd "$ladir" && pwd`
    if test -z "$abs_ladir"; then
        func_warning "cannot determine absolute directory name of
`$ladir'"
        func_warning "passing it literally to the linker, although it
might fail"
        abs_ladir="$ladir"
    fi
    ;;
esac
func_basename "$lib"
laname="$func_basename_result"

# Find the relevant object directory and library name.
if test "X$installed" = Xyes; then
    if test ! -f "$lt_sysroot$libdir/$linklib" && test -f
"$abs_ladir/$linklib"; then
        func_warning "library `$lib' was moved."
        dir="$ladir"
        absdir="$abs_ladir"
        libdir="$abs_ladir"
    else
        dir="$lt_sysroot$libdir"
        absdir="$lt_sysroot$libdir"
    fi
    test "X$hardcode_automatic" = Xyes && avoidtemprpath=yes
else
    if test ! -f "$ladir/$objdir/$linklib" && test -f
"$abs_ladir/$linklib"; then
        dir="$ladir"
        absdir="$abs_ladir"
        # Remove this search path later
        notinst_path+=" $abs_ladir"
    else
        dir="$ladir/$objdir"
        absdir="$abs_ladir/$objdir"
        # Remove this search path later
        notinst_path+=" $abs_ladir"
    fi
fi # $installed = yes

```

```

func_stripname 'lib' '.la' "$laname"
name=$func_stripname_result

# This library was specified with -dlpreopen.
if test "$pass" = dlpreopen; then
  if test -z "$libdir" && test "$linkmode" = prog; then
    func_fatal_error "only libraries may -dlpreopen a convenience
library: `\$lib'"
  fi
  case "$host" in
    # special handling for platforms with PE-DLLs.
    *cygwin* | *mingw* | *cegcc* )
      # Linker will automatically link against shared library if
both
      # static and shared are present.  Therefore, ensure we
extract
      # symbols from the import library if a shared library is
present
      # (otherwise, the dlopen module name will be incorrect).
We do
      # this by putting the import library name into
$newdlprefiles.
      # We recover the dlopen module name by 'saving' the la file
      # name in a special purpose variable, and (later)
extracting the
      # dlname from the la file.
      if test -n "$dlname"; then
        func_tr_sh "$dir/$linklib"
        eval "libfile_$func_tr_sh_result=\$abs_ladir/\$laname"
        newdlprefiles+=" $dir/$linklib"
      else
        newdlprefiles+=" $dir/$old_library"
        # Keep a list of preopened convenience libraries to check
        # that they are being used correctly in the link pass.
        test -z "$libdir" && \
          dlpreconveniencelibs+=" $dir/$old_library"
      fi
    ;;
    * )
      # Prefer using a static library (so that no silly _DYNAMIC
symbols
      # are required to link).
      if test -n "$old_library"; then
        newdlprefiles+=" $dir/$old_library"
        # Keep a list of preopened convenience libraries to check
        # that they are being used correctly in the link pass.
        test -z "$libdir" && \
          dlpreconveniencelibs+=" $dir/$old_library"
        # Otherwise, use the dlname, so that lt_dlopen finds it.
      elif test -n "$dlname"; then
        newdlprefiles+=" $dir/$dlname"
      else

```

```

        newdlprefiles+=" $dir/$linklib"
    fi
    ;;
esac
fi # $pass = dlpreopen

if test -z "$libdir"; then
    # Link the convenience library
    if test "$linkmode" = lib; then
        deplibs="$dir/$old_library $deplibs"
    elif test "$linkmode,$pass" = "prog,link"; then
        compile_deplibs="$dir/$old_library $compile_deplibs"
        finalize_deplibs="$dir/$old_library $finalize_deplibs"
    else
        deplibs="$lib $deplibs" # used for prog,scan pass
    fi
    continue
fi

if test "$linkmode" = prog && test "$pass" != link; then
    newlib_search_path+=" $ldir"
    deplibs="$lib $deplibs"

    linkalldeplibs=no
    if test "$link_all_deplibs" != no || test -z "$library_names"
||
        test "$build_libtool_libs" = no; then
        linkalldeplibs=yes
    fi

    tmp_libs=
    for deplib in $dependency_libs; do
        case $deplib in
            -L*) func_stripname '-L' '' "$deplib"
                func_resolve_sysroot "$func_stripname_result"
                newlib_search_path+=" $func_resolve_sysroot_result"
                ;;
        esac
        # Need to link against all dependency_libs?
        if test "$linkalldeplibs" = yes; then
            deplibs="$deplib $deplibs"
        else
            # Need to hardcode shared library paths
            # or/and link against static libraries
            newdependency_libs="$deplib $newdependency_libs"
        fi
        if $opt_preserve_dup_deps ; then
            case "$tmp_libs " in
                *" $deplib "*) specialdeplibs+=" $deplib" ;;
            esac
        fi
    done
fi

```

```

    tmp_libs+=" $deplib"
done # for deplib
continue
fi # $linkmode = prog...

if test "$linkmode,$pass" = "prog,link"; then
  if test -n "$library_names" &&
    { { test "$prefer_static_libs" = no ||
      test "$prefer_static_libs,$installed" = "built,yes"; }
  ||
    test -z "$old_library"; }; then
    # We need to hardcode the library path
    if test -n "$shlibpath_var" && test -z "$avoidtemprpath" ;
then
    # Make sure the rpath contains only unique directories.
    case "$temp_rpath:" in
    *"$absdir:") ;;
    *) temp_rpath+=" $absdir:" ;;
    esac
    fi

    # Hardcode the library path.
    # Skip directories that are in the system default run-time
    # search path.
    #case " $sys_lib_dlsearch_path " in
    #*" $absdir "*) ;;
    #*)
    # case "$compile_rpath " in
    # *" $absdir "*) ;;
    # *) compile_rpath+=" $absdir" ;;
    # esac
    # ;;
    #esac
    case " $sys_lib_dlsearch_path " in
    *" $libdir "*) ;;
    *)
    case "$finalize_rpath " in
    *" $libdir "*) ;;
    *) finalize_rpath+=" $libdir" ;;
    esac
    ;;
    esac
    fi # $linkmode,$pass = prog,link...

    if test "$alldeplibs" = yes &&
      { test "$deplibs_check_method" = pass_all ||
        { test "$build_libtool_libs" = yes &&
          test -n "$library_names"; }; }; then
        # We only need to search for static libraries
        continue
      fi
    fi
fi

```

```

link_static=no # Whether the deplib will be linked statically
use_static_libs=$prefer_static_libs
if test "$use_static_libs" = built && test "$installed" = yes;
then
    use_static_libs=no
fi
if test -n "$library_names" &&
    { test "$use_static_libs" = no || test -z "$old_library"; };
then
    case $host in
    *cygwin* | *mingw* | *cegcc*)
        # No point in relinking DLLs because paths are not encoded
        notinst_deplibs+=" $lib"
        need_relink=no
        ;;
    *)
        if test "$installed" = no; then
            notinst_deplibs+=" $lib"
            need_relink=yes
        fi
        ;;
    esac
    # This is a shared library

    # Warn about portability, can't link against -module's on some
    # systems (darwin). Don't bleat about dlopened modules though!
    dlopenmodule=""
    for dlpremoduletest in $dlprefiles; do
        if test "X$dlpremoduletest" = "X$lib"; then
            dlopenmodule="$dlpremoduletest"
            break
        fi
    done
    if test -z "$dlopenmodule" && test "$shouldnotlink" = yes &&
test "$pass" = link; then
        echo
        if test "$linkmode" = prog; then
            $ECHO "*** Warning: Linking the executable $output against
the loadable module"
        else
            $ECHO "*** Warning: Linking the shared library $output
against the loadable module"
        fi
        $ECHO "*** $linklib is not portable!"
    fi
    if test "$linkmode" = lib &&
        test "$hardcode_into_libs" = yes; then
        # Hardcode the library path.
        # Skip directories that are in the system default run-time
        # search path.
        #case " $sys_lib_dlsearch_path " in

```

```

#*) $absdir *) ;;
#*)
# case "$compile_rpath " in
# *) $absdir *) ;;
# *) compile_rpath+=" $absdir" ;;
# esac
# ;;
#esac
case " $sys_lib_dlsearch_path " in
*) $libdir *) ;;
*)
  case "$finalize_rpath " in
  *) $libdir *) ;;
  *) finalize_rpath+=" $libdir" ;;
  esac
  ;;
esac
fi

if test -n "$old_archive_from_expsyms_cmds"; then
# figure out the soname
set dummy $library_names
shift
realname="$1"
shift
libname=`eval "\\$ECHO \"$libname_spec\""`
# use dlname if we got it. it's perfectly good, no?
if test -n "$dlname"; then
  soname="$dlname"
elif test -n "$soname_spec"; then
# bleh windows
case $host in
*cygwin* | mingw* | *cegcc*)
  func_arith $current - $age
major=$func_arith_result
versuffix="-${major}"
  ;;
esac
eval soname="\${soname_spec}"
else
  soname="$realname"
fi

# Make a new name for the extract_expsyms_cmds to use
soroot="$soname"
func_basename "$soroot"
soname="$func_basename_result"
func_stripname 'lib' '.dll' "$soname"
newlib=libimp-${func_stripname_result}.a

# If the library has no export list, then create one now
if test -f "$output_objdir/$soname-def"; then :

```

```

else
    func_verbose "extracting exported symbol list from
\`$soname'"
    func_execute_cmds "$extract_expsyms_cmds" 'exit $?'
fi

# Create $newlib
if test -f "$output_objdir/$newlib"; then ;; else
    func_verbose "generating import library for \`$soname'"
    func_execute_cmds "$old_archive_from_expsyms_cmds" 'exit
$?'
fi
# make sure the library variables are pointing to the new
library
dir=$output_objdir
linklib=$newlib
fi # test -n "$old_archive_from_expsyms_cmds"

if test "$linkmode" = prog || test "$opt_mode" != relink; then
add_shlibpath=
add_dir=
add=
lib_linked=yes
case $hardcode_action in
immediate | unsupported)
    if test "$hardcode_direct" = no; then
add="$dir/$linklib"
case $host in
*-*-sco3.2v5.0.[024]*) add_dir="-L$dir" ;;
*-*-sysv4*uw2*) add_dir="-L$dir" ;;
*-*-sysv5OpenUNIX* | *-*-sysv5UnixWare7.[01].[10]* | \
*-*-unixware7*) add_dir="-L$dir" ;;
*-*-darwin* )
    # if the lib is a (non-dlopened) module then we can not
    # link against it, someone is ignoring the earlier
warnings
    if /usr/bin/file -L $add 2> /dev/null |
    $GREP ": [^:]* bundle" >/dev/null ; then
    if test "X$dlopenmodule" != "X$lib"; then
    $ECHO "*** Warning: lib $linklib is a module, not a
shared library"
    if test -z "$old_library" ; then
        echo
        echo "*** And there doesn't seem to be a static
archive available"
        echo "*** The link will probably fail, sorry"
    else
        add="$dir/$old_library"
    fi
    elif test -n "$old_library"; then
        add="$dir/$old_library"
    fi
fi

```



```

        fi
    esac
    elif test "$hardcode_minus_L" = no; then
    case $host in
    *-*-sunos*) add_shlibpath="$dir" ;;
    esac
    add_dir="-L$dir"
    add="-l$name"
    elif test "$hardcode_shlibpath_var" = no; then
    add_shlibpath="$dir"
    add="-l$name"
    else
    lib_linked=no
    fi
    ;;
relink)
    if test "$hardcode_direct" = yes &&
        test "$hardcode_direct_absolute" = no; then
    add="$dir/$linklib"
    elif test "$hardcode_minus_L" = yes; then
    add_dir="-L$absdir"
    # Try looking first in the location we're being installed

```

to.

```

    if test -n "$inst_prefix_dir"; then
        case $libdir in
        [\\/]*)
            add_dir+=" -L$inst_prefix_dir$libdir"
            ;;
        esac
    fi
    add="-l$name"
    elif test "$hardcode_shlibpath_var" = yes; then
    add_shlibpath="$dir"
    add="-l$name"
    else
    lib_linked=no
    fi
    ;;
*) lib_linked=no ;;
esac

if test "$lib_linked" != yes; then
    func_fatal_configuration "unsupported hardcode properties"
fi

if test -n "$add_shlibpath"; then
    case :$compile_shlibpath: in
    *:$add_shlibpath:*) ;;
    *) compile_shlibpath+=" $add_shlibpath:" ;;
    esac
fi
if test "$linkmode" = prog; then

```

```

        test -n "$add_dir" && compile_deplibs="$add_dir
$compile_deplibs"
        test -n "$add" && compile_deplibs="$add $compile_deplibs"
    else
        test -n "$add_dir" && deplibs="$add_dir $deplibs"
        test -n "$add" && deplibs="$add $deplibs"
        if test "$hardcode_direct" != yes &&
            test "$hardcode_minus_L" != yes &&
            test "$hardcode_shlibpath_var" = yes; then
            case :$finalize_shlibpath: in
            *):$libdir:*) ;;
            *) finalize_shlibpath+="$libdir:" ;;
            esac
        fi
    fi
fi

if test "$linkmode" = prog || test "$opt_mode" = relink; then
    add_shlibpath=
    add_dir=
    add=
    # Finalize command for both is simple: just hardcode it.
    if test "$hardcode_direct" = yes &&
        test "$hardcode_direct_absolute" = no; then
        add="$libdir/$linklib"
    elif test "$hardcode_minus_L" = yes; then
        add_dir="-L$libdir"
        add="-l$name"
    elif test "$hardcode_shlibpath_var" = yes; then
        case :$finalize_shlibpath: in
        *):$libdir:*) ;;
        *) finalize_shlibpath+="$libdir:" ;;
        esac
        add="-l$name"
    elif test "$hardcode_automatic" = yes; then
        if test -n "$inst_prefix_dir" &&
            test -f "$inst_prefix_dir$libdir/$linklib" ; then
            add="$inst_prefix_dir$libdir/$linklib"
        else
            add="$libdir/$linklib"
        fi
    else
        # We cannot seem to hardcode it, guess we'll fake it.
        add_dir="-L$lt_sysroot$libdir"
        # Try looking first in the location we're being installed
to.

        if test -n "$inst_prefix_dir"; then
        case $libdir in
        [\\/]*)
            add_dir+=" -L$inst_prefix_dir$libdir"
            ;;
        esac

```

```

        fi
        add="-l$name"
    fi

    if test "$linkmode" = prog; then
        test -n "$add_dir" && finalize_deplibs="$add_dir
$finalize_deplibs"
        test -n "$add" && finalize_deplibs="$add $finalize_deplibs"
    else
        test -n "$add_dir" && deplibs="$add_dir $deplibs"
        test -n "$add" && deplibs="$add $deplibs"
    fi
fi
elif test "$linkmode" = prog; then
    # Here we assume that one of hardcode_direct or
hardcode_minus_L
    # is not unsupported.  This is valid on all known static and
    # shared platforms.
    if test "$hardcode_direct" != unsupported; then
        test -n "$old_library" && linklib="$old_library"
        compile_deplibs="$dir/$linklib $compile_deplibs"
        finalize_deplibs="$dir/$linklib $finalize_deplibs"
    else
        compile_deplibs="-l$name -L$dir $compile_deplibs"
        finalize_deplibs="-l$name -L$dir $finalize_deplibs"
    fi
elif test "$build_libtool_libs" = yes; then
    # Not a shared library
    if test "$deplibs_check_method" != pass_all; then
        # We're trying link a shared library against a static one
        # but the system doesn't support it.

        # Just print a warning and add the library to dependency_libs
so
        # that the program can be linked against the static library.
        echo
        $ECHO "**** Warning: This system can not link to static lib
archive $lib."
        echo "**** I have the capability to make that library
automatically link in when"
        echo "**** you link to this library.  But I can only do this
if you have a"
        echo "**** shared version of the library, which you do not
appear to have."
        if test "$module" = yes; then
            echo "**** But as you try to build a module library, libtool
will still create "
            echo "**** a static module, that should work as long as the
dlopening application"
            echo "**** is linked with the -dlopen flag to resolve
symbols at runtime."
            if test -z "$global_symbol_pipe"; then

```

```

        echo
        echo "*** However, this would only work if libtool was able
to extract symbol"
        echo "*** lists from a program, using `nm' or equivalent,
but libtool could"
        echo "*** not find such a program. So, this module is
probably useless."
        echo "*** `nm' from GNU binutils and a full rebuild may
help."

        fi
        if test "$build_old_libs" = no; then
        build_libtool_libs=module
        build_old_libs=yes
        else
        build_libtool_libs=no
        fi
    fi
else
    deplibs="$dir/$old_library $deplibs"
    link_static=yes
fi
fi # link shared/static library?

if test "$linkmode" = lib; then
    if test -n "$dependency_libs" &&
    { test "$hardcode_into_libs" != yes ||
    test "$build_old_libs" = yes ||
    test "$link_static" = yes; }; then
    # Extract -R from dependency_libs
    temp_deplibs=
    for libdir in $dependency_libs; do
        case $libdir in
        -R*) func_stripname '-R' '' "$libdir"
            temp_xrpath=$func_stripname_result
            case " $xrpath " in
            *" $temp_xrpath ") ;;
            *) xrpath+="$temp_xrpath";;
            esac;;
        *) temp_deplibs+="$ libdir";;
        esac
    done
    dependency_libs="$temp_deplibs"
fi

newlib_search_path+="$absdir"
# Link against this library
test "$link_static" = no &&
newdependency_libs="$abs_ladir/$lname $newdependency_libs"
# ... and its dependency_libs
tmp_libs=
for deplib in $dependency_libs; do
    newdependency_libs="$deplib $newdependency_libs"

```

```

case $deplib in
    -L*) func_stripname '-L' '' "$deplib"
          func_resolve_sysroot "$func_stripname_result";;
    *) func_resolve_sysroot "$deplib" ;;
esac
if $opt_preserve_dup_deps ; then
case "$tmp_libs " in
    *" $func_resolve_sysroot_result "*)
        specialdeplibs+=" $func_resolve_sysroot_result" ;;
esac
fi
tmp_libs+=" $func_resolve_sysroot_result"
done

if test "$link_all_deplibs" != no; then
# Add the search paths of all dependency libraries
for deplib in $dependency_libs; do
    path=
    case $deplib in
        -L*) path="$deplib" ;;
        *.la)
            func_resolve_sysroot "$deplib"
            deplib=$func_resolve_sysroot_result
            func_dirname "$deplib" "" "."
            dir=$func_dirname_result
            # We need an absolute path.
            case $dir in
                [\./]* | [A-Za-z]:[\./]*) absdir="$dir" ;;
                *)
                    absdir=`cd "$dir" && pwd`
                    if test -z "$absdir"; then
                        func_warning "cannot determine absolute directory name
of `\$dir'"
                    fi
                    absdir="$dir"
            esac
            ;;
    esac
    if $GREP "^installed=no" $deplib > /dev/null; then
    case $host in
        *-*-darwin*)
            depdepl=
            eval deplibrary_names=`${SED} -n -e
's/^library_names=(.*)$/\1/p' $deplib`
            if test -n "$deplibrary_names" ; then
                for tmp in $deplibrary_names ; do
                    depdepl=$tmp
                done
                if test -f "$absdir/$objdir/$depdepl" ; then
                    depdepl="$absdir/$objdir/$depdepl"
                    darwin_install_name=`${OTOOL} -L $depdepl | awk '{if
(NR == 2) {print \$1;exit}}'`
                    if test -z "$darwin_install_name"; then

```



```

done
fi
if test "$pass" != dlopen; then
if test "$pass" != conv; then
    # Make sure lib_search_path contains only unique directories.
    lib_search_path=
    for dir in $newlib_search_path; do
        case "$lib_search_path " in
            *" $dir "*) ;;
            *) lib_search_path+=" $dir" ;;
        esac
    done
    newlib_search_path=
fi

if test "$linkmode,$pass" != "prog,link"; then
    vars="deplibs"
else
    vars="compile_deplibs finalize_deplibs"
fi
for var in $vars dependency_libs; do
    # Add libraries to $var in reverse order
    eval tmp_libs="\${$var}"
    new_libs=
    for deplib in $tmp_libs; do
        # FIXME: Pedantically, this is the right thing to do, so
        #         that some nasty dependency loop isn't accidentally
        #         broken:
        #new_libs="$deplib $new_libs"
        # Pragmatically, this seems to cause very few problems in
        # practice:
        case $deplib in
            -L*) new_libs="$deplib $new_libs" ;;
            -R*) ;;
            *)
                # And here is the reason: when a library appears more
                # than once as an explicit dependence of a library, or
                # is implicitly linked in more than once by the
                # compiler, it is considered special, and multiple
                # occurrences thereof are not removed. Compare this
                # with having the same library being listed as a
                # dependency of multiple other libraries: in this case,
                # we know (pedantically, we assume) the library does not
                # need to be listed more than once, so we keep only the
                # last copy. This is not always right, but it is rare
                # enough that we require users that really mean to play
                # such unportable linking tricks to link the library
                # using -Wl,-lname, so that libtool does not consider it
                # for duplicate removal.
                case " $specialdeplibs " in
                    *" $deplib "*) new_libs="$deplib $new_libs" ;;
                    *)

```

```

        case " $new_libs " in
        *" $deplib "*) ;;
        *) new_libs="$deplib $new_libs" ;;
        esac
        ;;
        esac
        ;;
    esac
done
tmp_libs=
for deplib in $new_libs; do
    case $deplib in
    -L*)
        case " $tmp_libs " in
        *" $deplib "*) ;;
        *) tmp_libs+="$deplib" ;;
        esac
        ;;
    *) tmp_libs+="$deplib" ;;
    esac
done
eval $var="\$tmp_libs\"
done # for var
fi
# Last step: remove runtime libs from dependency_libs
# (they stay in deplibs)
tmp_libs=
for i in $dependency_libs ; do
    case " $predeps $postdeps $compiler_lib_search_path " in
    *" $i "*)
        i=""
        ;;
    esac
    if test -n "$i" ; then
        tmp_libs+="$i"
    fi
done
dependency_libs=$tmp_libs
done # for pass
if test "$linkmode" = prog; then
    dlfiles="$newdlfiles"
fi
if test "$linkmode" = prog || test "$linkmode" = lib; then
    dlprefiles="$newdlprefiles"
fi

case $linkmode in
oldlib)
    if test -n "$dlfiles$dlprefiles" || test "$dlself" != no; then
        func_warning "\`-dlopen' is ignored for archives"
    fi

```



```

case " $deplibs" in
*\ -l* | *\ -L*)
func_warning "\`-l' and \`-L' are ignored for archives" ;;
esac

test -n "$rpath" && \
func_warning "\`-rpath' is ignored for archives"

test -n "$xrpath" && \
func_warning "\`-R' is ignored for archives"

test -n "$vinfo" && \
func_warning "\`-version-info/-version-number' is ignored for
archives"

test -n "$release" && \
func_warning "\`-release' is ignored for archives"

test -n "$export_symbols$export_symbols_regex" && \
func_warning "\`-export-symbols' is ignored for archives"

# Now set the variables for building old libraries.
build_libtool_libs=no
oldlibs="$output"
objs+="$old_deplibs"
;;

lib)
# Make sure we only generate libraries of the form `libNAME.la'.
case $outputname in
lib*)
func_stripname 'lib' '.la' "$outputname"
name=$func_stripname_result
eval shared_ext="\`$shrext_cmds\"
eval libname="\`$libname_spec\"
;;
*)
test "$module" = no && \
func_fatal_help "libtool library \`$output' must begin with
\`lib'"

if test "$need_lib_prefix" != no; then
# Add the "lib" prefix for modules if required
func_stripname ' ' '.la' "$outputname"
name=$func_stripname_result
eval shared_ext="\`$shrext_cmds\"
eval libname="\`$libname_spec\"
else
func_stripname ' ' '.la' "$outputname"
libname=$func_stripname_result
fi
;;

```

```

esac

if test -n "$objs"; then
if test "$deplibs_check_method" != pass_all; then
    func_fatal_error "cannot build libtool library ``$output' from
non-libtool objects on this host:$objs"
else
    echo
    $ECHO "*** Warning: Linking the shared library $output against
the non-libtool"
    $ECHO "*** objects $objs is not portable!"
    libobjs+="$objs"
fi
fi

test "$dlsel" != no && \
func_warning "\`-dlopen self' is ignored for libtool libraries"

set dummy $rpath
shift
test "$#" -gt 1 && \
func_warning "ignoring multiple \`-rpath's for a libtool library"

install_libdir="$1"

oldlibs=
if test -z "$rpath"; then
if test "$build_libtool_libs" = yes; then
    # Building a libtool convenience library.
    # Some compilers have problems with a `.al' extension so
    # convenience libraries should have the same extension an
    # archive normally would.
    oldlibs="$output_objdir/$libname.$libext $oldlibs"
    build_libtool_libs=convenience
    build_old_libs=yes
fi

test -n "$vinfo" && \
    func_warning "\`-version-info/-version-number' is ignored for
convenience libraries"

test -n "$release" && \
    func_warning "\`-release' is ignored for convenience libraries"
else

# Parse the version information argument.
save_ifs="$IFS"; IFS=:
set dummy $vinfo 0 0 0
shift
IFS="$save_ifs"

test -n "$7" && \

```

```

func_fatal_help "too many parameters to \`-version-info'"

# convert absolute version numbers to libtool ages
# this retains compatibility with .la files and attempts
# to make the code below a bit more comprehensible

case $vinfo_number in
yes)
  number_major="$1"
  number_minor="$2"
  number_revision="$3"
  #
  # There are really only two kinds -- those that
  # use the current revision as the major version
  # and those that subtract age and use age as
  # a minor version.  But, then there is irix
  # which has an extra 1 added just for fun
  #
  case $version_type in
# correct linux to gnu/linux during the next big refactor
darwin|linux|osf|windows|none)
  func_arith $number_major + $number_minor
  current=$func_arith_result
  age="$number_minor"
  revision="$number_revision"
  ;;
freebsd-aout|freebsd-elf|qnx|sunos)
  current="$number_major"
  revision="$number_minor"
  age="0"
  ;;
irix|nonstopux)
  func_arith $number_major + $number_minor
  current=$func_arith_result
  age="$number_minor"
  revision="$number_minor"
  lt_irix_increment=no
  ;;
esac
  ;;
no)
  current="$1"
  revision="$2"
  age="$3"
  ;;
esac

# Check that each of the things are valid numbers.
case $current in
0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-
9][0-9][0-9][0-9]) ;;
*)

```

```

    func_error "CURRENT \`${current}' must be a nonnegative integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

case $revision in
  0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-9][0-9][0-9][0-9]) ;;
  *)
    func_error "REVISION \`${revision}' must be a nonnegative
integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

case $age in
  0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-9][0-9][0-9][0-9]) ;;
  *)
    func_error "AGE \`${age}' must be a nonnegative integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

if test "$age" -gt "$current"; then
  func_error "AGE \`${age}' is greater than the current interface
number \`${current}'"
  func_fatal_error "\`${vinfo}' is not valid version information"
fi

# Calculate the version variables.
major=
versuffix=
verstring=
case $version_type in
  none) ;;

  darwin)
    # Like Linux, but with the current version available in
    # verstring for coding it into the library header
    func_arith $current - $age
    major=.$func_arith_result
    versuffix="$major.$age.$revision"
    # Darwin ld doesn't like 0 for these options...
    func_arith $current + 1
    minor_current=$func_arith_result
    xlcverstring="{wl}-compatibility_version {wl}$minor_current
${wl}-current_version {wl}$minor_current.$revision"
    verstring="-compatibility_version $minor_current -
current_version $minor_current.$revision"
    ;;

```

```

freebsd-aout)
    major=".${current}"
    versuffix=".${current}.${revision}";
    ;;

freebsd-elf)
    major=".${current}"
    versuffix=".${current}"
    ;;

irix | nonstopux)
    if test "X${lt_irix_increment}" = "Xno"; then
        func_arith $current - $age
    else
        func_arith $current - $age + 1
    fi
    major=${func_arith_result}

    case $version_type in
        nonstopux) verstring_prefix=nonstopux ;;
        *)         verstring_prefix=sgi ;;
    esac
    verstring="${verstring_prefix}${major}.${revision}"

    # Add in all the interfaces that we are compatible with.
    loop=$revision
    while test "$loop" -ne 0; do
        func_arith $revision - $loop
        iface=${func_arith_result}
        func_arith $loop - 1
        loop=${func_arith_result}
        verstring="${verstring_prefix}${major}.${iface}:${verstring}"
    done

    # Before this point, $major must not contain `.'.
    major=.$major
    versuffix="${major}.${revision}"
    ;;

linux) # correct to gnu/linux during the next big refactor
    func_arith $current - $age
    major=.$func_arith_result
    versuffix="${major}.${age}.${revision}"
    ;;

osf)
    func_arith $current - $age
    major=.$func_arith_result
    versuffix=".${current}.${age}.${revision}"
    verstring="${current}.${age}.${revision}"

    # Add in all the interfaces that we are compatible with.

```

```

loop=$age
while test "$loop" -ne 0; do
    func_arith $current - $loop
    iface=$func_arith_result
    func_arith $loop - 1
    loop=$func_arith_result
    verstring="$verstring:${iface}.0"
done

# Make executables depend on our current version.
verstring+="${current}.0"
;;

qnx)
    major=".$current"
    versuffix=".$current"
    ;;

sunos)
    major=".$current"
    versuffix=".$current.$revision"
    ;;

windows)
    # Use '-' rather than '.', since we only want one
    # extension on DOS 8.3 filesystems.
    func_arith $current - $age
    major=$func_arith_result
    versuffix="-${major}"
    ;;

*)
    func_fatal_configuration "unknown library version type
\`${version_type}`"
    ;;
esac

# Clear the version info if we defaulted, and they specified a
release.
if test -z "$vinfo" && test -n "$release"; then
    major=
    case $version_type in
    darwin)
        # we can't check for "0.0" in archive_cmds due to quoting
        # problems, so we reset it completely
        verstring=
        ;;
    *)
        verstring="0.0"
        ;;
    esac
    if test "$need_version" = no; then

```

```

        versuffix=
    else
        versuffix=".0.0"
    fi
fi

# Remove version info from name if versioning should be avoided
if test "$avoid_version" = yes && test "$need_version" = no; then
    major=
    versuffix=
    verstring=""
fi

# Check to see if the archive will have undefined symbols.
if test "$allow_undefined" = yes; then
    if test "$allow_undefined_flag" = unsupported; then
        func_warning "undefined symbols not allowed in $host shared
libraries"
        build_libtool_libs=no
        build_old_libs=yes
    fi
else
    # Don't allow undefined symbols.
    allow_undefined_flag="$no_undefined_flag"
fi

fi

func_generate_dlsyms "$libname" "$libname" "yes"
libobjs+=" $symfileobj"
test "X$libobjs" = "X " && libobjs=

if test "$opt_mode" != relink; then
# Remove our outputs, but don't remove object files since they
# may have been created when compiling PIC objects.
removelist=
tempremovelist=`$ECHO "$output_objdir/*" `
for p in $tempremovelist; do
    case $p in
        *.$objext | *.gcno)
            ;;
        $output_objdir/$outputname | $output_objdir/$libname.* |
$output_objdir/${libname}${release}.* )
            if test "X$precious_files_regex" != "X"; then
                if $ECHO "$p" | $EGREP -e "$precious_files_regex"
>/dev/null 2>&1
            then
                continue
            fi
        fi
        removelist+=" $p"
    ;;

```

```

        *) ;;
    esac
done
test -n "$removelist" && \
    func_show_eval "${RM}r \${removelist}"
fi

# Now set the variables for building old libraries.
if test "$build_old_libs" = yes && test "$build_libtool_libs" !=
convenience ; then
    oldlibs+=" $output_objdir/$libname.$libext"

    # Transform .lo files to .o files.
    oldobjs="$objs "`$ECHO "$libobjs" | $SP2NL | $SED
"/\.${libext}$/d; $lo2o" | $NL2SP`
    fi

    # Eliminate all temporary directories.
    #for path in $notinst_path; do
    #    lib_search_path=`$ECHO "$lib_search_path " | $SED "s% $path
% %g"`
    #    deplibs=`$ECHO "$deplibs " | $SED "s% -L$path % %g"`
    #    dependency_libs=`$ECHO "$dependency_libs " | $SED "s% -
L$path % %g"`
    #done

    if test -n "$xrpath"; then
    # If the user specified any rpath flags, then add them.
    temp_xrpath=
    for libdir in $xrpath; do
        func_replace_sysroot "$libdir"
        temp_xrpath+=" -R${func_replace_sysroot_result}"
    case "$finalize_rpath " in
        *" $libdir "*) ;;
        *) finalize_rpath+=" $libdir" ;;
    esac
    done
    if test "$hardcode_into_libs" != yes || test "$build_old_libs" =
yes; then
        dependency_libs="$temp_xrpath $dependency_libs"
    fi
    fi

    # Make sure dlfiles contains only unique files that won't be
dlpreopened
    old_dlfiles="$dlfiles"
    dlfiles=
    for lib in $old_dlfiles; do
    case " $dlprefiles $dlfiles " in
        *" $lib "*) ;;
        *) dlfiles+=" $lib" ;;
    esac

```



```

done

# Make sure dlprefiles contains only unique files
old_dlprefiles="$dlprefiles"
dlprefiles=
for lib in $old_dlprefiles; do
case "$dlprefiles " in
*" $lib ") ;;
*) dlprefiles+=" $lib" ;;
esac
done

if test "$build_libtool_libs" = yes; then
if test -n "$rpath"; then
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-beos* |
*-cegcc* | *-*-haiku*)
# these systems don't actually have a c library (as such)!
;;
*-*-rhapsody* | *-*-darwin1.[012])
# Rhapsody C library is in the System framework
deplibs+=" System.ltframework"
;;
*-*-netbsd*)
# Don't link with libc until the a.out ld.so is fixed.
;;
*-*-openbsd* | *-*-freebsd* | *-*-dragonfly*)
# Do not include libc due to us having libc/libc_r.
;;
*-*-sco3.2v5* | *-*-sco5v6*)
# Causes problems with __ctype
;;
*-*-sysv4.2uw2* | *-*-sysv5* | *-*-unixware* | *-*-OpenUNIX*)
# Compiler inserts libc in the correct place for threads to
work
;;
*)
# Add libc to deplibs on all other systems if necessary.
if test "$build_libtool_need_lc" = "yes"; then
deplibs+=" -lc"
fi
;;
esac
fi

# Transform deplibs into only deplibs that can be linked in
shared.
name_save=$name
libname_save=$libname
release_save=$release
versuffix_save=$versuffix
major_save=$major

```

```

# I'm not sure if I'm treating the release correctly.  I think
# release should show up in the -l (ie -lgmp5) so we don't want
to
# add it in twice.  Is that correct?
release=""
versuffix=""
major=""
newdeplibs=
droppeddeps=no
case $deplibs_check_method in
pass_all)
# Don't check for shared/static.  Everything works.
# This might be a little naive.  We might want to check
# whether the library exists or not.  But this is on
# osf3 & osf4 and I'm not really sure...  Just
# implementing what was already the behavior.
newdeplibs=$deplibs
;;
test_compile)
# This code stresses the "libraries are programs" paradigm to
its
# limits.  Maybe even breaks it.  We compile a program, linking
it
# against the deplibs as a proxy for the library.  Then we can
check
# whether they linked in statically or dynamically with ldd.
$opt_dry_run || $RM conftest.c
cat > conftest.c <<EOF
EOF
int main() { return 0; }

$opt_dry_run || $RM conftest
if $LTCC $LTCFLAGS -o conftest conftest.c $deplibs; then
  ldd_output=`ldd conftest`
  for i in $deplibs; do
    case $i in
      -l*)
        func_stripname -l '' "$i"
        name=$func_stripname_result
        if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
          case " $predeps $postdeps " in
            *" $i "*)
              newdeplibs+=" $i"
              i=""
            ;;
          esac
        fi
        if test -n "$i" ; then
          libname=`eval "\\$ECHO \"\$libname_spec\""`
          deplib_matches=`eval "\\$ECHO \"\$library_names_spec\""`
          set dummy $deplib_matches; shift
          deplib_match=$1

```

```

        if test `expr "$ldd_output" : ".*$deplib_match"` -ne 0 ;
then
    newdeplibs+=" $i"
else
    droppeddeps=yes
    echo
    $ECHO "*** Warning: dynamic linker does not accept
needed library $i."
    echo "*** I have the capability to make that library
automatically link in when"
    echo "*** you link to this library.  But I can only do
this if you have a"
    echo "*** shared version of the library, which I
believe you do not have"
    echo "*** because a test_compile did reveal that the
linker did not use it for"
    echo "*** its dynamic dependency list that programs get
resolved with at runtime."
    fi
    fi
    ;;
    *)
    newdeplibs+=" $i"
    ;;
esac
done
else
# Error occurred in the first compile.  Let's try to salvage
# the situation: Compile a separate program for each library.
for i in $deplibs; do
    case $i in
    -l*)
    func_stripname -l '' "$i"
    name=$func_stripname_result
    $opt_dry_run || $RM conftest
    if $LTCC $LTCFLAGS -o conftest conftest.c $i; then
        ldd_output=`ldd conftest`
        if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
            case " $predeps $postdeps " in
            *" $i "*)
                newdeplibs+=" $i"
                i=""
                ;;
            esac
        fi
        if test -n "$i" ; then
            libname=`eval "\\$ECHO \"\$libname_spec\""`
            deplib_matches=`eval "\\$ECHO \"\$library_names_spec\""`
            set dummy $deplib_matches; shift
            deplib_match=$1

```

```

        if test `expr "$ldd_output" : ".*$deplib_match"` -ne 0
; then
        newdeplibs+=" $i"
    else
        droppeddeps=yes
        echo
        $ECHO "*** Warning: dynamic linker does not accept
needed library $i."
        echo "*** I have the capability to make that library
automatically link in when"
        echo "*** you link to this library. But I can only
do this if you have a"
        echo "*** shared version of the library, which you do
not appear to have"
        echo "*** because a test_compile did reveal that the
linker did not use this one"
        echo "*** as a dynamic dependency that programs can
get resolved with at runtime."
        fi
    fi
    else
        droppeddeps=yes
        echo
        $ECHO "*** Warning! Library $i is needed by this library
but I was not able to"
        echo "*** make it link in! You will probably need to
install it or some"
        echo "*** library that it depends on before this library
will be fully"
        echo "*** functional. Installing it before continuing
would be even better."
        fi
    ;;
    *)
        newdeplibs+=" $i"
    ;;
esac
done
fi
;;
file_magic*)
set dummy $deplibs_check_method; shift
file_magic_regex=`expr "$deplibs_check_method" : "$1 \(.*)"`
for a_deplib in $deplibs; do
    case $a_deplib in
    -l*)
        func_stripname -l '' "$a_deplib"
        name=$func_stripname_result
        if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
            case " $predeps $postdeps " in
            *" $a_deplib ")

```

```

        newdeplibs+=" $a_deplib"
        a_deplib=""
        ;;
    esac
    fi
    if test -n "$a_deplib" ; then
        libname=`eval "\\$ECHO \"$libname_spec\""`
        if test -n "$file_magic_glob"; then
            libnameglob=`func_echo_all "$libname" | $SED -e
$file_magic_glob`
        else
            libnameglob=$libname
        fi
        test "$want_nocaseglob" = yes && nocaseglob=`shopt -p
nocaseglob`
        for i in $lib_search_path $sys_lib_search_path
$shlib_search_path; do
            if test "$want_nocaseglob" = yes; then
                shopt -s nocaseglob
                potential_libs=`ls $i/$libnameglob[.]* 2>/dev/null`
                $nocaseglob
            else
                potential_libs=`ls $i/$libnameglob[.]* 2>/dev/null`
            fi
            for potent_lib in $potential_libs; do
                # Follow soft links.
                if ls -lLd "$potent_lib" 2>/dev/null |
                $GREP " -> " >/dev/null; then
                    continue
                fi
                # The statement above tries to avoid entering an
                # endless loop below, in case of cyclic links.
                # We might still enter an endless loop, since a link
                # loop can be closed while we follow links,
                # but so what?
                potlib="$potent_lib"
                while test -h "$potlib" 2>/dev/null; do
                    potliblink=`ls -ld $potlib | ${SED} 's/. * -> //'`
                    case $potliblink in
                        [\\/*]* | [A-Za-z]:[\\/*]*) potlib="$potliblink";;
                        *) potlib=`$ECHO "$potlib" | $SED
's,[^/]*$,,'`"$potliblink";;
                    esac
                done
                if eval $file_magic_cmd "\\$potlib" 2>/dev/null |
                $SED -e 10q |
                $EGREP "$file_magic_regex" > /dev/null; then
                    newdeplibs+=" $a_deplib"
                    a_deplib=""
                    break 2
                fi
            done
        done
    fi
done

```

```

done
fi
if test -n "$a_deplib" ; then
droppeddeps=yes
echo
$ECHO "*** Warning: linker path does not have real file for
library $a_deplib."
echo "*** I have the capability to make that library
automatically link in when"
echo "*** you link to this library. But I can only do this
if you have a"
echo "*** shared version of the library, which you do not
appear to have"
echo "*** because I did check the linker path looking for a
file starting"
if test -z "$spotlib" ; then
$ECHO "*** with $libname but no candidates were found.
(...for file magic test)"
else
$ECHO "*** with $libname and none of the candidates
passed a file format test"
$ECHO "*** using a file magic. Last file checked:
$spotlib"
fi
fi
;;
*)
# Add a -L argument.
newdeplibs+=" $a_deplib"
;;
esac
done # Gone through all deplibs.
;;
match_pattern*)
set dummy $deplibs_check_method; shift
match_pattern_regex=`expr "$deplibs_check_method" : "$1
\(.*\)"`
for a_deplib in $deplibs; do
case $a_deplib in
-1*)
func_stripname -1 '' "$a_deplib"
name=$func_stripname_result
if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
case " $predeps $postdeps " in
*" $a_deplib ")
newdeplibs+=" $a_deplib"
a_deplib=""
;;
esac
fi
if test -n "$a_deplib" ; then

```

```

        libname=`eval "\\$ECHO \"\$libname_spec\""`
        for i in $lib_search_path $sys_lib_search_path
$shlib_search_path; do
            potential_libs=`ls $i/$libname[.]* 2>/dev/null`
            for potent_lib in $potential_libs; do
                potlib="$potent_lib" # see symlink-check above in
file_magic test
                if eval "\$ECHO \"\$potent_lib\" 2>/dev/null | $SED 10q
| \
                    $EGREP "$match_pattern_regex" > /dev/null; then
                        newdeplibs+=" $a_deplib"
                        a_deplib=""
                        break 2
                    fi
                done
            done
            fi
            if test -n "$a_deplib" ; then
                droppeddeps=yes
                echo
                $ECHO "*** Warning: linker path does not have real file for
library $a_deplib."
                echo "*** I have the capability to make that library
automatically link in when"
                echo "*** you link to this library. But I can only do this
if you have a"
                echo "*** shared version of the library, which you do not
appear to have"
                echo "*** because I did check the linker path looking for a
file starting"
                if test -z "$potlib" ; then
                    $ECHO "*** with $libname but no candidates were found.
(...for regex pattern test)"
                else
                    $ECHO "*** with $libname and none of the candidates
passed a file format test"
                    $ECHO "*** using a regex pattern. Last file checked:
$potlib"
                fi
            fi
            ;;
        *)
            # Add a -L argument.
            newdeplibs+=" $a_deplib"
            ;;
        esac
    done # Gone through all deplibs.
    ;;
    none | unknown | *)
        newdeplibs=""
        tmp_deplibs=`$ECHO " $deplibs" | $SED 's/ -lc$//; s/ -[LR][^
]*//g'`

```

```

        if test "X$allow_libtool_libs_with_static_runtimes" = "Xyes" ;
then
    for i in $predeps $postdeps ; do
        # can't use Xsed below, because $i might contain '/'
        tmp_deplibs=`$ECHO " $tmp_deplibs" | $SED "s,$i,, "`
    done
fi
case $tmp_deplibs in
*(!\      \ ]*)
    echo
    if test "X$dreplibs_check_method" = "Xnone"; then
        echo "*** Warning: inter-library dependencies are not
supported in this platform."
    else
        echo "*** Warning: inter-library dependencies are not known
to be supported."
    fi
    echo "*** All declared inter-library dependencies are being
dropped."
    droppeddeps=yes
    ;;
esac
;;
esac
versuffix=$versuffix_save
major=$major_save
release=$release_save
libname=$libname_save
name=$name_save

case $host in
*-*-rhapsody* | *-*-darwin1.[012])
    # On Rhapsody replace the C library with the System framework
    newdeplibs=`$ECHO " $newdeplibs" | $SED 's/ -lc /
System.ltframework /'`
    ;;
esac

if test "$droppeddeps" = yes; then
    if test "$module" = yes; then
        echo
        echo "*** Warning: libtool could not satisfy all declared
inter-library"
        $ECHO "*** dependencies of module $libname.  Therefore,
libtool will create"
        echo "*** a static module, that should work as long as the
dlopening"
        echo "*** application is linked with the -dlopen flag."
        if test -z "$global_symbol_pipe"; then
            echo
            echo "*** However, this would only work if libtool was able
to extract symbol"

```



```

        echo "*** lists from a program, using `nm` or equivalent,
but libtool could"
        echo "*** not find such a program.  So, this module is
probably useless."
        echo "*** `nm` from GNU binutils and a full rebuild may
help."
    fi
    if test "$build_old_libs" = no; then
        oldlibs="$output_objdir/$libname.$libext"
        build_libtool_libs=module
        build_old_libs=yes
    else
        build_libtool_libs=no
    fi
else
    echo "*** The inter-library dependencies that have been
dropped here will be"
    echo "*** automatically added whenever a program is linked
with this library"
    echo "*** or is declared to -dlopen it."

    if test "$allow_undefined" = no; then
        echo
        echo "*** Since this library must not contain undefined
symbols,"
        echo "*** because either the platform does not support them
or"

        echo "*** it was explicitly requested with -no-undefined,"
        echo "*** libtool will only create a static version of it."
        if test "$build_old_libs" = no; then
            oldlibs="$output_objdir/$libname.$libext"
            build_libtool_libs=module
            build_old_libs=yes
        else
            build_libtool_libs=no
        fi
    fi
fi
fi
fi
# Done checking deplibs!
deplibs=$newdeplibs
fi
# Time to change all our "foo.ltframework" stuff back to "-
framework foo"
case $host in
*-*-darwin*)
    newdeplibs=`$ECHO " $newdeplibs" | $SED 's% \([^
$]*\)\.ltframework% -framework \1%g'\`
    new_inherited_linker_flags=`$ECHO "
$new_inherited_linker_flags" | $SED 's% \([^
$]*\)\.ltframework% -
framework \1%g'\`

```

```

        deplibs=`$ECHO " $deplibs" | $SED 's% \([^ $]*\).ltframework% -
framework \1%g'`
        ;;
    esac

    # move library search paths that coincide with paths to not yet
    # installed libraries to the beginning of the library search
list
    new_libs=
    for path in $notinst_path; do
    case " $new_libs " in
    *" -L$path/$objdir "*) ;;
    *)
        case " $deplibs " in
        *" -L$path/$objdir "*)
            new_libs+=" -L$path/$objdir" ;;
        esac
        ;;
    esac
    done
    for deplib in $deplibs; do
    case $deplib in
    -L*)
        case " $new_libs " in
        *" $deplib "*) ;;
        *) new_libs+=" $deplib" ;;
        esac
        ;;
    *) new_libs+=" $deplib" ;;
    esac
    done
    deplibs="$new_libs"

    # All the library-specific variables (install_libdir is set
    above).
    library_names=
    old_library=
    dlname=

    # Test again, we may have decided not to build it any more
    if test "$build_libtool_libs" = yes; then
    # Remove ${wl} instances when linking with ld.
    # FIXME: should test the right_cmds variable.
    case $archive_cmds in
    *\$LD\ *) wl= ;;
    esac
    if test "$hardcode_into_libs" = yes; then
    # Hardcode the library paths
    hardcode_libdirs=
    dep_rpath=
    rpath="$finalize_rpath"
    test "$opt_mode" != relink && rpath="$compile_rpath$rpath"

```

```

for libdir in $rpath; do
    if test -n "$hardcode_libdir_flag_spec"; then
        func_replace_sysroot "$libdir"
        libdir=$func_replace_sysroot_result
        func_stripname '=' '' "$libdir"
        libdir=$func_stripname_result
        if test -n "$hardcode_libdir_separator"; then
            if test -z "$hardcode_libdirs"; then
                hardcode_libdirs="$libdir"
            else
                # Just accumulate the unique libdirs.
                case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
                *"$hardcode_libdir_separator$libdir$hardcode_libdir_separator"*)
                    ;;
                *)
                    hardcode_libdirs+="$hardcode_libdir_separator$libdir"
                    ;;
                esac
            fi
        else
            # We only want to hardcode in an rpath if it isn't in
the
            # default dlsearch path.
            func_normal_abspath "$libdir"
            libdir_norm=$func_normal_abspath_result
            case " $sys_lib_dlsearch_path " in
                * $libdir_norm *) ;;
                *) eval flag=\"$hardcode_libdir_flag_spec\"
                    dep_rpath+="$flag"
                    ;;
            esac
        fi
    elif test -n "$runpath_var"; then
        case "$perm_rpath " in
            * $libdir *) ;;
            *) perm_rpath+="$libdir" ;;
        esac
    fi
done
# Substitute the hardcoded libdirs into the rpath.
if test -n "$hardcode_libdir_separator" &&
    test -n "$hardcode_libdirs"; then
    libdir="$hardcode_libdirs"
    eval "dep_rpath=\"$hardcode_libdir_flag_spec\""
fi
if test -n "$runpath_var" && test -n "$perm_rpath"; then
    # We should set the runpath_var.
    rpath=
    for dir in $perm_rpath; do

```

```

        rpath+=" $dir:"
    done
    eval "$runpath_var=' $rpath\$$runpath_var'; export
$runpath_var"
    fi
    test -n "$dep_rpath" && deplibs="$dep_rpath $deplibs"
fi

shlibpath="$finalize_shlibpath"
test "$opt_mode" != relink &&
shlibpath="$compile_shlibpath$shlibpath"
if test -n "$shlibpath"; then
    eval "$shlibpath_var=' $shlibpath\$$shlibpath_var'; export
$shlibpath_var"
fi

# Get the real and link names of the library.
eval shared_ext="\$shrext_cmds\"
eval library_names="\$library_names_spec\"
set dummy $library_names
shift
realname="$1"
shift

if test -n "$soname_spec"; then
    eval soname="\$soname_spec\"
else
    soname="$realname"
fi
if test -z "$dlname"; then
    dlname=$soname
fi

lib="$output_objdir/$realname"
linknames=
for link
do
    linknames+=" $link"
done

# Use standard objects if they are pic
test -z "$pic_flag" && libobjs=`$ECHO "$libobjs" | $SP2NL | $SED
"$lo2o" | $NL2SP`
test "X$libobjs" = "X " && libobjs=

delfiles=
if test -n "$export_symbols" && test -n "$include_expsyms"; then
    $opt_dry_run || cp "$export_symbols"
"$output_objdir/$libname.uexp"
    export_symbols="$output_objdir/$libname.uexp"
    delfiles+=" $export_symbols"
fi

```

```

orig_export_symbols=
case $host_os in
cygwin* | mingw* | cegcc*)
    if test -n "$export_symbols" && test -z
"$export_symbols_regex"; then
        # exporting using user supplied symfile
        if test "x`$SED 1q $export_symbols`" != xEXPORTS; then
            # and it's NOT already a .def file. Must figure out
            # which of the given symbols are data symbols and tag
            # them as such. So, trigger use of export_symbols_cmds.
            # export_symbols gets reassigned inside the "prepare
            # the list of exported symbols" if statement, so the
            # include_expsyms logic still works.
            orig_export_symbols="$export_symbols"
            export_symbols=
            always_export_symbols=yes
        fi
    fi
;;
esac

# Prepare the list of exported symbols
if test -z "$export_symbols"; then
    if test "$always_export_symbols" = yes || test -n
"$export_symbols_regex"; then
        func_verbose "generating symbol list for `\$libname.la'"
        export_symbols="$output_objdir/$libname.exp"
        $opt_dry_run || $RM $export_symbols
        cmds=$export_symbols_cmds
        save_ifs="$IFS"; IFS='~'
        for cmd1 in $cmds; do
            IFS="$save_ifs"
            # Take the normal branch if the nm_file_list_spec branch
            # doesn't work or if tool conversion is not needed.
            case $nm_file_list_spec~$to_tool_file_cmd in
                *~func_convert_file_noop | *~func_convert_file_msys_to_w32
| ~*)
                    try_normal_branch=yes
                    eval cmd=\`$cmd1\`
                    func_len " $cmd"
                    len=${func_len_result}
                    ;;
                *)
                    try_normal_branch=no
                    ;;
            esac
            if test "$try_normal_branch" = yes \
&& { test "$len" -lt "$max_cmd_len" \
|| test "$max_cmd_len" -le -1; }
            then
                func_show_eval "$cmd" 'exit $?'

```

```

        skipped_export=false
        elif test -n "$nm_file_list_spec"; then
        func_basename "$output"
        output_la=${func_basename_result}
        save_libobjs=${libobjs}
        save_output=$output
        output=${output_objdir}/${output_la}.nm
        func_to_tool_file "$output"
        libobjs=$nm_file_list_spec${func_to_tool_file_result}
        delfiles+= " $output"
        func_verbose "creating $NM input file list: $output"
        for obj in $save_libobjs; do
            func_to_tool_file "$obj"
            $ECHO "$func_to_tool_file_result"
        done > "$output"
        eval cmd=\"\$cmd1\"
        func_show_eval "$cmd" 'exit $?'
        output=$save_output
        libobjs=$save_libobjs
        skipped_export=false
        else
        # The command line is too long to execute in one step.
        func_verbose "using reloadable object file for export
list..."
        skipped_export=:
        # Break out early, otherwise skipped_export may be
        # set to false by a later but shorter cmd.
        break
        fi
    done
    IFS=$save_ifs
    if test -n "$export_symbols_regex" && test "X$skipped_export"
!= "X:"; then
        func_show_eval '$EGREP -e "$export_symbols_regex"
"$export_symbols" > "${export_symbols}T"'
        func_show_eval '$MV "${export_symbols}T" "$export_symbols"'
    fi
    fi
    fi

    if test -n "$export_symbols" && test -n "$include_expsyms"; then
        tmp_export_symbols="$export_symbols"
        test -n "$orig_export_symbols" &&
tmp_export_symbols="$orig_export_symbols"
        $opt_dry_run || eval '$ECHO "$include_expsyms" | $SP2NL >>
"$tmp_export_symbols"'
    fi

    if test "X$skipped_export" != "X:" && test -n
"$orig_export_symbols"; then
        # The given exports_symbols file has to be filtered, so filter
it.

```

```

        func_verbose "filter symbol list for \`${libname}.la' to tag DATA
exports"
    # FIXME: $output_objdir/${libname}.filter potentially contains
lots of
    # 's' commands which not all seds can handle. GNU sed should be
fine
    # though. Also, the filter scales superlinearly with the number
of
    # global variables. join(1) would be nice here, but
unfortunately
    # isn't a blessed tool.
    $opt_dry_run || $SED -e '/[ ,]DATA;!d;s,\(.*\)\([
\,].*\),s|^\\1$|\\1\\2|,' < $export_symbols >
$output_objdir/${libname}.filter
    delfiles+=" $export_symbols $output_objdir/${libname}.filter"
    export_symbols=$output_objdir/${libname}.def
    $opt_dry_run || $SED -f $output_objdir/${libname}.filter <
$orig_export_symbols > $export_symbols
    fi

    tmp_deplibs=
    for test_deplib in $deplibs; do
        case "$convenience" in
            *" $test_deplib ") ;;
            *)
                tmp_deplibs+=" $test_deplib"
                ;;
        esac
    done
    deplibs="$tmp_deplibs"

    if test -n "$convenience"; then
        if test -n "$whole_archive_flag_spec" &&
            test "$compiler_needs_object" = yes &&
            test -z "$libobjs"; then
            # extract the archives, so we have objects to list.
            # TODO: could optimize this to just extract one archive.
            whole_archive_flag_spec=
        fi
        if test -n "$whole_archive_flag_spec"; then
            save_libobjs=$libobjs
            eval libobjs=\"\$libobjs $whole_archive_flag_spec\"
            test "X$libobjs" = "X " && libobjs=
        else
            gentop="$output_objdir/${outputname}x"
            generated+=" $gentop"

            func_extract_archives $gentop $convenience
            libobjs+=" $func_extract_archives_result"
            test "X$libobjs" = "X " && libobjs=
        fi
    fi
fi

```

```

    if test "$thread_safe" = yes && test -n "$thread_safe_flag_spec";
then
    eval flag="\$thread_safe_flag_spec\"
    linker_flags+=" $flag"
fi

# Make a backup of the uninstalled library when relinking
if test "$opt_mode" = relink; then
    $opt_dry_run || eval '(cd $output_objdir && $RM ${realname}U &&
$MV $realname ${realname}U)' || exit $?
fi

# Do each of the archive commands.
if test "$module" = yes && test -n "$module_cmds" ; then
    if test -n "$export_symbols" && test -n "$module_expsym_cmds";
then
        eval test_cmds="\$module_expsym_cmds\"
        cmds=$module_expsym_cmds
    else
        eval test_cmds="\$module_cmds\"
        cmds=$module_cmds
    fi
else
    if test -n "$export_symbols" && test -n "$archive_expsym_cmds";
then
        eval test_cmds="\$archive_expsym_cmds\"
        cmds=$archive_expsym_cmds
    else
        eval test_cmds="\$archive_cmds\"
        cmds=$archive_cmds
    fi
fi

if test "X$skipped_export" != "X:" &&
func_len "$test_cmds" &&
len=$func_len_result &&
test "$len" -lt "$max_cmd_len" || test "$max_cmd_len" -le -1;
then
    :
else
    # The command line is too long to link in one step, link
piecewise
    # or, if using GNU ld and skipped_export is not :, use a linker
    # script.

    # Save the value of $output and $libobjs because we want to
    # use them later.  If we have whole_archive_flag_spec, we
    # want to use save_libobjs as it was before
    # whole_archive_flag_spec was expanded, because we can't
    # assume the linker understands whole_archive_flag_spec.
    # This may have to be revisited, in case too many

```



```

# convenience libraries get linked in and end up exceeding
# the spec.
if test -z "$convenience" || test -z
"$whole_archive_flag_spec"; then
    save_libobjs=$libobjs
fi
save_output=$output
func_basename "$output"
output_la=$func_basename_result

# Clear the reloadable object creation command queue and
# initialize k to one.
test_cmds=
concat_cmds=
objlist=
last_robj=
k=1

if test -n "$save_libobjs" && test "X$skipped_export" != "X:"
&& test "$with_gnu_ld" = yes; then
    output=${output_objdir}/${output_la}.lnkscript
    func_verbose "creating GNU ld script: $output"
    echo 'INPUT (' > $output
    for obj in $save_libobjs
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result" >> $output
    done
    echo ')' >> $output
    delfiles+=" $output"
    func_to_tool_file "$output"
    output=$func_to_tool_file_result
elif test -n "$save_libobjs" && test "X$skipped_export" != "X:"
&& test "X$file_list_spec" != X; then
    output=${output_objdir}/${output_la}.lnk
    func_verbose "creating linker input file list: $output"
    : > $output
    set x $save_libobjs
    shift
    firstobj=
    if test "$compiler_needs_object" = yes; then
        firstobj="$1 "
        shift
    fi
    for obj
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result" >> $output
    done
    delfiles+=" $output"
    func_to_tool_file "$output"
    output=$firstobj\"$file_list_spec$func_to_tool_file_result\"

```

```

else
  if test -n "$save_libobjs"; then
    func_verbose "creating reloadable object files..."
    output=$output_objdir/$output_la-${k}.$objext
    eval test_cmds="\$reload_cmds\"
    func_len " $test_cmds"
    len0=$func_len_result
    len=$len0

    # Loop over the list of objects to be linked.
    for obj in $save_libobjs
    do
      func_len " $obj"
      func_arith $len + $func_len_result
      len=$func_arith_result
      if test "X$objlist" = X ||
         test "$len" -lt "$max_cmd_len"; then
        objlist+="$obj"
      else
        # The command $test_cmds is almost too long, add a
        # command to the queue.
        if test "$k" -eq 1 ; then
          # The first file doesn't have a previous command to
add.

          reload_objs=$objlist
          eval concat_cmds="\$reload_cmds\"
        else
          # All subsequent reloadable object files will link in
          # the last one created.
          reload_objs="$objlist $last_robj"
          eval concat_cmds="\$concat_cmds~$reload_cmds~\$RM
$last_robj\"
        fi
        last_robj=$output_objdir/$output_la-${k}.$objext
        func_arith $k + 1
        k=$func_arith_result
        output=$output_objdir/$output_la-${k}.$objext
        objlist="$obj"
        func_len " $last_robj"
        func_arith $len0 + $func_len_result
        len=$func_arith_result
      fi
    done
    # Handle the remaining objects by creating one last
    # reloadable object file. All subsequent reloadable object
    # files will link in the last one created.
    test -z "$concat_cmds" || concat_cmds=$concat_cmds~
    reload_objs="$objlist $last_robj"
    eval concat_cmds="\${concat_cmds}$reload_cmds\"
    if test -n "$last_robj"; then
      eval concat_cmds="\${concat_cmds}~\$RM $last_robj\"
    fi
  fi

```

```

    delfiles+=" $output"

else
    output=
fi

if ${skipped_export-false}; then
    func_verbose "generating symbol list for \`${libname}.la'"
    export_symbols="$output_objdir/${libname}.exp"
    $opt_dry_run || $RM $export_symbols
    libobjs=$output
    # Append the command to create the export file.
    test -z "$concat_cmds" || concat_cmds=$concat_cmds~
    eval concat_cmds="\`${concat_cmds}$export_symbols_cmds\"
    if test -n "$last_robj"; then
        eval concat_cmds="\`${concat_cmds}~\`$RM $last_robj\"
    fi
fi

test -n "$save_libobjs" &&
    func_verbose "creating a temporary reloadable object file:
$output"

# Loop through the commands generated above and execute them.
save_ifs="$IFS"; IFS='~'
for cmd in $concat_cmds; do
    IFS="$save_ifs"
    $opt_silent || {
        func_quote_for_expand "$cmd"
        eval "func_echo $func_quote_for_expand_result"
    }
    $opt_dry_run || eval "$cmd" || {
        lt_exit=$?

        # Restore the uninstalled library and exit
        if test "$opt_mode" = relink; then
            ( cd "$output_objdir" && \
              $RM "${realname}T" && \
              $MV "${realname}U" "$realname" )
        fi

        exit $lt_exit
    }
done
IFS="$save_ifs"

if test -n "$export_symbols_regex" && ${skipped_export-
false}; then
    func_show_eval '$EGREP -e "$export_symbols_regex"
"$export_symbols" > "${export_symbols}T"'
    func_show_eval '$MV "${export_symbols}T" "$export_symbols"'
fi

```

```

fi

    if ${skipped_export-false}; then
    if test -n "$export_symbols" && test -n "$include_expsyms";
then
    tmp_export_symbols="$export_symbols"
    test -n "$orig_export_symbols" &&
tmp_export_symbols="$orig_export_symbols"
    $opt_dry_run || eval '$ECHO "$include_expsyms" | $SP2NL >>
"$tmp_export_symbols"'
    fi

    if test -n "$orig_export_symbols"; then
    # The given exports_symbols file has to be filtered, so
filter it.
    func_verbose "filter symbol list for `\$libname.la' to tag
DATA exports"
    # FIXME: $output_objdir/$libname.filter potentially
contains lots of
    # 's' commands which not all seds can handle. GNU sed
should be fine
    # though. Also, the filter scales superlinearly with the
number of
    # global variables. join(1) would be nice here, but
unfortunately
    # isn't a blessed tool.
    $opt_dry_run || $SED -e '/[ ,]DATA/!d;s,\(.*\) \([
\,].*\),s|^1$|1\2|,' < $export_symbols >
$output_objdir/$libname.filter
    delfiles+=" $export_symbols $output_objdir/$libname.filter"
    export_symbols=$output_objdir/$libname.def
    $opt_dry_run || $SED -f $output_objdir/$libname.filter <
$orig_export_symbols > $export_symbols
    fi
    fi

    libobjs=$output
    # Restore the value of output.
    output=$save_output

    if test -n "$convenience" && test -n
"$whole_archive_flag_spec"; then
    eval libobjs="\`$libobjs $whole_archive_flag_spec\`"
    test "X$libobjs" = "X " && libobjs=
    fi
    # Expand the library linking commands again to reset the
# value of $libobjs for piecwise linking.

    # Do each of the archive commands.
    if test "$module" = yes && test -n "$module_cmds" ; then
    if test -n "$export_symbols" && test -n
"$module_expsym_cmds"; then

```

```

        cmds=$module_expsym_cmds
    else
        cmds=$module_cmds
    fi
else
    if test -n "$export_symbols" && test -n
"$archive_expsym_cmds"; then
        cmds=$archive_expsym_cmds
    else
        cmds=$archive_cmds
    fi
fi
fi

if test -n "$delfiles"; then
    # Append the command to remove temporary files to $cmds.
    eval cmds="\\"$cmds~\${RM} $delfiles\"
fi

# Add any objects from preloaded convenience libraries
if test -n "$dlprefiles"; then
    gentop="$output_objdir/${outputname}x"
    generated+=" $gentop"

    func_extract_archives $gentop $dlprefiles
    libobjs+=" $func_extract_archives_result"
    test "X$libobjs" = "X " && libobjs=
fi

save_ifs="$IFS"; IFS='~'
for cmd in $cmds; do
    IFS="$save_ifs"
    eval cmd="\\"$cmd\"
    $opt_silent || {
        func_quote_for_expand "$cmd"
        eval "func_echo $func_quote_for_expand_result"
    }
    $opt_dry_run || eval "$cmd" || {
        lt_exit=$?

        # Restore the uninstalled library and exit
        if test "$opt_mode" = relink; then
            ( cd "$output_objdir" && \
              $RM "${realname}T" && \
              $MV "${realname}U" "$realname" )
        fi

        exit $lt_exit
    }
done
IFS="$save_ifs"

```

```

# Restore the uninstalled library and exit
if test "$opt_mode" = relink; then
  $opt_dry_run || eval '(cd $output_objdir && $RM ${realname}T &&
$MV $realname ${realname}T && $MV ${realname}U $realname)' || exit $?

  if test -n "$convenience"; then
    if test -z "$whole_archive_flag_spec"; then
      func_show_eval '${RM}r "$gentop"'
    fi
  fi

  exit $EXIT_SUCCESS
fi

# Create links to the real library.
for linkname in $linknames; do
  if test "$realname" != "$linkname"; then
    func_show_eval '(cd "$output_objdir" && $RM "$linkname" &&
$LN_S "$realname" "$linkname")' 'exit $?'
  fi
done

# If -module or -export-dynamic was specified, set the dlname.
if test "$module" = yes || test "$export_dynamic" = yes; then
  # On all known operating systems, these are identical.
  dlname="$soname"
fi
fi
;;

obj)
  if test -n "$dlfiles$dlprefiles" || test "$dlself" != no; then
    func_warning "\`-dlopen' is ignored for objects"
  fi

  case " $deplibs" in
    *\ -l* | *\ -L*)
    func_warning "\`-l' and \`-L' are ignored for objects" ;;
  esac

  test -n "$rpath" && \
    func_warning "\`-rpath' is ignored for objects"

  test -n "$xrpath" && \
    func_warning "\`-R' is ignored for objects"

  test -n "$vinfo" && \
    func_warning "\`-version-info' is ignored for objects"

  test -n "$release" && \
    func_warning "\`-release' is ignored for objects"

```

```

    case $output in
    *.lo)
    test -n "$objs$old_deplibs" && \
        func_fatal_error "cannot build library object ``$output' from
non-libtool objects"

    libobj=$output
    func_lo2o "$libobj"
    obj=$func_lo2o_result
    ;;
    *)
    libobj=
    obj="$output"
    ;;
    esac

    # Delete the old objects.
    $opt_dry_run || $RM $obj $libobj

    # Objects from convenience libraries.  This assumes
    # single-version convenience libraries.  Whenever we create
    # different ones for PIC/non-PIC, this we'll have to duplicate
    # the extraction.
    reload_conv_objs=
    gentop=
    # reload_cmds runs $LD directly, so let us get rid of
    # -Wl from whole_archive_flag_spec and hope we can get by with
    # turning comma into space..
    wl=

    if test -n "$convenience"; then
    if test -n "$whole_archive_flag_spec"; then
        eval tmp_whole_archive_flags=\`"$whole_archive_flag_spec"\`
        reload_conv_objs=$reload_objs\ `"$ECHO
"$tmp_whole_archive_flags" | $SED 's|,| |g'\`
    else
        gentop="$output_objdir/${obj}x"
        generated+=" $gentop"

        func_extract_archives $gentop $convenience
        reload_conv_objs="$reload_objs $func_extract_archives_result"
    fi
    fi

    # If we're not building shared, we need to use non_pic_objs
    test "$build_libtool_libs" != yes && libobjs="$non_pic_objects"

    # Create the old-style object.
    reload_objs="$objs$old_deplibs "`"$ECHO "$libobjs" | $SP2NL |
$SED "/\.${libext}$/d; /\.lib$/d; $lo2o" | $NL2SP`" $reload_conv_objs"
### testsuite: skip nested quoting test

```

```

output="$obj"
func_execute_cmds "$reload_cmds" 'exit $?'

# Exit if we aren't doing a library object file.
if test -z "$libobj"; then
if test -n "$gentop"; then
  func_show_eval '${RM}r "$gentop"'
fi

exit $EXIT_SUCCESS
fi

if test "$build_libtool_libs" != yes; then
if test -n "$gentop"; then
  func_show_eval '${RM}r "$gentop"'
fi

# Create an invalid libtool object if no PIC, so that we don't
# accidentally link it into a program.
# $show "echo timestamp > $libobj"
# $opt_dry_run || eval "echo timestamp > $libobj" || exit $?
exit $EXIT_SUCCESS
fi

if test -n "$pic_flag" || test "$pic_mode" != default; then
# Only do commands if we really have different PIC objects.
reload_objs="$libobjs $reload_conv_objs"
output="$libobj"
func_execute_cmds "$reload_cmds" 'exit $?'
fi

if test -n "$gentop"; then
func_show_eval '${RM}r "$gentop"'
fi

exit $EXIT_SUCCESS
;;

prog)
  case $host in
  *cygwin*) func_stripname '' '.exe' "$output"
            output=${func_stripname_result}.exe;;
  esac
  test -n "$vinfo" && \
  func_warning "\`-version-info' is ignored for programs"

  test -n "$release" && \
  func_warning "\`-release' is ignored for programs"

  test "$preload" = yes \
    && test "$dlopen_support" = unknown \
    && test "$dlopen_self" = unknown \

```



```
&& test "$dlopen_self_static" = unknown && \  
  func_warning "`LT_INIT([dlopen])' not used. Assuming no dlopen  
support."
```

```
  case $host in  
    *--rhapsody* | *--darwin1.[012])  
      # On Rhapsody replace the C library is the System framework  
      compile_deplibs=`$ECHO " $compile_deplibs" | $SED 's/ -lc /  
System.ltframework /'`  
      finalize_deplibs=`$ECHO " $finalize_deplibs" | $SED 's/ -lc /  
System.ltframework /'`  
      ;;  
    esac
```

```
  case $host in  
    *--darwin*)  
      # Don't allow lazy linking, it breaks C++ global constructors  
      # But is supposedly fixed on 10.4 or later (yay!).  
      if test "$tagname" = CXX ; then  
        case ${MACOSX_DEPLOYMENT_TARGET-10.0} in  
          10.[0123])  
            compile_command+=" ${wl}-bind_at_load"  
            finalize_command+=" ${wl}-bind_at_load"  
            ;;  
          esac  
        fi  
        # Time to change all our "foo.ltframework" stuff back to "-  
framework foo"  
        compile_deplibs=`$ECHO " $compile_deplibs" | $SED 's% \([^  
$]*\)\.ltframework% -framework \1%g'`  
        finalize_deplibs=`$ECHO " $finalize_deplibs" | $SED 's% \([^  
$]*\)\.ltframework% -framework \1%g'`  
        ;;  
      esac
```

```
  # move library search paths that coincide with paths to not yet  
  # installed libraries to the beginning of the library search
```

```
list  
  new_libs=  
  for path in $notinst_path; do  
    case " $new_libs " in  
      *" -L$path/$objdir "*) ;;  
    *)  
      case " $compile_deplibs " in  
        *" -L$path/$objdir "*)  
          new_libs+=" -L$path/$objdir" ;;  
      esac  
      ;;  
    esac  
  done  
  for deplib in $compile_deplibs; do
```

```

case $deplib in
-L*)
  case " $new_libs " in
  *" $deplib "*) ;;
  *) new_libs+=" $deplib" ;;
  esac
  ;;
*) new_libs+=" $deplib" ;;
esac
done
compile_deplibs="$new_libs"

compile_command+=" $compile_deplibs"
finalize_command+=" $finalize_deplibs"

if test -n "$rpath$xrpath"; then
# If the user specified any rpath flags, then add them.
for libdir in $rpath $xrpath; do
  # This is the magic to use -rpath.
  case "$finalize_rpath " in
  *" $libdir "*) ;;
  *) finalize_rpath+=" $libdir" ;;
  esac
done
fi

# Now hardcode the library paths
rpath=
hardcode_libdirs=
for libdir in $compile_rpath $finalize_rpath; do
if test -n "$hardcode_libdir_flag_spec"; then
  func_replace_sysroot "$libdir"
  libdir=$func_replace_sysroot_result
  func_stripname '=' ' ' "$libdir"
  libdir=$func_stripname_result
  if test -n "$hardcode_libdir_separator"; then
    if test -z "$hardcode_libdirs"; then
      hardcode_libdirs="$libdir"
    else
      # Just accumulate the unique libdirs.
      case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
*" $hardcode_libdir_separator$libdir$hardcode_libdir_separator"*)
        ;;
        *)
          hardcode_libdirs+=" $hardcode_libdir_separator$libdir"
          ;;
      esac
    fi
  fi
fi
done

```

```

else
    # We only want to hardcode in an rpath if it isn't in the
    # default dlsearch path.
    func_normal_abspath "$libdir"
    libdir_norm=$func_normal_abspath_result
    case " $sys_lib_dlsearch_path " in
    *" $libdir_norm ") ;;
    *) eval flag=\ "$hardcode_libdir_flag_spec\ "
        rpath+=" $flag"
        ;;
    esac
fi
elif test -n "$runpath_var"; then
    case "$perm_rpath " in
    *" $libdir ") ;;
    *) perm_rpath+=" $libdir" ;;
    esac
fi
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-cegcc*)
    testbindir=`${ECHO} "$libdir" | ${SED} -e 's*/lib$*/bin*'`
    case :$dllsearchpath: in
    *:$libdir:*) ;;
    :) dllsearchpath=$libdir;;
    *) dllsearchpath+=":$libdir";;
    esac
case :$dllsearchpath: in
    *:$testbindir:*) ;;
    :) dllsearchpath=$testbindir;;
    *) dllsearchpath+=":$testbindir";;
    esac
;;
esac
done
# Substitute the hardcoded libdirs into the rpath.
if test -n "$hardcode_libdir_separator" &&
test -n "$hardcode_libdirs"; then
libdir="$hardcode_libdirs"
eval rpath=\ "$hardcode_libdir_flag_spec\ "
fi
compile_rpath="$rpath"

rpath=
hardcode_libdirs=
for libdir in $finalize_rpath; do
if test -n "$hardcode_libdir_flag_spec"; then
    if test -n "$hardcode_libdir_separator"; then
        if test -z "$hardcode_libdirs"; then
            hardcode_libdirs="$libdir"
        else
            # Just accumulate the unique libdirs.

```

```

        case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
*"${hardcode_libdir_separator}$libdir${hardcode_libdir_separator}"*)
    ;;
    *)
        hardcode_libdirs+="${hardcode_libdir_separator}$libdir"
        ;;
    esac
fi
else
    # We only want to hardcode in an rpath if it isn't in the
    # default dlsearch path.
    case " $sys_lib_dlsearch_path " in
    *" $libdir "*) ;;
    *) eval flag="\${hardcode_libdir_flag_spec}"
        rpath+="$flag"
        ;;
    esac
fi
elif test -n "$runpath_var"; then
    case "$finalize_perm_rpath " in
    *" $libdir "*) ;;
    *) finalize_perm_rpath+="$libdir" ;;
    esac
fi
done
# Substitute the hardcoded libdirs into the rpath.
if test -n "$hardcode_libdir_separator" &&
test -n "$hardcode_libdirs"; then
libdir="$hardcode_libdirs"
eval rpath="\${hardcode_libdir_flag_spec}"
fi
finalize_rpath="$rpath"

if test -n "$libobjs" && test "$build_old_libs" = yes; then
# Transform all the library objects into standard objects.
compile_command=`$ECHO "$compile_command" | $SP2NL | $SED "$1o2o"
| $NL2SP`
finalize_command=`$ECHO "$finalize_command" | $SP2NL | $SED
"$1o2o" | $NL2SP`
fi

func_generate_dlsyms "$outputname" "@PROGRAM@" "no"

# template prelinking step
if test -n "$prelink_cmds"; then
func_execute_cmds "$prelink_cmds" 'exit $?'
fi

wrappers_required=yes

```

```

case $host in
*cegcc* | *mingw32ce*)
    # Disable wrappers for cegcc and mingw32ce hosts, we are cross
    compiling anyway.
    wrappers_required=no
    ;;
*cygwin* | *mingw* )
    if test "$build_libtool_libs" != yes; then
        wrappers_required=no
    fi
    ;;
*)
    if test "$need_relink" = no || test "$build_libtool_libs" !=
yes; then
        wrappers_required=no
    fi
    ;;
esac
if test "$wrappers_required" = no; then
# Replace the output file specification.
compile_command=`$ECHO "$compile_command" | $SED
's%@OUTPUT@%' "$output" '%g'`
link_command="$compile_command$compile_rpath"

# We have no uninstalled library dependencies, so finalize right
now.
exit_status=0
func_show_eval "$link_command" 'exit_status=$?'

if test -n "$postlink_cmds"; then
    func_to_tool_file "$output"
    postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%' "$output" '%g' -e
's%@TOOL_OUTPUT@%' "$func_to_tool_file_result" '%g'`
    func_execute_cmds "$postlink_cmds" 'exit $?'
fi

# Delete the generated files.
if test -f "$output_objdir/${outputname}S.${objext}"; then
    func_show_eval '$RM "$output_objdir/${outputname}S.${objext}"'
fi

exit $exit_status
fi

if test -n "$compile_shlibpath$finalize_shlibpath"; then
    compile_command="$shlibpath_var=\"${compile_shlibpath$finalize_shl
ibpath}\$shlibpath_var\" $compile_command"
fi
if test -n "$finalize_shlibpath"; then
    finalize_command="$shlibpath_var=\"${finalize_shlibpath}\$shlibpat
h_var\" $finalize_command"

```

```

fi

compile_var=
finalize_var=
if test -n "$runpath_var"; then
if test -n "$perm_rpath"; then
    # We should set the runpath_var.
    rpath=
    for dir in $perm_rpath; do
        rpath+="$dir:"
    done
    compile_var="$runpath_var=\"\$rpath\$$runpath_var\" "
fi
if test -n "$finalize_perm_rpath"; then
    # We should set the runpath_var.
    rpath=
    for dir in $finalize_perm_rpath; do
        rpath+="$dir:"
    done
    finalize_var="$runpath_var=\"\$rpath\$$runpath_var\" "
fi
fi

if test "$no_install" = yes; then
# We don't need to create a wrapper script.
link_command="$compile_var$compile_command$compile_rpath"
# Replace the output file specification.
link_command=`$ECHO "$link_command" | $SED
's%@OUTPUT@%'"$output"'%g'\`
    # Delete the old output file.
    $opt_dry_run || $RM $output
    # Link the executable and exit
    func_show_eval "$link_command" 'exit $?'

if test -n "$postlink_cmds"; then
    func_to_tool_file "$output"
    postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%'"$output"'%g' -e
's%@TOOL_OUTPUT@%'"$func_to_tool_file_result"'%g'\`
        func_execute_cmds "$postlink_cmds" 'exit $?'
fi

exit $EXIT_SUCCESS
fi

if test "$hardcode_action" = relink; then
# Fast installation is not supported
link_command="$compile_var$compile_command$compile_rpath"
relink_command="$finalize_var$finalize_command$finalize_rpath"

func_warning "this platform does not like uninstalled shared
libraries"

```

```

func_warning "\`$output' will be relinked during installation"
else
if test "$fast_install" != no; then
    link_command="$finalize_var$compile_command$finalize_rpath"
    if test "$fast_install" = yes; then
        relink_command=`$ECHO
"$compile_var$compile_command$compile_rpath" | $SED
's%@OUTPUT@%\$progdir/\$file%g'\`
    else
        # fast_install is set to needless
        relink_command=
    fi
else
    link_command="$compile_var$compile_command$compile_rpath"
    relink_command="$finalize_var$finalize_command$finalize_rpath"
fi
fi

# Replace the output file specification.
link_command=`$ECHO "$link_command" | $SED
's%@OUTPUT@%'"$output_objdir/$outputname"'%g'\`

# Delete the old output files.
$opt_dry_run || $RM $output $output_objdir/$outputname
$output_objdir/lt-$outputname

func_show_eval "$link_command" 'exit $?'

if test -n "$postlink_cmds"; then
func_to_tool_file "$output_objdir/$outputname"
postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%'"$output_objdir/$outputname"'%g' -e
's%@TOOL_OUTPUT@%'"$func_to_tool_file_result"'%g'\`
func_execute_cmds "$postlink_cmds" 'exit $?'
fi

# Now create the wrapper script.
func_verbose "creating $output"

# Quote the relink command for shipping.
if test -n "$relink_command"; then
# Preserve any variables that may affect compiler behavior
for var in $variables_saved_for_relink; do
    if eval test -z \"\${$var+set}\"; then
        relink_command="{ test -z \"\${$var+set}\" || $lt_unset $var
|| { $var=; export $var; }; }; $relink_command"
    elif eval var_value=\${$var}; test -z "$var_value"; then
        relink_command="$var=; export $var; $relink_command"
    else
        func_quote_for_eval "$var_value"
        relink_command="$var=$func_quote_for_eval_result; export
$var; $relink_command"

```

```

    fi
done
relink_command="(cd `pwd`; $relink_command)"
relink_command=`$ECHO "$relink_command" | $SED
"$sed_quote_subst"`
    fi

# Only actually do things if not in dry run mode.
$opt_dry_run || {
# win32 will think the script is a binary if it has
# a .exe suffix, so we strip it off here.
case $output in
*.exe) func_stripname '' '.exe' "$output"
        output=$func_stripname_result ;;
esac
# test for cygwin because mv fails w/o .exe extensions
case $host in
*cygwin*)
    exeext=.exe
    func_stripname '' '.exe' "$outputname"
    outputname=$func_stripname_result ;;
*) exeext= ;;
esac
case $host in
*cygwin* | *mingw* )
    func_dirname_and_basename "$output" "" "."
    output_name=$func_basename_result
    output_path=$func_dirname_result
    cwrappersource="$output_path/$objdir/lt-$output_name.c"
    cwrapper="$output_path/$output_name.exe"
    $RM $cwrappersource $cwrapper
    trap "$RM $cwrappersource $cwrapper; exit $EXIT_FAILURE" 1 2
15

    func_emit_cwrapperexe_src > $cwrappersource

# The wrapper executable is built using the $host compiler,
# because it contains $host paths and files. If cross-
# compiling, it, like the target executable, must be
# executed on the $host or under an emulation environment.
$opt_dry_run || {
    $LTCC $LTCFLAGS -o $cwrapper $cwrappersource
    $STRIP $cwrapper
}

# Now, create the wrapper script for func_source use:
func_ltwrapper_scriptname $cwrapper
$RM $func_ltwrapper_scriptname_result
trap "$RM $func_ltwrapper_scriptname_result; exit
$EXIT_FAILURE" 1 2 15
$opt_dry_run || {
    # note: this script will not be executed, so do not chmod.

```



```

        if test "x$build" = "x$host" ; then
            $cwrapper --lt-dump-script >
$func_ltwrapper_scriptname_result
        else
            func_emit_wrapper no > $func_ltwrapper_scriptname_result
            fi
        }
    ;;
* )
    $RM $output
    trap "$RM $output; exit $EXIT_FAILURE" 1 2 15

    func_emit_wrapper no > $output
    chmod +x $output
    ;;
esac
}
exit $EXIT_SUCCESS
;;
esac

```

```

# See if we need to build an old-fashioned archive.
for oldlib in $oldlibs; do

```

```

    if test "$build_libtool_libs" = convenience; then
        oldobjs="$libobjs_save $symfileobj"
        addlibs="$convenience"
        build_libtool_libs=no
    else
        if test "$build_libtool_libs" = module; then
            oldobjs="$libobjs_save"
            build_libtool_libs=no
        else
            oldobjs="$old_deplibs $non_pic_objects"
            if test "$preload" = yes && test -f "$symfileobj"; then
                oldobjs+="$symfileobj"
            fi
        fi
        addlibs="$old_convenience"
    fi

    if test -n "$addlibs"; then
        gentop="$output_objdir/${outputname}x"
        generated+="$gentop"
    fi

    func_extract_archives $gentop $addlibs
    oldobjs+="$func_extract_archives_result"
    fi

    # Do each command in the archive commands.
    if test -n "$old_archive_from_new_cmds" && test
"$build_libtool_libs" = yes; then

```

```

cmds=$old_archive_from_new_cmds
else

# Add any objects from preloaded convenience libraries
if test -n "$dlprefiles"; then
  gentop="$output_objdir/${outputname}x"
  generated+=" $gentop"

  func_extract_archives $gentop $dlprefiles
  oldobjs+=" $func_extract_archives_result"
fi

# POSIX demands no paths to be encoded in archives. We have
# to avoid creating archives with duplicate basenames if we
# might have to extract them afterwards, e.g., when creating a
# static archive out of a convenience library, or when linking
# the entirety of a libtool archive into another (currently
# not supported by libtool).
if (for obj in $oldobjs
    do
      func_basename "$obj"
      $ECHO "$func_basename_result"
    done | sort | sort -uc >/dev/null 2>&1); then
  :
else
  echo "copying selected object files to avoid basename
conflicts..."
  gentop="$output_objdir/${outputname}x"
  generated+=" $gentop"
  func_mkdir_p "$gentop"
  save_oldobjs=$oldobjs
  oldobjs=
  counter=1
  for obj in $save_oldobjs
  do
    func_basename "$obj"
    objbase="$func_basename_result"
    case " $oldobjs " in
      " ") oldobjs=$obj ;;
      *[\ /]"$objbase "*)
        while ;; do
          # Make sure we don't pick an alternate name that also
          # overlaps.
          newobj=lt$counter-$objbase
          func_arith $counter + 1
          counter=$func_arith_result
          case " $oldobjs " in
            *[\ /]"$newobj "*) ;;
            *) if test ! -f "$gentop/$newobj"; then break; fi ;;
          esac
        done

```

```

        func_show_eval "ln $obj $gentop/$newobj || cp $obj
$gentop/$newobj"
        oldobjs+=" $gentop/$newobj"
        ;;
        *) oldobjs+=" $obj" ;;
    esac
done
fi
func_to_tool_file "$oldlib" func_convert_file_msys_to_w32
tool_oldlib=$func_to_tool_file_result
eval cmds="\$old_archive_cmds\"

func_len " $cmds"
len=$func_len_result
if test "$len" -lt "$max_cmd_len" || test "$max_cmd_len" -le -1;
then
    cmds=$old_archive_cmds
elif test -n "$archiver_list_spec"; then
    func_verbose "using command file archive linking..."
    for obj in $oldobjs
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result"
done > $output_objdir/$libname.libcmd
        func_to_tool_file "$output_objdir/$libname.libcmd"
        oldobjs=" $archiver_list_spec$func_to_tool_file_result"
        cmds=$old_archive_cmds
    else
        # the command line is too long to link in one step, link in
parts
        func_verbose "using piecewise archive linking..."
        save_RANLIB=$RANLIB
        RANLIB=:
        objlist=
        concat_cmds=
        save_oldobjs=$oldobjs
        oldobjs=
        # Is there a better way of finding the last object in the list?
        for obj in $save_oldobjs
        do
            last_oldobj=$obj
        done
        eval test_cmds="\$old_archive_cmds\"
        func_len " $test_cmds"
        len0=$func_len_result
        len=$len0
        for obj in $save_oldobjs
        do
            func_len " $obj"
            func_arith $len + $func_len_result
            len=$func_arith_result
            objlist+=" $obj"

```

```

if test "$len" -lt "$max_cmd_len"; then
:
else
# the above command should be used before it gets too long
oldobjs=$objlist
if test "$obj" = "$last_oldobj" ; then
RANLIB=$save_RANLIB
fi
test -z "$concat_cmds" || concat_cmds=$concat_cmds~
eval concat_cmds="\${concat_cmds}$old_archive_cmds\"
objlist=
len=$len0
fi
done
RANLIB=$save_RANLIB
oldobjs=$objlist
if test "X$oldobjs" = "X" ; then
eval cmds="\${concat_cmds}\"
else
eval cmds="\${concat_cmds}~\${old_archive_cmds}\"
fi
fi
fi
func_execute_cmds "$cmds" 'exit $?'
```

```
done

test -n "$generated" && \
func_show_eval "${RM}r$generated"
```

```
# Now create the libtool archive.
case $output in
*.la)
old_library=
test "$build_old_libs" = yes && old_library="$libname.$libext"
func_verbose "creating $output"

# Preserve any variables that may affect compiler behavior
for var in $variables_saved_for_relink; do
if eval test -z "\${$var+set}"; then
relink_command="{ test -z "\${$var+set}" || $lt_unset $var ||
{ $var=; export $var; }; }; $relink_command"
elif eval var_value=\${$var}; test -z "$var_value"; then
relink_command="$var=; export $var; $relink_command"
else
func_quote_for_eval "$var_value"
relink_command="$var=${func_quote_for_eval_result}; export $var;
$relink_command"
fi
done
# Quote the link command for shipping.
relink_command="(cd `pwd`; $SHELL $progpash $preserve_args --
mode=relink $libtool_args @inst_prefix_dir)"
```

```

        relink_command=`$ECHO "$relink_command" | $SED
"$sed_quote_subst"`
        if test "$hardcode_automatic" = yes ; then
            relink_command=
        fi

        # Only create the output if not a dry run.
        $opt_dry_run || {
            for installed in no yes; do
                if test "$installed" = yes; then
                    if test -z "$install_libdir"; then
                        break
                    fi
                    output="$output_objdir/$outputname"i
                    # Replace all uninstalled libtool libraries with the
installed ones
                    newdependency_libs=
                    for deplib in $dependency_libs; do
                        case $deplib in
                            *.la)
                                func_basename "$deplib"
                                name="$func_basename_result"
                                func_resolve_sysroot "$deplib"
                                eval libdir=`${SED} -n -e 's/^libdir=\.*)$/\1/p'
$func_resolve_sysroot_result`
                                test -z "$libdir" && \
                                    func_fatal_error "`$deplib' is not a valid libtool
archive"
                                newdependency_libs+=" ${lt_sysroot:+}$libdir/$name"
                                ;;
                            -L*)
                                func_stripname -L '' "$deplib"
                                func_replace_sysroot "$func_stripname_result"
                                newdependency_libs+=" -L$func_replace_sysroot_result"
                                ;;
                            -R*)
                                func_stripname -R '' "$deplib"
                                func_replace_sysroot "$func_stripname_result"
                                newdependency_libs+=" -R$func_replace_sysroot_result"
                                ;;
                            *) newdependency_libs+=" $deplib" ;;
                        esac
                    done
                    dependency_libs="$newdependency_libs"
                    newdlfiles=

                    for lib in $dlfiles; do
                        case $lib in
                            *.la)
                                func_basename "$lib"
                                name="$func_basename_result"
                                eval libdir=`${SED} -n -e 's/^libdir=\.*)$/\1/p' $lib`

```

```

test -z "$libdir" && \
    func_fatal_error "\`$lib' is not a valid libtool archive"
newdlfiles+=" ${lt_sysroot:+}$libdir/$name"
;;
*) newdlfiles+=" $lib" ;;
esac
done
dlfiles="$newdlfiles"
newdlprefiles=
for lib in $dlprefiles; do
    case $lib in
        *.la)
            # Only pass preopened files to the pseudo-archive (for
            # eventual linking with the app. that links it) if we
            # didn't already link the preopened objects directly into
            # the library:
            func_basename "$lib"
            name="$func_basename_result"
            eval libdir=`${SED} -n -e 's/^libdir=\.(\.*)$/\1/p' $lib`
            test -z "$libdir" && \
                func_fatal_error "\`$lib' is not a valid libtool archive"
            newdlprefiles+=" ${lt_sysroot:+}$libdir/$name"
            ;;
        esac
    done
    dlprefiles="$newdlprefiles"
else
    newdlfiles=
    for lib in $dlfiles; do
        case $lib in
            [\\/] * | [A-Za-z]:[\\/] *) abs="$lib" ;;
            *) abs=`pwd`"/$lib" ;;
        esac
        newdlfiles+=" $abs"
    done
    dlfiles="$newdlfiles"
    newdlprefiles=
    for lib in $dlprefiles; do
        case $lib in
            [\\/] * | [A-Za-z]:[\\/] *) abs="$lib" ;;
            *) abs=`pwd`"/$lib" ;;
        esac
        newdlprefiles+=" $abs"
    done
    dlprefiles="$newdlprefiles"
fi
$RM $output
# place dlname in correct position for cygwin
# In fact, it would be nice if we could use this code for all
target
# systems that can't hard-code library paths into their
executables

```

```

        # and that have no shared library path variable independent of
PATH,
        # but it turns out we can't easily determine that from
inspecting
        # libtool variables, so we have to hard-code the OSs to which
it
        # applies here; at the moment, that means platforms that use
the PE
        # object format with DLL files.  See the long comment at the
top of
        # tests/bindir.at for full details.
tdlname=$dlname
case $host,$output,$installed,$module,$dlname in
    *cygwin*,*lai,yes,no,*.dll | *mingw*,*lai,yes,no,*.dll |
*cegcc*,*lai,yes,no,*.dll)
        # If a -bindir argument was supplied, place the dll there.
        if test "x$bindir" != x ;
        then
            func_relative_path "$install_libdir" "$bindir"
            tdlname=$func_relative_path_result$dlname
        else
            # Otherwise fall back on heuristic.
            tdlname=../bin/$dlname
        fi
        ;;
    esac
    $ECHO > $output "\
# $outputname - a libtool library file
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION
#
# Please DO NOT delete this file!
# It is necessary for linking the library.

# The name that we can dlopen(3).
dlname='$tdlname'

# Names of this library.
library_names='$library_names'

# The name of the static archive.
old_library='$old_library'

# Linker flags that can not go in dependency_libs.
inherited_linker_flags='$new_inherited_linker_flags'

# Libraries that this one depends upon.
dependency_libs='$dependency_libs'

# Names of additional weak libraries provided by this library
weak_library_names='$weak_libs'

# Version information for $libname.

```

```

current=$current
age=$age
revision=$revision

# Is this an already installed library?
installed=$installed

# Should we warn about portability when linking against -modules?
shouldnotlink=$module

# Files to dlopen/dlpreopen
dlopen='$dlfiles'
dlpreopen='$dlprefiles'

# Directory that this library needs to be installed in:
libdir='$install_libdir'
    if test "$installed" = no && test "$need_relink" = yes; then
        $ECHO >> $output "\
relink_command=\"\${relink_command}\""
    fi
done
}

# Do a symbolic link so that the libtool archive can be found in
# LD_LIBRARY_PATH before the program is installed.
func_show_eval '( cd "$output_objdir" && $RM "$outputname" &&
$LN_S "../$outputname" "$outputname" )' 'exit $?'
;;
esac
exit $EXIT_SUCCESS
}

{ test "$opt_mode" = link || test "$opt_mode" = relink; } &&
    func_mode_link ${1+"$@"}

# func_mode_uninstall arg...
func_mode_uninstall ()
{
    $opt_debug
    RM="$nonopt"
    files=
    rmforce=
    exit_status=0

    # This variable tells wrapper scripts just to set variables rather
    # than running their programs.
    libtool_install_magic="$magic"

    for arg
    do
        case $arg in

```



```

    -f) RM+=" $arg"; rmforce=yes ;;
    -*) RM+=" $arg" ;;
    *) files+=" $arg" ;;
  esac
done

test -z "$RM" && \
  func_fatal_help "you must specify an RM program"

rmdirs=

for file in $files; do
  func_dirname "$file" "" "."
  dir="$func_dirname_result"
  if test "X$dir" = X.; then
    odir="$objdir"
  else
    odir="$dir/$objdir"
  fi
  func_basename "$file"
  name="$func_basename_result"
  test "$opt_mode" = uninstall && odir="$dir"

  # Remember odir for removal later, being careful to avoid
duplicates
  if test "$opt_mode" = clean; then
    case " $rmdirs " in
      *" $odir *) ;;
      *) rmdirs+=" $odir" ;;
    esac
  fi

  # Don't error if the file doesn't exist and rm -f was used.
  if { test -L "$file"; } >/dev/null 2>&1 ||
    { test -h "$file"; } >/dev/null 2>&1 ||
    test -f "$file"; then
    :
  elif test -d "$file"; then
    exit_status=1
    continue
  elif test "$rmforce" = yes; then
    continue
  fi

  rmfiles="$file"

  case $name in
    *.la)
      # Possibly a libtool archive, so verify it.
      if func_lalib_p "$file"; then
        func_source $dir/$name

```

```

# Delete the libtool libraries and symlinks.
for n in $library_names; do
  rmfiles+=" $odir/$n"
done
test -n "$old_library" && rmfiles+=" $odir/$old_library"

case "$opt_mode" in
clean)
  case " $library_names " in
*" $dlname ") ;;
*) test -n "$dlname" && rmfiles+=" $odir/$dlname" ;;
esac
test -n "$libdir" && rmfiles+=" $odir/$name $odir/${name}i"
;;
uninstall)
  if test -n "$library_names"; then
    # Do each command in the postuninstall commands.
    func_execute_cmds "$postuninstall_cmds" 'test "$rmforce" =
yes || exit_status=1'
    fi

    if test -n "$old_library"; then
      # Do each command in the old_postuninstall commands.
      func_execute_cmds "$old_postuninstall_cmds" 'test
"$rmforce" = yes || exit_status=1'
      fi
    # FIXME: should reinstall the best remaining shared library.
    ;;
  esac
fi
;;

*.lo)
# Possibly a libtool object, so verify it.
if func_lalib_p "$file"; then

# Read the .lo file
func_source $dir/$name

# Add PIC object to the list of files to remove.
if test -n "$pic_object" &&
test "$pic_object" != none; then
  rmfiles+=" $dir/$pic_object"
fi

# Add non-PIC object to the list of files to remove.
if test -n "$non_pic_object" &&
test "$non_pic_object" != none; then
  rmfiles+=" $dir/$non_pic_object"
fi
fi
;;

```

```

*)
if test "$opt_mode" = clean ; then
  noexename=$name
  case $file in
    *.exe)
      func_stripname ' ' '.exe' "$file"
      file=$func_stripname_result
      func_stripname ' ' '.exe' "$name"
      noexename=$func_stripname_result
      # $file with .exe has already been added to rmfiles,
      # add $file without .exe
      rmfiles+="$ $file"
      ;;
  esac
  # Do a test to see if this is a libtool program.
  if func_ltwrapper_p "$file"; then
    if func_ltwrapper_executable_p "$file"; then
      func_ltwrapper_scriptname "$file"
      relink_command=
      func_source $func_ltwrapper_scriptname_result
      rmfiles+="$ $func_ltwrapper_scriptname_result"
    else
      relink_command=
      func_source $dir/$noexename
    fi

    # note $name still contains .exe if it was in $file
    originally
    # as does the version of $file that was added into $rmfiles
    rmfiles+="$ $odir/$name $odir/${name}S.${objext}"
    if test "$fast_install" = yes && test -n "$relink_command";
  then
    rmfiles+="$ $odir/lt-$name"
  fi
  if test "X$noexename" != "X$name" ; then
    rmfiles+="$ $odir/lt-${noexename}.c"
  fi
  fi
  ;;
esac
func_show_eval "$RM $rmfiles" 'exit_status=1'
done

# Try to remove the ${objdir}s in the directories where we deleted
files
for dir in $rmdirs; do
  if test -d "$dir"; then
    func_show_eval "rmdir $dir >/dev/null 2>&1"
  fi
done

```

```

    exit $exit_status
}

{ test "$opt_mode" = uninstall || test "$opt_mode" = clean; } &&
    func_mode_uninstall ${1+"$@"}

test -z "$opt_mode" && {
    help="$generic_help"
    func_fatal_help "you must specify a MODE"
}

test -z "$exec_cmd" && \
    func_fatal_help "invalid operation mode \`${opt_mode}'"

if test -n "$exec_cmd"; then
    eval exec "$exec_cmd"
    exit $EXIT_FAILURE
fi

exit $exit_status

# The TAGs below are defined such that we never get into a situation
# in which we disable both kinds of libraries.  Given conflicting
# choices, we go for a static library, that is the most portable,
# since we can't tell whether shared libraries were disabled because
# the user asked for that or because the platform doesn't support
# them.  This is particularly important on AIX, because we don't
# support having both static and shared libraries enabled at the same
# time on that platform, so we default to a shared-only configuration.
# If a disable-shared tag is given, we'll fallback to a static-only
# configuration.  But we'll never go from static-only to shared-only.

# ### BEGIN LIBTOOL TAG CONFIG: disable-shared
build_libtool_libs=no
build_old_libs=yes
# ### END LIBTOOL TAG CONFIG: disable-shared

# ### BEGIN LIBTOOL TAG CONFIG: disable-static
build_old_libs=`case $build_libtool_libs in yes) echo no;; *) echo
yes;; esac`
# ### END LIBTOOL TAG CONFIG: disable-static

# Local Variables:
# mode:shell-script
# sh-indentation:2
# End:
# vi:sw=2

```

File = arm-oe-linux-gnueabi-libtool.~1~

#!/bin/sh

arm-oe-linux-gnueabi-libtool - Provide generalized library-building support services.

Generated automatically by config.status (dbus) 1.6.8

Libtool was configured on host pc45:

NOTE: Changes made to this file will be lost: look at ltmain.sh.

#

Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004, 2005,

2006, 2007, 2008, 2009, 2010, 2011 Free Software Foundation, Inc.

Written by Gordon Matzigkeit, 1996

#

This file is part of GNU Libtool.

#

GNU Libtool is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

#

As a special exception to the GNU General Public License, if you distribute this file as part of a program or library that is built using GNU Libtool, you may include this file under the same distribution terms that you use for the rest of that program.

#

GNU Libtool is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

#

You should have received a copy of the GNU General Public License along with GNU Libtool; see the file COPYING. If not, a copy can be downloaded from <http://www.gnu.org/licenses/gpl.html>, or obtained by writing to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

The names of the tagged configurations supported by this script.
available_tags="CXX RC "

BEGIN LIBTOOL CONFIG

Which release of libtool.m4 was used?

macro_version=2.4.2

macro_revision=1.3337

Whether or not to build shared libraries.

build_libtool_libs=yes

```
# Whether or not to build static libraries.
build_old_libs=yes

# What type of objects to build.
pic_mode=default

# Whether or not to optimize for fast installation.
fast_install=yes

# Shell to use when invoking shell scripts.
SHELL="/bin/sh"

# An echo program that protects backslashes.
ECHO="printf %s\\n"

# The PATH separator for the build system.
PATH_SEPARATOR=":"

# The host system.
host_alias=arm-oe-linux-gnueabi
host=arm-oe-linux-gnueabi
host_os=linux-gnueabi

# The build system.
build_alias=i686-linux
build=i686-pc-linux-gnu
build_os=linux-gnu

# A sed program that does not truncate output.
SED="/bin/sed"

# Sed that helps us avoid accidentally triggering echo(1) options like
-n.
Xsed="$SED -e 1s/^X/"

# A grep program that handles long lines.
GREP="/bin/grep"

# An ERE matcher.
EGREP="/bin/grep -E"

# A literal string matcher.
FGREP="/bin/grep -F"

# A BSD- or MS-compatible name lister.
NM="arm-linux-gnueabi-hf-nm"

# Whether we need soft or hard links.
LN_S="ln -s"

# What is the maximum length of a command?
```

```
max_cmd_len=1572864

# Object file suffix (normally "o").
objext=o

# Executable file suffix (normally "").
exeext=

# whether the shell understands "unset".
lt_unset=unset

# turn spaces into newlines.
SP2NL="tr \\040 \\012"

# turn newlines into spaces.
NL2SP="tr \\015\\012 \\040\\040"

# convert $build file names to $host format.
to_host_file_cmd=func_convert_file_noop

# convert $build files to toolchain format.
to_tool_file_cmd=func_convert_file_noop

# An object symbol dumper.
OBJDUMP="arm-linux-gnueabi-hf-objdump"

# Method to check whether dependent libraries are shared objects.
deplibs_check_method="pass_all"

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd="\$MAGIC_CMD"

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=""

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob="no"

# DLL creation program.
DLLTOOL="false"

# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd="printf %s\\n"

# The archiver.
AR="arm-linux-gnueabi-hf-ar"

# Flags to create an archive.
AR_FLAGS="cru"
```

```

# How to feed a file listing to the archiver.
archiver_list_spec="@@"

# A symbol stripping program.
STRIP="arm-linux-gnueabi-hf-strip"

# Commands used to install an old-style archive.
RANLIB="arm-linux-gnueabi-hf-ranlib"
old_postinstall_cmds="chmod 644 \${oldlib}~\${RANLIB} \${tool_oldlib}"
old_postuninstall_cmds=""

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=no

# A C compiler.
LTCC="arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork
-mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# LTCC compiler flags.
LTCFLAGS=" -Wall -Wextra -Wchar-subscripts -Wmissing-declarations -
Wmissing-prototypes -Wnested-externs -Wpointer-arith -Wcast-align -
Wno-address -Wfloat-equal -Wdeclaration-after-statement -Wno-unused-
label -Wno-missing-field-initializers -Wno-unused-parameter -Wno-sign-
compare -Wno-pointer-sign -Wno-type-limits -fno-common -fno-strict-
aliasing -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types"

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe="sed -n -e 's/^[^*]
*\([ABCDGIRSTW][ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/\1 \2 \2/p' | sed '/ __gnu_lto/d'"

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl="sed -n -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/extern int \1();/p' -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/extern char \1;/p'"

# Transform the output of nm in a C name address pair.
global_symbol_to_c_name_address="sed -n -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/ {\1\1, (void *) 0},/p' -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/ {\1\2, (void *) \&\2},/p'"

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix="sed -n -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/ {\1\1, (void *) 0},/p' -e 's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/ {\1\2, (void *) \&\2},/p' -e
's/^[^*]
*\([ABCDGIRSTW]*\) \([_A-Za-z][_A-Za-z0-9]*\)
\$/ {\1lib\2, (void *)
\&\2},/p'"

```



```
# Specify filename containing input files for $NM.
nm_file_list_spec="@#"

# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-vp1

# The name of the directory that contains temporary libtool files.
objdir=.libs

# Used to examine libraries when file_magic_cmd begins with "file".
MAGIC_CMD=file

# Must we lock files when doing compilation?
need_locks="no"

# Manifest tool.
MANIFEST_TOOL=":"

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
DSYMUTIL=""

# Tool to change global to local symbols on Mac OS X.
NMEDIT=""

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=""

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=""

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=""

# Old archive suffix (normally "a").
libext=a

# Shared library suffix (normally ".so").
shrext_cmds=".so"

# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=""

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink="PATH LD_LIBRARY_PATH LD_RUN_PATH
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH"
```

```
# Do we need the "lib" prefix for modules?
need_lib_prefix=no

# Do we need a version for libraries?
need_version=no

# Library versioning type.
version_type=linux

# Shared library runtime path variable.
runpath_var=LD_RUN_PATH

# Shared library path variable.
shlibpath_var=LD_LIBRARY_PATH

# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=no

# Format of library name prefix.
libname_spec="lib$name"

# List of archive names. First name is the real one, the rest are
links.
# The last name is the one that the linker finds with -lNAME
library_names_spec="\${libname}\${release}\${shared_ext}\${versuffix}
\${libname}\${release}\${shared_ext}\$major \${libname}\${shared_ext}"

# The coded name of the library, if different from the real name.
soname_spec="\${libname}\${release}\${shared_ext}\$major"

# Permission mode override for installation of shared libraries.
install_override_mode=""

# Command to use after installation of a shared archive.
postinstall_cmds=""

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=""

# Commands used to finish a libtool library installation in a
directory.
finish_cmds="PATH=\\\"\\\"$PATH:/sbin\\\" ldconfig -n \${libdir}"

# As "finish_cmds", except a single script fragment to be evaled but
# not shown.
finish_eval=""

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=yes

# Compile-time system search path for libraries.
```

```
sys_lib_search_path_spec="/home/gangadhar/gcc-linaro-arm-linux-
gnueabihf-4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf/4.7.3
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/lib/gcc /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib "
```

```
# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec="/usr/lib /lib"
```

```
# Whether dlopen is supported.
dlopen_support=unknown
```

```
# Whether dlopen of programs is supported.
dlopen_self=unknown
```

```
# Whether dlopen of statically linked programs is supported.
dlopen_self_static=unknown
```

```
# Commands to strip libraries.
old_striplib="arm-linux-gnueabihf-strip --strip-debug"
striplib="arm-linux-gnueabihf-strip --strip-unneeded"
```

```
# The linker used to build libraries.
LD="arm-linux-gnueabihf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"
```

```
# How to create reloadable object files.
reload_flag=" -r"
reload_cmds="\$LD\$reload_flag -o \$output\$reload_objs"
```

```
# Commands used to build an old-style archive.
old_archive_cmds="\$AR \$AR_FLAGS \$oldlib\$oldobjs~\$RANLIB
\$stool_oldlib"
```

```
# A language specific compiler.
CC="arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"
```

```
# Is the compiler the GNU compiler?
with_gcc=yes
```

```
# Compiler flag to turn off builtin functions.
```

```

no_builtin_flag=" -fno-builtin"

# Additional compiler flags for building library objects.
pic_flag=" -fPIC -DPIC"

# How to pass a linker flag through the compiler.
wl="-Wl,"

# Compiler flag to prevent dynamic linking.
link_static_flag="-static"

# Does compiler simultaneously support -c and -o options?
compiler_c_o="yes"

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=no

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=no

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec="\${wl}--export-dynamic"

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec="\${wl}--whole-archive\${convenience} \${wl}--no-whole-archive"

# Whether the compiler copes with passing no objects directly.
compiler_needs_object="no"

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=""

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=""

# Commands used to build a shared archive.
archive_cmds="\${CC} -shared \${pic_flag} \${libobjs} \${deplibs}
\${compiler_flags} \${wl}-soname \${wl}\${soname} -o \${lib}"
archive_expsym_cmds="echo \{\} { global:\{\} >
\${output_objdir}/\${libname}.ver~
cat \${export_symbols} | sed -e \{\}s/\{\}(.*\{\})/\{\}1;/\{\}"
>> \${output_objdir}/\${libname}.ver~
echo \{\} "local: *; }; \{\}" >> \${output_objdir}/\${libname}.ver~
\${CC} -shared \${pic_flag} \${libobjs} \${deplibs} \${compiler_flags}
\${wl}-soname \${wl}\${soname} \${wl}-version-script
\${wl}\${output_objdir}/\${libname}.ver -o \${lib}"

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=""

```

```
module_expsym_cmds=""

# Whether we are building with GNU ld or not.
with_gnu_ld="yes"

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=""

# Flag that enforces no undefined symbols.
no_undefined_flag=""

# Flag to hardcode $libdir into a binary during linking.
# This must work even if $libdir does not exist
hardcode_libdir_flag_spec="\${wl}-rpath \${wl}\$libdir"

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=""

# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=no

# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting ${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=no

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=no

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=unsupported

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=no

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=no
```

```
# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=unknown

# Set to "yes" if exported symbols are required.
always_export_symbols=no

# The commands to list exported symbols.
export_symbols_cmds="\$NM \$libobjs \$convenience |
\$global_symbol_pipe | \$SED 's/.*/ /' | sort | uniq >
\$export_symbols"

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms="_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*"

# Symbols that must always be exported.
include_expsyms=""

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=""

# Commands necessary for finishing linking programs.
postlink_cmds=""

# Specify filename containing input files.
file_list_spec=""

# How to hardcode a shared library path into an executable.
hardcode_action=immediate

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=""

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=""
postdep_objects=""
predeps=""
postdeps=""

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=""

# ### END LIBTOOL CONFIG

# libtool (GNU libtool) 2.4.2
# Written by Gordon Matzigkeit <gord@gnu.ai.mit.edu>, 1996
```

```

# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004, 2005,
2006,
# 2007, 2008, 2009, 2010, 2011 Free Software Foundation, Inc.
# This is free software; see the source for copying conditions. There
is NO
# warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.

# GNU Libtool is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html,
# or obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

# Usage: $progname [OPTION]... [MODE-ARG]...
#
# Provide generalized library-building support services.
#
#      --config                show all configuration variables
#      --debug                enable verbose shell tracing
#  -n, --dry-run              display commands without modifying any
files
#      --features              display basic configuration information
and exit
#      --mode=MODE            use operation mode MODE
#      --preserve-dup-deps    don't remove duplicate dependency
libraries
#      --quiet, --silent      don't print informational messages
#      --no-quiet, --no-silent
#                              print informational messages (default)
#      --no-warn              don't display warning messages
#      --tag=TAG              use configuration variables from tag TAG
#  -v, --verbose              print more informational messages than
default
#      --no-verbose           don't print the extra informational
messages
#      --version              print version information

```

```

# -h, --help, --help-all  print short, long, or detailed help
message
#
# MODE must be one of the following:
#
#       clean          remove files from the build directory
#       compile        compile a source file into a libtool
object
#       execute        automatically set library path, then run
a program
#       finish         complete the installation of libtool
libraries
#       install        install libraries or executables
#       link           create a library or an executable
#       uninstall      remove libraries from an installed
directory
#
# MODE-ARGS vary depending on the MODE.  When passed as first option,
# `--mode=MODE' may be abbreviated as `MODE' or a unique abbreviation
of that.
# Try `$progname --help --mode=MODE' for a more detailed description
of MODE.
#
# When reporting a bug, please describe a test case to reproduce it
and
# include the following information:
#
#       host-triplet:  $host
#       shell:         $SHELL
#       compiler:      $LTCC
#       compiler flags:  $LTCFLAGS
#       linker:        $LD (gnu? $with_gnu_ld)
#       $progname:    (GNU libtool) 2.4.2
#       automake:     $automake_version
#       autoconf:     $autoconf_version
#
# Report bugs to <bug-libtool@gnu.org>.
# GNU libtool home page: <http://www.gnu.org/software/libtool/>.
# General help using GNU software: <http://www.gnu.org/gethelp/>.

PROGRAM=libtool
PACKAGE=libtool
VERSION=2.4.2
TIMESTAMP=""
package_revision=1.3337

# Be Bourne compatible
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
  emulate sh
  NULLCMD=:
  # Zsh 3.x and 4.x performs word splitting on ${1+"$@"}, which
  # is contrary to our usage.  Disable this feature.

```



```

    alias -g '${1+"$@"}'='"$@"'
    setopt NO_GLOB_SUBST
else
    case `(set -o) 2>/dev/null` in *posix*) set -o posix;; esac
fi
BIN_SH=xpg4; export BIN_SH # for Tru64
DUALCASE=1; export DUALCASE # for MKS sh

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
    eval 'cat <<_LTECHO_EOF'
$1
_LTECHO_EOF'
}

# NLS nuisances: We save the old values to restore during execute
mode.
lt_user_locale=
lt_safe_locale=
for lt_var in LANG LANGUAGE LC_ALL LC_CTYPE LC_COLLATE LC_MESSAGES
do
    eval "if test \"\${$lt_var+set}\" = set; then
        save_$lt_var=\${$lt_var}
        $lt_var=C
        export $lt_var
        lt_user_locale=\"\$lt_var=\\\$save_`$lt_var`; \${lt_user_locale}\"
        lt_safe_locale=\"\$lt_var=C; \${lt_safe_locale}\"
    fi"
done
LC_ALL=C
LANGUAGE=C
export LANGUAGE LC_ALL

$lt_unset CDPATH

# Work around backward compatibility issue on IRIX 6.5. On IRIX 6.4+,
sh
# is ksh but when the shell is invoked as "sh" and the current value
of
# the _XPG environment variable is not equal to 1 (one), the special
# positional parameter $0, within a function call, is the name of the
# function.
progpam="$0"

: ${CP="cp -f"}
test "${ECHO+set}" = set || ECHO=${as_echo-'printf %s\n'}
: ${EGREP="egrep"}
: ${FGREP="fgrep"}

```

```

: ${GREP="grep"}
: ${LN_S="ln -s"}
: ${MAKE="make"}
: ${MKDIR="mkdir"}
: ${MV="mv -f"}
: ${RM="rm -f"}
: ${SED="sed"}
: ${SHELL="${CONFIG_SHELL-}/bin/sh"}
: ${Xsed="$SED -e 1s/^X//"}

# Global variables:
EXIT_SUCCESS=0
EXIT_FAILURE=1
EXIT_MISMATCH=63 # $? = 63 is used to indicate version mismatch to
missing.
EXIT_SKIP=77 # $? = 77 is used to indicate a skipped test to
automake.

exit_status=$EXIT_SUCCESS

# Make sure IFS has a sensible default
lt_nl='
'
IFS=" $lt_nl"

dirname="s,/[^/]*$,,"
basename="s,^.*/,,"

# func_dirname file append nondir_replacement
# Compute the dirname of FILE. If nonempty, add APPEND to the result,
# otherwise set result to NONDIR_REPLACEMENT.
func_dirname ()
{
    case ${1} in
        /*) func_dirname_result="${1%/*}${2}" ;;
        * ) func_dirname_result="${3}" ;;
    esac
} # Extended-shell func_dirname implementation

# func_basename file
func_basename ()
{
    func_basename_result="${1##*/}"
} # Extended-shell func_basename implementation

# func_dirname_and_basename file append nondir_replacement
# perform func_basename and func_dirname in a single function
# call:
#   dirname: Compute the dirname of FILE. If nonempty,
#             add APPEND to the result, otherwise set result

```

```

#           to NONDIR_REPLACEMENT.
#           value returned in "$func_dirname_result"
#   basename: Compute filename of FILE.
#           value returned in "$func_basename_result"
# Implementation must be kept synchronized with func_dirname
# and func_basename. For efficiency, we do not delegate to
# those functions but instead duplicate the functionality here.
func_dirname_and_basename ()
{
    case ${1} in
        */*) func_dirname_result="${1%/*}${2}" ;;
        *  ) func_dirname_result="${3}" ;;
    esac
    func_basename_result="${1##*/}"
} # Extended-shell func_dirname_and_basename implementation

# func_stripname prefix suffix name
# strip PREFIX and SUFFIX off of NAME.
# PREFIX and SUFFIX must not contain globbing or regex special
# characters, hashes, percent signs, but SUFFIX may contain a leading
# dot (in which case that matches only a dot).
# func_stripname prefix name
func_stripname ()
{
    # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are
    # positional parameters, so assign one to ordinary parameter
    first.
    func_stripname_result=${3}
    func_stripname_result=${func_stripname_result#"${1}" }
    func_stripname_result=${func_stripname_result%"${2}" }
} # Extended-shell func_stripname implementation

# These SED scripts presuppose an absolute path with a trailing slash.
pathcar='s,^\([^/]*\) .*$, \1, '
pathcdr='s,^[^/]*,, '
removedotparts=':dotsl
            s@/\./@/g
            t dotsl
            s,/\.$/,, '
collapseslashes='s@/\{1,\}@/g'
finalslash='s,/*$,/, '

# func_normal_abspath PATH
# Remove doubled-up and trailing slashes, "." path components,
# and cancel out any ".." path components in PATH after making
# it an absolute path.
#           value returned in "$func_normal_abspath_result"
func_normal_abspath ()
{
    # Start from root dir and reassemble the path.

```

```

func_normal_abspath_result=
func_normal_abspath_tpath=$1
func_normal_abspath_altnamespace=
case $func_normal_abspath_tpath in
  "")
    # Empty path, that just means $cwd.
    func_stripname '' '/' "`pwd`"
    func_normal_abspath_result=$func_stripname_result
    return
  ;;
  # The next three entries are used to spot a run of precisely
  # two leading slashes without using negated character classes;
  # we take advantage of case's first-match behaviour.
  //(*)
    # Unusual form of absolute path, do nothing.
  ;;
  //(*)
    # Not necessarily an ordinary path; POSIX reserves leading '//'
    # and for example Cygwin uses it to access remote file shares
    # over CIFS/SMB, so we conserve a leading double slash if found.
    func_normal_abspath_altnamespace=/
  ;;
  /*)
    # Absolute path, do nothing.
  ;;
  *)
    # Relative path, prepend $cwd.
    func_normal_abspath_tpath=`pwd`/$func_normal_abspath_tpath
  ;;
esac
# Cancel out all the simple stuff to save iterations. We also want
# the path to end with a slash for ease of parsing, so make sure
# there is one (and only one) here.
func_normal_abspath_tpath=`$ECHO "$func_normal_abspath_tpath" | $SED
\
    -e "$removedotparts" -e "$collapseslashes" -e "$finalslash"`
while :; do
  # Processed it all yet?
  if test "$func_normal_abspath_tpath" = / ; then
    # If we ascended to the root using ".." the result may be empty
now.
    if test -z "$func_normal_abspath_result" ; then
      func_normal_abspath_result=/
    fi
    break
  fi
  func_normal_abspath_tcomponent=`$ECHO "$func_normal_abspath_tpath"
| $SED \
    -e "$pathcar"`
  func_normal_abspath_tpath=`$ECHO "$func_normal_abspath_tpath" |
$SED \
    -e "$pathcdr"`

```

```

# Figure out what to do with it
case $func_normal_abspath_tcomponent in
    "")
        # Trailing empty path component, ignore it.
        ;;
    ..)
        # Parent dir; strip last assembled component from result.
        func_dirname "$func_normal_abspath_result"
        func_normal_abspath_result=$func_dirname_result
        ;;
    *)
        # Actual path component, append it.

func_normal_abspath_result=$func_normal_abspath_result/$func_normal_ab
spath_tcomponent
        ;;
esac
done
# Restore leading double-slash if one was found on entry.

func_normal_abspath_result=$func_normal_abspath_altnamespace$func_norm
al_abspath_result
}

# func_relative_path SRCDIR DSTDIR
# generates a relative path from SRCDIR to DSTDIR, with a trailing
# slash if non-empty, suitable for immediately appending a filename
# without needing to append a separator.
#           value returned in "$func_relative_path_result"
func_relative_path ()
{
    func_relative_path_result=
    func_normal_abspath "$1"
    func_relative_path_tlibdir=$func_normal_abspath_result
    func_normal_abspath "$2"
    func_relative_path_tbindir=$func_normal_abspath_result

# Ascend the tree starting from libdir
while ;; do
    # check if we have found a prefix of bindir
    case $func_relative_path_tbindir in
        $func_relative_path_tlibdir)
            # found an exact match
            func_relative_path_tcancelled=
            break
            ;;
        $func_relative_path_tlibdir*)
            # found a matching prefix
            func_stripname "$func_relative_path_tlibdir" ''
"$func_relative_path_tbindir"
            func_relative_path_tcancelled=$func_stripname_result
            if test -z "$func_relative_path_result"; then

```

```

        func_relative_path_result=.
    fi
    break
    ;;
*)
    func_dirname $func_relative_path_tlibdir
    func_relative_path_tlibdir=${func_dirname_result}
    if test "x$func_relative_path_tlibdir" = x ; then
        # Have to descend all the way to the root!
        func_relative_path_result=./$func_relative_path_result
        func_relative_path_tcancelled=$func_relative_path_tbindir
        break
    fi
    func_relative_path_result=./$func_relative_path_result
    ;;
esac
done

# Now calculate path; take care to avoid doubling-up slashes.
func_stripname '/' '/' "$func_relative_path_result"
func_relative_path_result=$func_stripname_result
func_stripname '/' '/' "$func_relative_path_tcancelled"
if test "x$func_stripname_result" != x ; then

func_relative_path_result=${func_relative_path_result}/${func_stripnam
e_result}
fi

# Normalisation. If bindir is libdir, return empty string,
# else relative path ending with a slash; either way, target
# file name can be directly appended.
if test ! -z "$func_relative_path_result"; then
    func_stripname './' '' "$func_relative_path_result/"
    func_relative_path_result=$func_stripname_result
fi
}

# The name of this program:
func_dirname_and_basename "$progpdir"
progname=$func_basename_result

# Make sure we have an absolute path for reexecution:
case $progpdir in
    [\\/] * | [A-Za-z]: \\ * ) ;;
    * [\\/] * )
        progdir=$func_dirname_result
        progdir=`cd "$progdir" && pwd`
        progpath="$progdir/$progname"
        ;;
*)
    save_IFS="$IFS"
    IFS=${PATH_SEPARATOR-:}

```

```

    for progdir in $PATH; do
        IFS="$save_IFS"
        test -x "$progdir/$progname" && break
    done
    IFS="$save_IFS"
    test -n "$progdir" || progdir=`pwd`
    proppath="$progdir/$progname"
    ;;
esac

# Sed substitution that helps us do robust quoting.  It backslashifies
# metacharacters that are still active within double-quoted strings.
Xsed="${SED}"' -e 1s/^X//'
sed_quote_subst='s/\([\"$\\]\)/\\\1/g'

# Same as above, but do not quote variable references.
double_quote_subst='s/\([\"\\]\)/\\\1/g'

# Sed substitution that turns a string into a regex matching for the
# string literally.
sed_make_literal_regex='s,[].[^$\\*\/],\\&,g'

# Sed substitution that converts a w32 file name or path
# which contains forward slashes, into one that contains
# (escaped) backslashes.  A very naive implementation.
lt_sed_naive_backslashify='s|\\|\\\\|g;s|/|\\\\|g;s|\\|\\\\\\\\|g'

# Re-`\\` parameter expansions in output of double_quote_subst that
# were
# `\\`-ed in input to the same.  If an odd number of `\\` preceded a '$'
# in input to double_quote_subst, that '$' was protected from
# expansion.
# Since each input `\\` is now two `\\`s, look for any number of runs of
# four `\\`s followed by two `\\`s and then a '$'.  `\\` that '$'.
bs='\\'
bs2='\\\\'
bs4='\\\\\\\\'
dollar='\$'
sed_double_backslash="\
s/$bs4/&\\
/g
s/^$bs2$dollar/$bs&/
s/\\([^\$bs]\\\\)$bs2$dollar/\\1$bs2$bs$dollar/g
s/\\n//g"

# Standard options:
opt_dry_run=false
opt_help=false
opt_quiet=false
opt_verbose=false
opt_warning=:

```

```

# func_echo arg...
# Echo program name prefixed message, along with the current mode
# name if it has been set yet.
func_echo ()
{
    $ECHO "$progname: ${opt_mode+$opt_mode: }$*"
}

# func_verbose arg...
# Echo program name prefixed message in verbose mode only.
func_verbose ()
{
    $opt_verbose && func_echo "${1+"$@"}

    # A bug in bash halts the script if the last line of a function
    # fails when set -e is in force, so we need another command to
    # work around that:
    :
}

# func_echo_all arg...
# Invoke $ECHO with all args, space-separated.
func_echo_all ()
{
    $ECHO "$*"
}

# func_error arg...
# Echo program name prefixed message to standard error.
func_error ()
{
    $ECHO "$progname: ${opt_mode+$opt_mode: }"${1+"$@"} 1>&2
}

# func_warning arg...
# Echo program name prefixed warning message to standard error.
func_warning ()
{
    $opt_warning && $ECHO "$progname: ${opt_mode+$opt_mode: }warning:
"${1+"$@"} 1>&2

    # bash bug again:
    :
}

# func_fatal_error arg...
# Echo program name prefixed message to standard error, and exit.
func_fatal_error ()
{
    func_error "${1+"$@"}
    exit $EXIT_FAILURE
}

```



```

# func_fatal_help arg...
# Echo program name prefixed message to standard error, followed by
# a help hint, and exit.
func_fatal_help ()
{
    func_error ${1+"$@"}
    func_fatal_error "$help"
}
help="Try \`${progname} --help' for more information." ## default

# func_grep expression filename
# Check whether EXPRESSION matches any line of FILENAME, without
# output.
func_grep ()
{
    $GREP "$1" "$2" >/dev/null 2>&1
}

# func_mkdir_p directory-path
# Make sure the entire path to DIRECTORY-PATH is available.
func_mkdir_p ()
{
    my_directory_path="$1"
    my_dir_list=

    if test -n "$my_directory_path" && test "$opt_dry_run" != ":";
then
    # Protect directory names starting with '-'
    case $my_directory_path in
        -*) my_directory_path="./$my_directory_path" ;;
    esac

    # While some portion of DIR does not yet exist...
    while test ! -d "$my_directory_path"; do
        # ...make a list in topmost first order. Use a colon
delimited
        # list incase some portion of path contains whitespace.
        my_dir_list="$my_directory_path:$my_dir_list"

        # If the last portion added has no slash in it, the list is
done
        case $my_directory_path in */*) ;; *) break ;; esac

        # ...otherwise throw away the child directory and loop
        my_directory_path=`$ECHO "$my_directory_path" | $SED -e
"$dirname" `
    done
    my_dir_list=`$ECHO "$my_dir_list" | $SED 's,:*$,,' `

```

```

save_mkdir_p_IFS="$IFS"; IFS=':'
for my_dir in $my_dir_list; do
IFS="$save_mkdir_p_IFS"
    # mkdir can fail with a `File exist' error if two processes
    # try to create one of the directories concurrently. Don't
    # stop in that case!
    $MKDIR "$my_dir" 2>/dev/null || :
done
IFS="$save_mkdir_p_IFS"

# Bail out if we (or some other process) failed to create a
directory.
test -d "$my_directory_path" || \
    func_fatal_error "Failed to create \`${1}'"
fi
}

# func_mktempdir [string]
# Make a temporary directory that won't clash with other running
# libtool processes, and avoids race conditions if possible. If
# given, STRING is the basename for that directory.
func_mktempdir ()
{
    my_template="${TMPDIR-/tmp}/${1-$progname}"

    if test "$opt_dry_run" = ":"; then
        # Return a directory name, but don't create it in dry-run mode
        my_tmpdir="${my_template}--$$"
    else

        # If mktemp works, use that first and foremost
        my_tmpdir=`mktemp -d "${my_template}-XXXXXXXX" 2>/dev/null`

        if test ! -d "$my_tmpdir"; then
            # Failing that, at least try and use $RANDOM to avoid a race
            my_tmpdir="${my_template}-${RANDOM-0}$$"

            save_mktempdir_umask=`umask`
            umask 0077
            $MKDIR "$my_tmpdir"
            umask $save_mktempdir_umask
        fi

        # If we're not in dry-run mode, bomb out on failure
        test -d "$my_tmpdir" || \
            func_fatal_error "cannot create temporary directory
\`${my_tmpdir}'"
        fi

        $ECHO "$my_tmpdir"

```

```

}

# func_quote_for_eval arg
# Aesthetically quote ARG to be evaled later.
# This function returns two values: FUNC_QUOTE_FOR_EVAL_RESULT
# is double-quoted, suitable for a subsequent eval, whereas
# FUNC_QUOTE_FOR_EVAL_UNQUOTED_RESULT has merely all characters
# which are still active within double quotes backslashified.
func_quote_for_eval ()
{
    case $1 in
        *[\$\`\"$]*)
            func_quote_for_eval_unquoted_result=`$ECHO "$1" | $SED
"$sed_quote_subst" ` ;;
        *)
            func_quote_for_eval_unquoted_result="$1" ;;
    esac

    case $func_quote_for_eval_unquoted_result in
        # Double-quote args containing shell metacharacters to delay
        # word splitting, command substitution and and variable
        # expansion for a subsequent eval.
        # Many Bourne shells cannot handle close brackets correctly
        # in scan sets, so we specify it separately.
        *[[\~\#\^\&\*\(\)\{\}\|\|;\<\>\?\'\ \ ]*|*|*"")
            func_quote_for_eval_result="\"$func_quote_for_eval_unquoted_result\"
                ;;
        *)
            func_quote_for_eval_result="$func_quote_for_eval_unquoted_result"
    esac
}

# func_quote_for_expand arg
# Aesthetically quote ARG to be evaled later; same as above,
# but do not quote variable references.
func_quote_for_expand ()
{
    case $1 in
        *[\$\`\"$]*)
            my_arg=`$ECHO "$1" | $SED \
                -e "$double_quote_subst" -e "$sed_double_backslash" ` ;;
        *)
            my_arg="$1" ;;
    esac

    case $my_arg in
        # Double-quote args containing shell metacharacters to delay
        # word splitting and command substitution for a subsequent eval.

```



```

    eval "$lt_user_locale
          $my_cmd"
    my_status=$?
    eval "$lt_safe_locale"
    if test "$my_status" -eq 0; then ;; else
    eval "(exit $my_status); $my_fail_exp"
    fi
  fi
}

# func_tr_sh
# Turn $1 into a string suitable for a shell variable name.
# Result is stored in $func_tr_sh_result. All characters
# not in the set a-zA-Z0-9_ are replaced with '_'. Further,
# if $1 begins with a digit, a '_' is prepended as well.
func_tr_sh ()
{
  case $1 in
    [0-9]* | *[^a-zA-Z0-9_]*)
      func_tr_sh_result=`$ECHO "$1" | $SED 's/^\([0-9]\)/_\1/; s/[^a-zA-Z0-9_]/_/g'`
      ;;
    * )
      func_tr_sh_result=$1
      ;;
  esac
}

# func_version
# Echo version message to standard output and exit.
func_version ()
{
  $opt_debug

  $SED -n '/(C)/!b go
:more
/\./!{
  N
  s/\n# / /
  b more
}
:go
/^# '$PROGRAM' (GNU /,/# warranty; / {
  s/^# //
  s/^# *$//
  s/\((C)\) [ 0-9,-]*\([ 1-9][0-9]*\)/\1\2/
  p
}' < "$progpath"
  exit $?
}

```

```

# func_usage
# Echo short help message to standard output and exit.
func_usage ()
{
    $opt_debug

    $SED -n '/^# Usage:/,/^# *.*--help/ {
        s/^# //
        s/^# *$//
        s/\$progname/'$progname'/
        p
    }' < "$progpath"
    echo
    $ECHO "run \"\$progname --help | more' for full usage"
    exit $?
}

# func_help [NOEXIT]
# Echo long help message to standard output and exit,
# unless 'noexit' is passed as argument.
func_help ()
{
    $opt_debug

    $SED -n '/^# Usage:/,/# Report bugs to/ {
        :print
        s/^# //
        s/^# *$//
        s*\$progname*\$progname'*
        s*\$host*'"$host"'*
        s*\$SHELL*'"$SHELL"'*
        s*\$LTCC*'"$LTCC"'*
        s*\$LTCFLAGS*'"$LTCFLAGS"'*
        s*\$LD*'"$LD"'*
        s/\$with_gnu_ld/'"$with_gnu_ld"'/
        s/\$automake_version/'"'`(${AUTOMAKE-automake} --version)
2>/dev/null |$SED lq`"/
        s/\$autoconf_version/'"'`(${AUTOCONF-autoconf} --version)
2>/dev/null |$SED lq`"/
        p
        d
    }
    /^# .* home page:/b print
    /^# General help using/b print
    ' < "$progpath"
    ret=$?
    if test -z "$1"; then
        exit $ret
    fi
}

# func_missing_arg argname

```

```

# Echo program name prefixed message to standard error and set global
# exit_cmd.
func_missing_arg ()
{
    $opt_debug

    func_error "missing argument for $1."
    exit_cmd=exit
}

# func_split_short_opt shortopt
# Set func_split_short_opt_name and func_split_short_opt_arg shell
# variables after splitting SHORTOPT after the 2nd character.
func_split_short_opt ()
{
    func_split_short_opt_arg=${1#??}
    func_split_short_opt_name=${1%"$func_split_short_opt_arg"}
} # Extended-shell func_split_short_opt implementation

# func_split_long_opt longopt
# Set func_split_long_opt_name and func_split_long_opt_arg shell
# variables after splitting LONGOPT at the '=' sign.
func_split_long_opt ()
{
    func_split_long_opt_name=${1%=*}
    func_split_long_opt_arg=${1#*=}
} # Extended-shell func_split_long_opt implementation

exit_cmd=:

magic="%%MAGIC variable%%"
magic_exe="%%MAGIC EXE variable%%"

# Global variables.
nonopt=
preserve_args=
lo2o="s/\\.lo\\$/.${objext}/"
o2lo="s/\\.\\${objext}\\$/.lo/"
extracted_archives=
extracted_serial=0

# If this variable is set in any of the actions, the command in it
# will be execed at the end. This prevents here-documents from being
# left over by shells.
exec_cmd=

```

```

# func_append var value
# Append VALUE to the end of shell variable VAR.
func_append ()
{
    eval "${1}+=\${2}"
} # Extended-shell func_append implementation

# func_append_quoted var value
# Quote VALUE and append to the end of shell variable VAR, separated
# by a space.
func_append_quoted ()
{
    func_quote_for_eval "${2}"
    eval "${1}+=\ \ \${func_quote_for_eval_result}"
} # Extended-shell func_append_quoted implementation

# func_arith arithmetic-term...
func_arith ()
{
    func_arith_result=$(( $* ))
} # Extended-shell func_arith implementation

# func_len string
# STRING may not start with a hyphen.
func_len ()
{
    func_len_result=${#1}
} # Extended-shell func_len implementation

# func_lo2o object
func_lo2o ()
{
    case ${1} in
        *.lo) func_lo2o_result=${1%.lo}.${objext} ;;
        *)    func_lo2o_result=${1} ;;
    esac
} # Extended-shell func_lo2o implementation

# func_xform libobj-or-source
func_xform ()
{
    func_xform_result=${1%.*}.lo
} # Extended-shell func_xform implementation

# func_fatal_configuration arg...
# Echo program name prefixed message to standard error, followed by
# a configuration failure hint, and exit.

```



```

func_fatal_configuration ()
{
    func_error ${1+"$@"}
    func_error "See the $PACKAGE documentation for more information."
    func_fatal_error "Fatal configuration error."
}

# func_config
# Display the configuration for all the tags in this script.
func_config ()
{
    re_begincf='^# ### BEGIN LIBTOOL'
    re_endcf='^# ### END LIBTOOL'

    # Default configuration.
    $SED "1,/$re_begincf CONFIG/d;/$re_endcf CONFIG/,\`$d" <
"$proppath"

    # Now print the configurations for the tags.
    for tagname in $taglist; do
        $SED -n "/$re_begincf TAG CONFIG: $tagname\$/,$re_endcf TAG
CONFIG: $tagname\$/p" < "$proppath"
    done

    exit $?
}

# func_features
# Display the features supported by this script.
func_features ()
{
    echo "host: $host"
    if test "$build_libtool_libs" = yes; then
        echo "enable shared libraries"
    else
        echo "disable shared libraries"
    fi
    if test "$build_old_libs" = yes; then
        echo "enable static libraries"
    else
        echo "disable static libraries"
    fi

    exit $?
}

# func_enable_tag tagname
# Verify that TAGNAME is valid, and either flag an error and exit, or
# enable the TAGNAME tag. We also add TAGNAME to the global $taglist
# variable here.
func_enable_tag ()

```

```

{
# Global variable:
tagname="$1"

re_begincf="^# ### BEGIN LIBTOOL TAG CONFIG: $tagname\$"
re_endcf="^# ### END LIBTOOL TAG CONFIG: $tagname\$"
sed_extractcf="/$re_begincf/,/$re_endcf/p"

# Validate tagname.
case $tagname in
*[_A-Za-z0-9,/]*)
func_fatal_error "invalid tag name: $tagname"
;;
esac

# Don't test for the "default" C tag, as we know it's
# there but not specially marked.
case $tagname in
CC) ;;
*)
if $GREP "$re_begincf" "$progp" >/dev/null 2>&1; then
taglist="$taglist $tagname"

# Evaluate the configuration. Be careful to quote the path
# and the sed script, to avoid splitting on whitespace, but
# also don't use non-portable quotes within backquotes within
# quotes we have to do it in 2 steps:
extractedcf=`$SED -n -e "$sed_extractcf" < "$progp" `
eval "$extractedcf"
else
func_error "ignoring unknown tag $tagname"
fi
;;
esac
}

# func_check_version_match
# Ensure that we are using m4 macros, and libtool script from the same
# release of libtool.
func_check_version_match ()
{
if test "$package_revision" != "$macro_revision"; then
if test "$VERSION" != "$macro_version"; then
if test -z "$macro_version"; then
cat >&2 <<_LT_EOF
$progname: Version mismatch error. This is $PACKAGE $VERSION, but the
$progname: definition of this LT_INIT comes from an older release.
$progname: You should recreate aclocal.m4 with macros from $PACKAGE
$VERSION
$progname: and run autoconf again.
_LT_EOF
else

```

```

        cat >&2 <<_LT_EOF
$progname: Version mismatch error.  This is $PACKAGE $VERSION, but the
$progname: definition of this LT_INIT comes from $PACKAGE
$macro_version.
$progname: You should recreate aclocal.m4 with macros from $PACKAGE
$VERSION
$progname: and run autoconf again.
_LT_EOF
    fi
    else
        cat >&2 <<_LT_EOF
$progname: Version mismatch error.  This is $PACKAGE $VERSION,
revision $package_revision,
$progname: but the definition of this LT_INIT comes from revision
$macro_revision.
$progname: You should recreate aclocal.m4 with macros from revision
$package_revision
$progname: of $PACKAGE $VERSION and run autoconf again.
_LT_EOF
    fi

    exit $EXIT_MISMATCH
fi
}

# Shorthand for --mode=foo, only valid as the first argument
case $1 in
clean|clea|cle|cl)
    shift; set dummy --mode clean ${1+"$@"}; shift
    ;;
compile|compil|compi|comp|com|co|c)
    shift; set dummy --mode compile ${1+"$@"}; shift
    ;;
execute|execut|execu|exec|exe|ex|e)
    shift; set dummy --mode execute ${1+"$@"}; shift
    ;;
finish|finis|fini|fin|fi|f)
    shift; set dummy --mode finish ${1+"$@"}; shift
    ;;
install|instal|insta|inst|ins|in|i)
    shift; set dummy --mode install ${1+"$@"}; shift
    ;;
link|lin|li|l)
    shift; set dummy --mode link ${1+"$@"}; shift
    ;;
uninstall|uninstal|uninsta|uninst|unins|unin|uni|un|u)
    shift; set dummy --mode uninstall ${1+"$@"}; shift
    ;;
esac

```

```

# Option defaults:
opt_debug=:
opt_dry_run=false
opt_config=false
opt_preserve_dup_deps=false
opt_features=false
opt_finish=false
opt_help=false
opt_help_all=false
opt_silent=:
opt_warning=:
opt_verbose=:
opt_silent=false
opt_verbose=false

# Parse options once, thoroughly.  This comes as soon as possible in
the
# script to make things like `--version' happen as quickly as we can.
{
  # this just eases exit handling
  while test $# -gt 0; do
    opt="$1"
    shift
    case $opt in
      --debug|-x)      opt_debug='set -x'
                       func_echo "enabling shell trace mode"
                       $opt_debug
                       ;;
      --dry-run|--dryrun|-n)
                       opt_dry_run=:
                       ;;
      --config)
                       opt_config=:
func_config
                       ;;
      --dlopen|-dlopen)
                       optarg="$1"
                       opt_dlopen="${opt_dlopen+$opt_dlopen
}$optarg"
                       shift
                       ;;
      --preserve-dup-deps)
                       opt_preserve_dup_deps=:
                       ;;
      --features)
                       opt_features=:
func_features
                       ;;
      --finish)
                       opt_finish=:

```

```

set dummy --mode finish ${1+"$@"}; shift
    ;;
    --help)
        opt_help=:
        ;;
    --help-all)
        opt_help_all=:
opt_help=': help-all'
        ;;
    --mode)
        test $# = 0 && func_missing_arg $opt && break
        optarg="$1"
        opt_mode="$optarg"
case $optarg in
    # Valid mode arguments:
    clean|compile|execute|finish|install|link|relink|uninstall) ;;

    # Catch anything else as an error
    *) func_error "invalid argument for $opt"
        exit_cmd=exit
        break
        ;;
esac
        shift
        ;;
    --no-silent|--no-quiet)
        opt_silent=false
preserve_args+=" $opt"
        ;;
    --no-warning|--no-warn)
        opt_warning=false
preserve_args+=" $opt"
        ;;
    --no-verbose)
        opt_verbose=false
preserve_args+=" $opt"
        ;;
    --silent|--quiet)
        opt_silent=:
preserve_args+=" $opt"
        opt_verbose=false
        ;;
    --verbose|-v)
        opt_verbose=:
preserve_args+=" $opt"
opt_silent=false
        ;;
    --tag)
        test $# = 0 && func_missing_arg $opt && break
        optarg="$1"
        opt_tag="$optarg"
preserve_args+=" $opt $optarg"

```

```

func_enable_tag "$optarg"
    shift
    ;;

    -\?|-h)          func_usage          ;;
    --help)          func_help           ;;
    --version) func_version              ;;

    # Separate optargs to long options:
    --*=*)
        func_split_long_opt "$opt"
        set dummy "$func_split_long_opt_name"
"$func_split_long_opt_arg" ${1+"$@"}
        shift
        ;;

    # Separate non-argument short options:
    -\?*|-h*|-n*|-v*)
        func_split_short_opt "$opt"
        set dummy "$func_split_short_opt_name" "--"
"$func_split_short_opt_arg" ${1+"$@"}
        shift
        ;;

    --)              break                ;;
    -*)              func_fatal_help "unrecognized option \`$opt'" ;;
    *)              set dummy "$opt" ${1+"$@"}; shift; break ;;
esac
done

# Validate options:

# save first non-option argument
if test "$#" -gt 0; then
    nonopt="$opt"
    shift
fi

# preserve --debug
test "$opt_debug" = : || preserve_args+=" --debug"

case $host in
    *cygwin* | *mingw* | *pw32* | *cegcc*)
        # don't eliminate duplications in $postdeps and $predeps
        opt_duplicate_compiler_generated_deps=:
        ;;
    *)
        opt_duplicate_compiler_generated_deps=$opt_preserve_dup_deps
        ;;
esac

$opt_help || {

```

```

# Sanity checks first:
func_check_version_match

if test "$build_libtool_libs" != yes && test "$build_old_libs" !=
yes; then
    func_fatal_configuration "not configured to build any kind of
library"
fi

# Darwin sucks
eval std_shrext="\$shrext_cmds\"

# Only execute mode is allowed to have -dlopen flags.
if test -n "$opt_dlopen" && test "$opt_mode" != execute; then
    func_error "unrecognized option \`-dlopen'"
    $ECHO "$help" 1>&2
    exit $EXIT_FAILURE
fi

# Change the help message to a mode-specific one.
generic_help="$help"
help="Try \`$progname --help --mode=$opt_mode' for more
information."
}

# Bail if the options were screwed
$exit_cmd $EXIT_FAILURE
}

## ----- ##
##   Main.   ##
## ----- ##

# func_lalib_p file
# True iff FILE is a libtool `.la' library or `.lo' object file.
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_lalib_p ()
{
    test -f "$1" &&
    $SED -e 4q "$1" 2>/dev/null \
    | $GREP "^# Generated by .*$PACKAGE" > /dev/null 2>&1
}

# func_lalib_unsafe_p file
# True iff FILE is a libtool `.la' library or `.lo' object file.
# This function implements the same check as func_lalib_p without
# resorting to external programs. To this end, it redirects stdin and

```

```

# closes it afterwards, without saving the original file descriptor.
# As a safety measure, use it only where a negative result would be
# fatal anyway. Works if `file' does not exist.
func_lalib_unsafe_p ()
{
    lalib_p=no
    if test -f "$1" && test -r "$1" && exec 5<&0 <"$1"; then
        for lalib_p_1 in 1 2 3 4
        do
            read lalib_p_line
            case "$lalib_p_line" in
                \#\ Generated\ by\ *$PACKAGE* ) lalib_p=yes; break;;
            esac
        done
        exec 0<&5 5<&-
    fi
    test "$lalib_p" = yes
}

# func_ltwrapper_script_p file
# True iff FILE is a libtool wrapper script
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_script_p ()
{
    func_lalib_p "$1"
}

# func_ltwrapper_executable_p file
# True iff FILE is a libtool wrapper executable
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_executable_p ()
{
    func_ltwrapper_exec_suffix=
    case $1 in
        *.exe) ;;
        *) func_ltwrapper_exec_suffix=.exe ;;
    esac
    $GREP "$magic_exe" "$1$func_ltwrapper_exec_suffix" >/dev/null 2>&1
}

# func_ltwrapper_scriptname file
# Assumes file is an ltwrapper_executable
# uses $file to determine the appropriate filename for a
# temporary ltwrapper_script.
func_ltwrapper_scriptname ()
{
    func_dirname_and_basename "$1" "" "."
    func_stripname '' '.exe' "$func_basename_result"
}

```



```

func_ltwrapper_scriptname_result="$func_dirname_result/$objdir/${func_
stripname_result}_ltshwrapper"
}

# func_ltwrapper_p file
# True iff FILE is a libtool wrapper script or wrapper executable
# This function is only a basic sanity check; it will hardly flush out
# determined imposters.
func_ltwrapper_p ()
{
    func_ltwrapper_script_p "$1" || func_ltwrapper_executable_p "$1"
}

# func_execute_cmds commands fail_cmd
# Execute tilde-delimited COMMANDS.
# If FAIL_CMD is given, eval that upon failure.
# FAIL_CMD may read-access the current command in variable CMD!
func_execute_cmds ()
{
    $opt_debug
    save_ifs=$IFS; IFS='~'
    for cmd in $1; do
        IFS=$save_ifs
        eval cmd="\$cmd\"
        func_show_eval "$cmd" "${2-:}"
    done
    IFS=$save_ifs
}

# func_source file
# Source FILE, adding directory component if necessary.
# Note that it is not necessary on cygwin/mingw to append a dot to
# FILE even if both FILE and FILE.exe exist: automatic-append-.exe
# behavior happens only for exec(3), not for open(2)! Also, sourcing
# `FILE.' does not work on cygwin managed mounts.
func_source ()
{
    $opt_debug
    case $1 in
        /* | *\*) . "$1" ;;
        *) . "./$1" ;;
    esac
}

# func_resolve_sysroot PATH
# Replace a leading = in PATH with a sysroot. Store the result into
# func_resolve_sysroot_result
func_resolve_sysroot ()

```

```

{
  func_resolve_sysroot_result=$1
  case $func_resolve_sysroot_result in
  =*)
    func_stripname '=' '' "$func_resolve_sysroot_result"
    func_resolve_sysroot_result=${lt_sysroot}${func_stripname_result}
    ;;
  esac
}

# func_replace_sysroot PATH
# If PATH begins with the sysroot, replace it with = and
# store the result into func_replace_sysroot_result.
func_replace_sysroot ()
{
  case "$lt_sysroot:$1" in
  ?*:"$lt_sysroot"*)
    func_stripname "$lt_sysroot" '' "$1"
    func_replace_sysroot_result="=${func_stripname_result}"
    ;;
  *)
    # Including no sysroot.
    func_replace_sysroot_result=$1
    ;;
  esac
}

# func_infer_tag arg
# Infer tagged configuration to use if any are available and
# if one wasn't chosen via the "--tag" command line option.
# Only attempt this if the compiler in the base compile
# command doesn't match the default compiler.
# arg is usually of the form 'gcc ...'
func_infer_tag ()
{
  $opt_debug
  if test -n "$available_tags" && test -z "$tagname"; then
    CC_quoted=
    for arg in $CC; do
      func_append_quoted CC_quoted "$arg"
    done
    CC_expanded=`func_echo_all $CC`
    CC_quoted_expanded=`func_echo_all $CC_quoted`
    case $@ in
    # Blanks in the command may have been stripped by the calling
shell,
    # but not from the CC environment variable when configure was
run.
    " $CC "* | "$CC "* | " $CC_expanded "* | "$CC_expanded "* | \
    " $CC_quoted"* | "$CC_quoted "* | " $CC_quoted_expanded "* |
"$CC_quoted_expanded "*) ;;
    # Blanks at the start of $base_compile will cause this to fail

```

```

# if we don't check for them as well.
*)
for z in $available_tags; do
    if $GREP "^# ### BEGIN LIBTOOL TAG CONFIG: $z$" < "$progp" >
/dev/null; then
    # Evaluate the configuration.
    eval "`${SED} -n -e '/^# ### BEGIN LIBTOOL TAG CONFIG:
'$z'$/,/^# ### END LIBTOOL TAG CONFIG: '$z'$/p' < $progp`"
    CC_quoted=
    for arg in $CC; do
        # Double-quote args containing other shell metacharacters.
        func_append_quoted CC_quoted "$arg"
    done
    CC_expanded=`func_echo_all $CC`
    CC_quoted_expanded=`func_echo_all $CC_quoted`
    case "$@" in
        " $CC "*" | "$CC "*" | " $CC_expanded "*" | "$CC_expanded "*" | \
        " $CC_quoted"* | "$CC_quoted "*" | " $CC_quoted_expanded "*" |
"$CC_quoted_expanded "*))
        # The compiler in the base compile command matches
        # the one in the tagged configuration.
        # Assume this is the tagged configuration we want.
        tagname=$z
        break
        ;;
    esac
fi
done
# If $tagname still isn't set, then no tagged configuration
# was found and let the user know that the "--tag" command
# line option must be used.
if test -z "$tagname"; then
    func_echo "unable to infer tagged configuration"
    func_fatal_error "specify a tag with `--tag'"
# else
#     func_verbose "using $tagname tagged configuration"
fi
;;
esac
fi
}

```

```

# func_write_libtool_object output_name pic_name nonpic_name
# Create a libtool object file (analogous to a ".la" file),
# but don't create it if we're doing a dry run.
func_write_libtool_object ()
{
    write_libobj=${1}
    if test "$build_libtool_libs" = yes; then
        write_lobj=`'$2`\'
    fi
}

```

```

else
    write_lobj=none
fi

if test "$build_old_libs" = yes; then
    write_oldobj='\${3}\'
else
    write_oldobj=none
fi

$opt_dry_run || {
    cat >${write_libobj}T <<EOF
# $write_libobj - a libtool object file
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION
#
# Please DO NOT delete this file!
# It is necessary for linking the library.

# Name of the PIC object.
pic_object=$write_lobj

# Name of the non-PIC object
non_pic_object=$write_oldobj

EOF
    $MV "${write_libobj}T" "${write_libobj}"
}

#####
# FILE NAME AND PATH CONVERSION HELPER FUNCTIONS #
#####

# func_convert_core_file_wine_to_w32 ARG
# Helper function used by file name conversion functions when $build
# is *nix,
# and $host is mingw, cygwin, or some other w32 environment. Relies on
# a
# correctly configured wine environment available, with the winepath
# program
# in $build's $PATH.
#
# ARG is the $build file name to be converted to w32 format.
# Result is available in $func_convert_core_file_wine_to_w32_result,
# and will
# be empty on error (or when ARG is empty)
func_convert_core_file_wine_to_w32 ()
{
    $opt_debug
    func_convert_core_file_wine_to_w32_result="$1"
    if test -n "$1"; then

```

```

    # Unfortunately, winepath does not exit with a non-zero error
code, so we
    # are forced to check the contents of stdout. On the other hand,
if the
    # command is not found, the shell will set an exit code of 127 and
print
    # *an error message* to stdout. So we must check for both error
code of
    # zero AND non-empty stdout, which explains the odd construction:
    func_convert_core_file_wine_to_w32_tmp=`winepath -w "$1"
2>/dev/null`
    if test "$?" -eq 0 && test -n
"${func_convert_core_file_wine_to_w32_tmp}"; then
        func_convert_core_file_wine_to_w32_result=`$ECHO
"$func_convert_core_file_wine_to_w32_tmp" |
        $SED -e "$lt_sed_naive_backslashify"`
    else
        func_convert_core_file_wine_to_w32_result=
    fi
fi
}
# end: func_convert_core_file_wine_to_w32

```

```

# func_convert_core_path_wine_to_w32 ARG
# Helper function used by path conversion functions when $build is
*nix, and
# $host is mingw, cygwin, or some other w32 environment. Relies on a
correctly
# configured wine environment available, with the winepath program in
$build's
# $PATH. Assumes ARG has no leading or trailing path separator
characters.
#
# ARG is path to be converted from $build format to win32.
# Result is available in $func_convert_core_path_wine_to_w32_result.
# Unconvertible file (directory) names in ARG are skipped; if no
directory names
# are convertible, then the result may be empty.
func_convert_core_path_wine_to_w32 ()
{
    $opt_debug
    # unfortunately, winepath doesn't convert paths, only file names
    func_convert_core_path_wine_to_w32_result=""
    if test -n "$1"; then
        oldIFS=$IFS
        IFS=:
        for func_convert_core_path_wine_to_w32_f in $1; do
            IFS=$oldIFS
            func_convert_core_file_wine_to_w32
"$func_convert_core_path_wine_to_w32_f"
            if test -n "$func_convert_core_file_wine_to_w32_result" ; then

```

```

        if test -z "$func_convert_core_path_wine_to_w32_result"; then

func_convert_core_path_wine_to_w32_result="$func_convert_core_file_wine_to_w32_result"
        else
            func_append func_convert_core_path_wine_to_w32_result
";$func_convert_core_file_wine_to_w32_result"
            fi
        fi
    done
    IFS=$oldIFS
fi
}
# end: func_convert_core_path_wine_to_w32

# func_cygpath ARGS...
# Wrapper around calling the cygpath program via LT_CYGPATH. This is
used when
# when (1) $build is *nix and Cygwin is hosted via a wine environment;
or (2)
# $build is MSYS and $host is Cygwin, or (3) $build is Cygwin. In case
(1) or
# (2), returns the Cygwin file name or path in func_cygpath_result
(input
# file name or path is assumed to be in w32 format, as previously
converted
# from $build's *nix or MSYS format). In case (3), returns the w32
file name
# or path in func_cygpath_result (input file name or path is assumed
to be in
# Cygwin format). Returns an empty string on error.
#
# ARGS are passed to cygpath, with the last one being the file name or
path to
# be converted.
#
# Specify the absolute *nix (or w32) name to cygpath in the LT_CYGPATH
# environment variable; do not put it in $PATH.
func_cygpath ()
{
    $opt_debug
    if test -n "$LT_CYGPATH" && test -f "$LT_CYGPATH"; then
        func_cygpath_result=`$LT_CYGPATH "$@" 2>/dev/null`
        if test "$?" -ne 0; then
            # on failure, ensure result is empty
            func_cygpath_result=
        fi
    else
        func_cygpath_result=
        func_error "LT_CYGPATH is empty or specifies non-existent file:
\`$LT_CYGPATH'"
    fi
}

```

```

    fi
}
#end: func_cygpath

# func_convert_core_msys_to_w32 ARG
# Convert file name or path ARG from MSYS format to w32 format.
Return
# result in func_convert_core_msys_to_w32_result.
func_convert_core_msys_to_w32 ()
{
    $opt_debug
    # awkward: cmd appends spaces to result
    func_convert_core_msys_to_w32_result=`( cmd //c echo "$1" )
2>/dev/null |
    ${SED} -e 's/[ ]*$//' -e "$!t_sed_naive_backslashify"`
}
#end: func_convert_core_msys_to_w32

# func_convert_file_check ARG1 ARG2
# Verify that ARG1 (a file name in $build format) was converted to
$host
# format in ARG2. Otherwise, emit an error message, but continue
(resetting
# func_to_host_file_result to ARG1).
func_convert_file_check ()
{
    $opt_debug
    if test -z "$2" && test -n "$1" ; then
        func_error "Could not determine host file name corresponding to"
        func_error "  \`$1'"
        func_error "Continuing, but uninstalled executables may not work."
        # Fallback:
        func_to_host_file_result="$1"
    fi
}
# end func_convert_file_check

# func_convert_path_check FROM_PATHSEP TO_PATHSEP FROM_PATH TO_PATH
# Verify that FROM_PATH (a path in $build format) was converted to
$host
# format in TO_PATH. Otherwise, emit an error message, but continue,
resetting
# func_to_host_file_result to a simplistic fallback value (see below).
func_convert_path_check ()
{
    $opt_debug
    if test -z "$4" && test -n "$3"; then
        func_error "Could not determine the host path corresponding to"
        func_error "  \`$3'"
    fi
}

```

```

func_error "Continuing, but uninstalled executables may not work."
# Fallback. This is a deliberately simplistic "conversion" and
# should not be "improved". See libtool.info.
if test "x$1" != "x$2"; then
    lt_replace_pathsep_chars="s|$1|$2|g"
    func_to_host_path_result=`echo "$3" |
        $SED -e "$lt_replace_pathsep_chars"`
else
    func_to_host_path_result="$3"
fi
fi
}
# end func_convert_path_check

# func_convert_path_front_back_pathsep FRONTPAT BACKPAT REPL ORIG
# Modifies func_to_host_path_result by prepending REPL if ORIG matches
FRONTPAT
# and appending REPL if ORIG matches BACKPAT.
func_convert_path_front_back_pathsep ()
{
    $opt_debug
    case $4 in
    $1 ) func_to_host_path_result="$3$func_to_host_path_result"
        ;;
    esac
    case $4 in
    $2 ) func_to_host_path_result+="$3"
        ;;
    esac
}
# end func_convert_path_front_back_pathsep

#####
# $build to $host FILE NAME CONVERSION FUNCTIONS #
#####
# invoked via `sto_host_file_cmd ARG'
#
# In each case, ARG is the path to be converted from $build to $host
format.
# Result will be available in $func_to_host_file_result.

# func_to_host_file ARG
# Converts the file name ARG from $build format to $host format.
Return result
# in func_to_host_file_result.
func_to_host_file ()
{
    $opt_debug
    sto_host_file_cmd "$1"
}

```



```

}
# end func_to_host_file

# func_to_tool_file ARG LAZY
# converts the file name ARG from $build format to toolchain format.
Return
# result in func_to_tool_file_result. If the conversion in use is
listed
# in (the comma separated) LAZY, no conversion takes place.
func_to_tool_file ()
{
    $opt_debug
    case , $2, in
        *, "$to_tool_file_cmd", *)
            func_to_tool_file_result=$1
            ;;
        *)
            $to_tool_file_cmd "$1"
            func_to_tool_file_result=$func_to_host_file_result
            ;;
    esac
}
# end func_to_tool_file

# func_convert_file_noop ARG
# Copy ARG to func_to_host_file_result.
func_convert_file_noop ()
{
    func_to_host_file_result="$1"
}
# end func_convert_file_noop

# func_convert_file_msys_to_w32 ARG
# Convert file name ARG from (mingw) MSYS to (mingw) w32 format;
automatic
# conversion to w32 is not available inside the cwrapper. Returns
result in
# func_to_host_file_result.
func_convert_file_msys_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_msys_to_w32 "$1"
        func_to_host_file_result="$func_convert_core_msys_to_w32_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_msys_to_w32

```

```

# func_convert_file_cygwin_to_w32 ARG
# Convert file name ARG from Cygwin to w32 format. Returns result in
# func_to_host_file_result.
func_convert_file_cygwin_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        # because $build is cygwin, we call "the" cygpath in $PATH; no
        # need to use
        # LT_CYGPATH in this case.
        func_to_host_file_result=`cygpath -m "$1"`
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_cygwin_to_w32

# func_convert_file_nix_to_w32 ARG
# Convert file name ARG from *nix to w32 format. Requires a wine
# environment
# and a working winepath. Returns result in func_to_host_file_result.
func_convert_file_nix_to_w32 ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_file_wine_to_w32 "$1"

    func_to_host_file_result="$func_convert_core_file_wine_to_w32_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_nix_to_w32

# func_convert_file_msys_to_cygwin ARG
# Convert file name ARG from MSYS to Cygwin format. Requires
# LT_CYGPATH set.
# Returns result in func_to_host_file_result.
func_convert_file_msys_to_cygwin ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        func_convert_core_msys_to_w32 "$1"
        func_cygpath -u "$func_convert_core_msys_to_w32_result"
        func_to_host_file_result="$func_cygpath_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}

```

```

}
# end func_convert_file_msys_to_cygwin

# func_convert_file_nix_to_cygwin ARG
# Convert file name ARG from *nix to Cygwin format.  Requires Cygwin
installed
# in a wine environment, working winepath, and LT_CYGPATH set.
Returns result
# in func_to_host_file_result.
func_convert_file_nix_to_cygwin ()
{
    $opt_debug
    func_to_host_file_result="$1"
    if test -n "$1"; then
        # convert from *nix to w32, then use cygpath to convert from w32
to cygwin.
        func_convert_core_file_wine_to_w32 "$1"
        func_cygpath -u "$func_convert_core_file_wine_to_w32_result"
        func_to_host_file_result="$func_cygpath_result"
    fi
    func_convert_file_check "$1" "$func_to_host_file_result"
}
# end func_convert_file_nix_to_cygwin

#####
# $build to $host PATH CONVERSION FUNCTIONS #
#####
# invoked via `to_host_path_cmd ARG'
#
# In each case, ARG is the path to be converted from $build to $host
format.
# The result will be available in $func_to_host_path_result.
#
# Path separators are also converted from $build format to $host
format.  If
# ARG begins or ends with a path separator character, it is preserved
(but
# converted to $host format) on output.
#
# All path conversion functions are named using the following
convention:
#   file name conversion function      : func_convert_file_X_to_Y ()
#   path conversion function          : func_convert_path_X_to_Y ()
# where, for any given $build/$host combination the 'X_to_Y' value is
the
# same.  If conversion functions are added for new $build/$host
combinations,
# the two new functions must follow this pattern, or
func_init_to_host_path_cmd
# will break.

```

```

# func_init_to_host_path_cmd
# Ensures that function "pointer" variable $to_host_path_cmd is set to
the
# appropriate value, based on the value of $to_host_file_cmd.
to_host_path_cmd=
func_init_to_host_path_cmd ()
{
    $opt_debug
    if test -z "$to_host_path_cmd"; then
        func_stripname 'func_convert_file_' '' "$to_host_file_cmd"
        to_host_path_cmd="func_convert_path_${func_stripname_result}"
    fi
}

# func_to_host_path ARG
# Converts the path ARG from $build format to $host format. Return
result
# in func_to_host_path_result.
func_to_host_path ()
{
    $opt_debug
    func_init_to_host_path_cmd
    $to_host_path_cmd "$1"
}
# end func_to_host_path

# func_convert_path_noop ARG
# Copy ARG to func_to_host_path_result.
func_convert_path_noop ()
{
    func_to_host_path_result="$1"
}
# end func_convert_path_noop

# func_convert_path_msys_to_w32 ARG
# Convert path ARG from (mingw) MSYS to (mingw) w32 format; automatic
# conversion to w32 is not available inside the cwrapper. Returns
result in
# func_to_host_path_result.
func_convert_path_msys_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # Remove leading and trailing path separator characters from ARG.
        MSYS
    fi
}

```

```

    # behavior is inconsistent here; cygpath turns them into '.' and
';.';
    # and winepath ignores them completely.
    func_stripname : : "$1"
    func_to_host_path_tmpl=$func_stripname_result
    func_convert_core_msys_to_w32 "$func_to_host_path_tmpl"
    func_to_host_path_result="$func_convert_core_msys_to_w32_result"
    func_convert_path_check : ";" \
        "$func_to_host_path_tmpl" "$func_to_host_path_result"
    func_convert_path_front_back_pathsep ":*" "':" ";" "$1"
fi
}
# end func_convert_path_msys_to_w32

# func_convert_path_cygwin_to_w32 ARG
# Convert path ARG from Cygwin to w32 format. Returns result in
# func_to_host_file_result.
func_convert_path_cygwin_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmpl=$func_stripname_result
        func_to_host_path_result=`cygpath -m -p "$func_to_host_path_tmpl"`
        func_convert_path_check : ";" \
            "$func_to_host_path_tmpl" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "':" ";" "$1"
    fi
}
# end func_convert_path_cygwin_to_w32

# func_convert_path_nix_to_w32 ARG
# Convert path ARG from *nix to w32 format. Requires a wine
environment and
# a working winepath. Returns result in func_to_host_file_result.
func_convert_path_nix_to_w32 ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmpl=$func_stripname_result
        func_convert_core_path_wine_to_w32 "$func_to_host_path_tmpl"

func_to_host_path_result="$func_convert_core_path_wine_to_w32_result"
        func_convert_path_check : ";" \
            "$func_to_host_path_tmpl" "$func_to_host_path_result"
    fi
}
# end func_convert_path_nix_to_w32

```

```

    func_convert_path_front_back_pathsep ":*" "*:;" ";" "$1"
fi
}
# end func_convert_path_nix_to_w32

# func_convert_path_msys_to_cygwin ARG
# Convert path ARG from MSYS to Cygwin format.  Requires LT_CYGPATH
set.
# Returns result in func_to_host_file_result.
func_convert_path_msys_to_cygwin ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # See func_convert_path_msys_to_w32:
        func_stripname : : "$1"
        func_to_host_path_tmp1=$func_stripname_result
        func_convert_core_msys_to_w32 "$func_to_host_path_tmp1"
        func_cygpath -u -p "$func_convert_core_msys_to_w32_result"
        func_to_host_path_result="$func_cygpath_result"
        func_convert_path_check : : \
            "$func_to_host_path_tmp1" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*:;" : "$1"
    fi
}
# end func_convert_path_msys_to_cygwin

# func_convert_path_nix_to_cygwin ARG
# Convert path ARG from *nix to Cygwin format.  Requires Cygwin
installed in a
# a wine environment, working winepath, and LT_CYGPATH set.  Returns
result in
# func_to_host_file_result.
func_convert_path_nix_to_cygwin ()
{
    $opt_debug
    func_to_host_path_result="$1"
    if test -n "$1"; then
        # Remove leading and trailing path separator characters from
        # ARG. msys behavior is inconsistent here, cygpath turns them
        # into '.;' and ';.', and winepath ignores them completely.
        func_stripname : : "$1"
        func_to_host_path_tmp1=$func_stripname_result
        func_convert_core_path_wine_to_w32 "$func_to_host_path_tmp1"
        func_cygpath -u -p "$func_convert_core_path_wine_to_w32_result"
        func_to_host_path_result="$func_cygpath_result"
        func_convert_path_check : : \
            "$func_to_host_path_tmp1" "$func_to_host_path_result"
        func_convert_path_front_back_pathsep ":*" "*:;" : "$1"
    fi
}

```

```

}
# end func_convert_path_nix_to_cygwin

# func_mode_compile arg...
func_mode_compile ()
{
    $opt_debug
    # Get the compilation command and the source file.
    base_compile=
    srcfile="$nonopt" # always keep a non-empty value in "srcfile"
    suppress_opt=yes
    suppress_output=
    arg_mode=normal
    libobj=
    later=
    pie_flag=

    for arg
    do
        case $arg_mode in
            arg )
                # do not "continue". Instead, add this to base_compile
                lastarg="$arg"
                arg_mode=normal
                ;;

            target )
                libobj="$arg"
                arg_mode=normal
                continue
                ;;

            normal )
                # Accept any command-line options.
                case $arg in
                    -o)
                        test -n "$libobj" && \
                            func_fatal_error "you cannot specify \`-o' more than once"
                        arg_mode=target
                        continue
                        ;;

                    -pie | -fpie | -fPIE)
                        pie_flag+=" $arg"
                        continue
                        ;;

                    -shared | -static | -prefer-pic | -prefer-non-pic)
                        later+=" $arg"
                        continue
                        ;;
                esac
            ;;
        esac
    done
}

```

```

-no-suppress)
    suppress_opt=no
    continue
    ;;

-Xcompiler)
    arg_mode=arg # the next one goes into the "base_compile" arg
list
    continue # The current "srcfile" will either be retained
or
    ;; # replaced later. I would guess that would be a
bug.

-Wc,*)
    func_stripname '-Wc,' '' "$arg"
    args=$func_stripname_result
    lastarg=
    save_ifs="$IFS"; IFS=', '
    for arg in $args; do
        IFS="$save_ifs"
        func_append_quoted lastarg "$arg"
    done
    IFS="$save_ifs"
    func_stripname ' ' '' "$lastarg"
    lastarg=$func_stripname_result

    # Add the arguments to base_compile.
    base_compile+=" $lastarg"
    continue
    ;;

*)
    # Accept the current argument as the source file.
    # The previous "srcfile" becomes the current argument.
    #
    lastarg="$srcfile"
    srcfile="$arg"
    ;;
esac # case $arg
;;
esac # case $arg_mode

# Aesthetically quote the previous argument.
func_append_quoted base_compile "$lastarg"
done # for arg

case $arg_mode in
arg)
    func_fatal_error "you must specify an argument for -Xcompile"
    ;;
target)

```



```

    func_fatal_error "you must specify a target with \`-o'"
    ;;
*)
    # Get the name of the library object.
    test -z "$libobj" && {
        func_basename "$srcfile"
        libobj="$func_basename_result"
    }
    ;;
esac

# Recognize several different file suffixes.
# If the user specifies -o file.o, it is replaced with file.lo
case $libobj in
*.c | *.C | *.f | *.F | *.F90 | *.f90 | *.f95 | *.F95 | \
*.ada | *.adb | *.ads | *.asm | \
*.c++ | *.cc | *.ii | *.class | *.cpp | *.cxx | \
*.c[0-9] | *.for | *.java | *.go | *.obj | *.sx | *.cu | *.cup)
    func_xform "$libobj"
    libobj=$func_xform_result
    ;;
esac

case $libobj in
*.lo) func_lo2o "$libobj"; obj=$func_lo2o_result ;;
*)
    func_fatal_error "cannot determine name of library object from
\`$libobj'"
    ;;
esac

func_infer_tag $base_compile

for arg in $later; do
    case $arg in
    -shared)
        test "$build_libtool_libs" != yes && \
            func_fatal_configuration "can not build a shared library"
        build_old_libs=no
        continue
        ;;

    -static)
        build_libtool_libs=no
        build_old_libs=yes
        continue
        ;;

    -prefer-pic)
        pic_mode=yes
        continue
        ;;

```

```

    -prefer-non-pic)
pic_mode=no
continue
;;
esac
done

func_quote_for_eval "$libobj"
test "X$libobj" != "X$func_quote_for_eval_result" \
    && $ECHO "X$libobj" | $GREP '[~#^*{};<>?'"'"' &()|`$[]' \
    && func_warning "libobj name `$libobj' may not contain shell
special characters."
func_dirname_and_basename "$obj" "/" ""
objname="$func_basename_result"
xdir="$func_dirname_result"
lobj=${xdir}$objdir/$objname

test -z "$base_compile" && \
    func_fatal_help "you must specify a compilation command"

# Delete any leftover library objects.
if test "$build_old_libs" = yes; then
    removelist="$obj $lobj $libobj ${libobj}T"
else
    removelist="$lobj $libobj ${libobj}T"
fi

# On Cygwin there's no "real" PIC flag so we must build both
object types
case $host_os in
cygwin* | mingw* | pw32* | os2* | cegcc*)
    pic_mode=default
    ;;
esac
if test "$pic_mode" = no && test "$deplibs_check_method" !=
pass_all; then
    # non-PIC code in shared libraries is not supported
    pic_mode=default
fi

# Calculate the filename of the output object if compiler does
# not support -o with -c
if test "$compiler_c_o" = no; then
    output_obj=`$ECHO "$srcfile" | $SED 's%^.*/%%;
s%\.[^.]*$%%'\`.$objext}
    lockfile="$output_obj.lock"
else
    output_obj=
    need_locks=no
    lockfile=
fi

```

```

# Lock this critical section if it is needed
# We use this script file to make the link, it avoids creating a
new file
if test "$need_locks" = yes; then
    until $opt_dry_run || ln "$proppath" "$lockfile" 2>/dev/null; do
        func_echo "Waiting for $lockfile to be removed"
        sleep 2
    done
elif test "$need_locks" = warn; then
    if test -f "$lockfile"; then
        $ECHO "\
*** ERROR, $lockfile exists and contains:
`cat $lockfile 2>/dev/null`

```

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```

    $opt_dry_run || $RM $removelist
    exit $EXIT_FAILURE
fi
    removelist+=" $output_obj"
    $ECHO "$srcfile" > "$lockfile"
fi

$opt_dry_run || $RM $removelist
removelist+=" $lockfile"
trap '$opt_dry_run || $RM $removelist; exit $EXIT_FAILURE' 1 2 15

func_to_tool_file "$srcfile" func_convert_file_msys_to_w32
srcfile=$func_to_tool_file_result
func_quote_for_eval "$srcfile"
qsrcfile=$func_quote_for_eval_result

# Only build a PIC object if we are building libtool libraries.
if test "$build_libtool_libs" = yes; then
    # Without this assignment, base_compile gets emptied.
    fbsd_hideous_sh_bug=$base_compile

    if test "$pic_mode" != no; then
        command="$base_compile $qsrcfile $pic_flag"
    else
        # Don't build PIC code
        command="$base_compile $qsrcfile"
    fi

    func_mkdir_p "$xdir$objdir"

```

```

    if test -z "$output_obj"; then
    # Place PIC objects in $objdir
    command+=" -o $lobj"
    fi

    func_show_eval_locale "$command" \
        'test -n "$output_obj" && $RM $removelist; exit
$EXIT_FAILURE'

    if test "$need_locks" = warn &&
        test "X`cat $lockfile 2>/dev/null`" != "X$srcfile"; then
    $ECHO "\
*** ERROR, $lockfile contains:
`cat $lockfile 2>/dev/null`

```

but it should contain:
\$srcfile

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```

    $opt_dry_run || $RM $removelist
    exit $EXIT_FAILURE
    fi

    # Just move the object if needed, then go on to compile the next
one
    if test -n "$output_obj" && test "X$output_obj" != "X$lobj";
then
    func_show_eval '$MV "$output_obj" "$lobj"' \
        'error=$?; $opt_dry_run || $RM $removelist; exit $error'
    fi

    # Allow error messages only from the first compilation.
    if test "$suppress_opt" = yes; then
    suppress_output=' >/dev/null 2>&1'
    fi
    fi

    # Only build a position-dependent object if we build old
libraries.
    if test "$build_old_libs" = yes; then
        if test "$pic_mode" != yes; then
        # Don't build PIC code
        command="$base_compile $qsrcfile$pie_flag"
        else
        command="$base_compile $qsrcfile $pic_flag"
        fi
    fi

```

```

    if test "$compiler_c_o" = yes; then
    command+=" -o $obj"
    fi

    # Suppress compiler output if we already did a PIC compilation.
    command+=" $suppress_output"
    func_show_eval_locale "$command" \
        '$opt_dry_run || $RM $removelist; exit $EXIT_FAILURE'

    if test "$need_locks" = warn &&
        test "X`cat $lockfile 2>/dev/null`" != "X$srcfile"; then
    $ECHO "\
*** ERROR, $lockfile contains:
`cat $lockfile 2>/dev/null`

```

but it should contain:
\$srcfile

This indicates that another process is trying to use the same temporary object file, and libtool could not work around it because your compiler does not support \'-c' and \'-o' together. If you repeat this compilation, it may succeed, by chance, but you had better avoid parallel builds (make -j) in this platform, or get a better compiler."

```

    $opt_dry_run || $RM $removelist
    exit $EXIT_FAILURE
    fi

    # Just move the object if needed
    if test -n "$output_obj" && test "X$output_obj" != "X$obj"; then
    func_show_eval '$MV "$output_obj" "$obj"' \
        'error=$?; $opt_dry_run || $RM $removelist; exit $error'
    fi
    fi

    $opt_dry_run || {
        func_write_libtool_object "$libobj" "$objdir/$objname"
"$objname"

        # Unlock the critical section if it was locked
        if test "$need_locks" != no; then
        removelist=$lockfile
        $RM "$lockfile"
        fi
    }

    exit $EXIT_SUCCESS
}

$opt_help || {
    test "$opt_mode" = compile && func_mode_compile ${1+"$@"}

```

```

}

func_mode_help ()
{
    # We need to display help for each of the modes.
    case $opt_mode in
        "")
            # Generic help is extracted from the usage comments
            # at the start of this file.
            func_help
            ;;

        clean)
            $ECHO \
"Usage: $progname [OPTION]... --mode=clean RM [RM-OPTION]... FILE...

Remove files from the build directory.

RM is the name of the program to use to delete files associated with
each FILE
(typically `'/bin/rm').  RM-OPTIONS are options (such as `-f') to be
passed
to RM.

If FILE is a libtool library, object or program, all the files
associated
with it are deleted. Otherwise, only FILE itself is deleted using RM."
            ;;

        compile)
            $ECHO \
"Usage: $progname [OPTION]... --mode=compile COMPILE-COMMAND...
SOURCEFILE

Compile a source file into a libtool library object.

This mode accepts the following additional options:

    -o OUTPUT-FILE      set the output file name to OUTPUT-FILE
    -no-suppress        do not suppress compiler output for multiple
passes
    -prefer-pic         try to build PIC objects only
    -prefer-non-pic     try to build non-PIC objects only
    -shared             do not build a `'.o' file suitable for static
linking
    -static             only build a `'.o' file suitable for static
linking
    -Wc,FLAG           pass FLAG directly to the compiler

COMPILE-COMMAND is a command to be used in creating a `standard'
object file
from the given SOURCEFILE.

```

The output file name is determined by removing the directory component from SOURCEFILE, then substituting the C source code suffix '\`c' with the library object suffix, '\`lo'."

```
;;  
  
execute)  
$ECHO \  
"Usage: $progname [OPTION]... --mode=execute COMMAND [ARGS]...
```

Automatically set library path, then run a program.

This mode accepts the following additional options:

```
-dlopen FILE      add the directory containing FILE to the library  
path
```

This mode sets the library path environment variable according to '\`-dlopen' flags.

If any of the ARGS are libtool executable wrappers, then they are translated into their corresponding uninstalled binary, and any of their required library directories are added to the library path.

Then, COMMAND is executed, with ARGS as arguments."

```
;;  
  
finish)  
$ECHO \  
"Usage: $progname [OPTION]... --mode=finish [LIBDIR]...
```

Complete the installation of libtool libraries.

Each LIBDIR is a directory that contains libtool libraries.

The commands that this mode executes may require superuser privileges. Use the '\`--dry-run' option if you just want to see what would be executed."

```
;;  
  
install)  
$ECHO \  
"Usage: $progname [OPTION]... --mode=install INSTALL-COMMAND...
```

Install executables or libraries.

INSTALL-COMMAND is the installation command. The first component should be either the `'install'` or `'cp'` program.

The following components of INSTALL-COMMAND are treated specially:

`-inst-prefix-dir PREFIX-DIR` Use PREFIX-DIR as a staging area for installation

The rest of the components are interpreted as arguments to that command (only BSD-compatible install options are recognized)."

`;;`

link)

`$ECHO \`

"Usage: \$progname [OPTION]... --mode=link LINK-COMMAND...

Link object files or libraries together to form another library, or to create an executable program.

LINK-COMMAND is a command using the C compiler that you would use to create a program from several object files.

The following components of LINK-COMMAND are treated specially:

`-all-static` do not do any dynamic linking at all
`-avoid-version` do not add a version suffix if possible
`-bindir BINDIR` specify path to binaries directory (for systems where libraries must be found in the PATH setting at runtime)
`-dlopen FILE` `'-dlpreopen'` FILE if it cannot be dlopened at runtime
`-dlpreopen FILE` link in FILE and add its symbols to `lt_preloaded_symbols`
`-export-dynamic` allow symbols from OUTPUT-FILE to be resolved with `dlsym(3)`
`-export-symbols SYMFILE` try to export only the symbols listed in SYMFILE
`-export-symbols-regex REGEX` try to export only the symbols matching REGEX
`-LLIBDIR` search LIBDIR for required installed libraries
`-lname` OUTPUT-FILE requires the installed library libNAME
`-module` build a library that can dlopened
`-no-fast-install` disable the fast-install mode
`-no-install` link a not-installable executable
`-no-undefined` declare that a library does not refer to external symbols
`-o OUTPUT-FILE` create OUTPUT-FILE from the specified objects


```

-objectlist FILE  Use a list of object files found in FILE to
specify objects
-precious-files-regex REGEX
                        don't remove output files matching REGEX
-release RELEASE  specify package release information
-rpath LIBDIR      the created library will eventually be installed
in LIBDIR
-R[ ]LIBDIR        add LIBDIR to the runtime path of programs and
libraries
-shared            only do dynamic linking of libtool libraries
-shrext SUFFIX     override the standard shared library file
extension
-static            do not do any dynamic linking of uninstalled
libtool libraries
-static-libtool-libs
                        do not do any dynamic linking of libtool libraries
-version-info CURRENT[:REVISION[:AGE]]
                        specify library version info [each variable
defaults to 0]
-weak LIBNAME      declare that the target provides the LIBNAME
interface
-Wc,FLAG           pass linker-specific FLAG directly to the compiler
-Xcompiler FLAG    pass linker-specific FLAG directly to the compiler
-Wl,FLAG           pass linker-specific FLAG directly to the linker
-Xlinker FLAG      pass linker-specific FLAG directly to the linker
-XCc linker FLAG   pass link-specific FLAG to the compiler driver
(CC)

```

All other options (arguments beginning with \`-') are ignored.

Every other argument is treated as a filename. Files ending in \`.la' are treated as uninstalled libtool libraries, other files are standard or library object files.

If the OUTPUT-FILE ends in \`.la', then a libtool library is created, only library objects (\`.lo' files) may be specified, and \`-rpath' is required, except when creating a convenience library.

If OUTPUT-FILE ends in \`.a' or \`.lib', then a standard library is created using \`ar' and \`ranlib', or on Windows using \`.lib'.

If OUTPUT-FILE ends in \`.lo' or \`\${objext}', then a reloadable object file is created, otherwise an executable program is created."

```
;;
```

```

uninstall)
  $ECHO \

```

```
"Usage: $progname [OPTION]... --mode=uninstall RM [RM-OPTION]...
FILE...
```

Remove libraries from an installation directory.

RM is the name of the program to use to delete files associated with each FILE (typically `\/bin/rm`). RM-OPTIONS are options (such as `\/-f`) to be passed to RM.

If FILE is a libtool library, all the files associated with it are deleted.

Otherwise, only FILE itself is deleted using RM."

```
;;

*)
  func_fatal_help "invalid operation mode \`${opt_mode}'"
;;
esac

echo
$ECHO "Try \`${progname} --help' for more information about other
modes."
}
```

Now that we've collected a possible --mode arg, show help if necessary

```
if $opt_help; then
  if test "$opt_help" = :; then
    func_mode_help
  else
    {
      func_help noexit
      for opt_mode in compile link execute install finish uninstall
clean; do
        func_mode_help
      done
    } | sed -n '1p; 2,$s/^Usage:/ or: /p'
    {
      func_help noexit
      for opt_mode in compile link execute install finish uninstall
clean; do
        echo
        func_mode_help
      done
    } |
    sed '1d
      /^When reporting/,/^Report/{
        H
        d
      }
    }
```

```

    $x
    /information about other modes/d
    /more detailed .*MODE/d
    s/^Usage:.*--mode=\([^\ ]*\) .*/Description of \1 mode:/'
fi
exit $?
fi

# func_mode_execute arg...
func_mode_execute ()
{
    $opt_debug
    # The first argument is the command name.
    cmd="$nonopt"
    test -z "$cmd" && \
        func_fatal_help "you must specify a COMMAND"

    # Handle -dlopen flags immediately.
    for file in $opt_dlopen; do
        test -f "$file" \
            || func_fatal_help "`$file' is not a file"

        dir=
        case $file in
            *.la)
                func_resolve_sysroot "$file"
                file=$func_resolve_sysroot_result

                # Check to see that this really is a libtool archive.
                func_lalib_unsafe_p "$file" \
                    || func_fatal_help "`$lib' is not a valid libtool archive"

                # Read the libtool library.
                dlname=
                library_names=
                func_source "$file"

                # Skip this library if it cannot be dlopened.
                if test -z "$dlname"; then
                    # Warn if it was a shared library.
                    test -n "$library_names" && \
                        func_warning "`$file' was not linked with `'-export-
dynamic'"
                    continue
                fi

                func_dirname "$file" "" "."
                dir="$func_dirname_result"

                if test -f "$dir/$objdir/$dlname"; then
                    dir+="/$objdir"
                fi
            fi
        fi

        func_dirname "$file" "" "."
        dir="$func_dirname_result"

        if test -f "$dir/$objdir/$dlname"; then
            dir+="/$objdir"
        fi
    done
}

```

```

else
    if test ! -f "$dir/$dlname"; then
        func_fatal_error "cannot find \"$dlname' in \"$dir' or
\`$dir/$objdir'"
    fi
fi
;;

*.lo)
# Just add the directory containing the .lo file.
func_dirname "$file" "" "."
dir="$func_dirname_result"
;;

*)
func_warning "\`-dlopen' is ignored for non-libtool libraries and
objects"
continue
;;
esac

# Get the absolute pathname.
absdir=`cd "$dir" && pwd`
test -n "$absdir" && dir="$absdir"

# Now add the directory to shlibpath_var.
if eval "test -z \"\${shlibpath_var}\""; then
eval "$shlibpath_var=\"\`$dir\""
else
eval "$shlibpath_var=\"\`$dir:\${shlibpath_var}\""
fi
done

# This variable tells wrapper scripts just to set shlibpath_var
# rather than running their programs.
libtool_execute_magic="$magic"

# Check if any of the arguments is a wrapper script.
args=
for file
do
case $file in
-* | *.la | *.lo ) ;;
*)
# Do a test to see if this is really a libtool program.
if func_ltwrapper_script_p "$file"; then
func_source "$file"
# Transform arg to wrapped name.
file="$progdir/$program"
elif func_ltwrapper_executable_p "$file"; then
func_ltwrapper_scriptname "$file"
func_source "$func_ltwrapper_scriptname_result"

```

```

        # Transform arg to wrapped name.
        file="$progdir/$program"
    fi
;;
esac
# Quote arguments (to preserve shell metacharacters).
func_append_quoted args "$file"
done

if test "X$opt_dry_run" = Xfalse; then
    if test -n "$shlibpath_var"; then
        # Export the shlibpath_var.
        eval "export $shlibpath_var"
    fi

    # Restore saved environment variables
    for lt_var in LANG LANGUAGE LC_ALL LC_CTYPE LC_COLLATE
LC_MESSAGES
    do
        eval "if test \"\${save_$lt_var+set}\" = set; then
            $lt_var=\$save_$lt_var; export $lt_var
        else
            $lt_unset $lt_var
        fi"
    done

    # Now prepare to actually exec the command.
    exec_cmd="\$cmd$args"
else
    # Display what would be done.
    if test -n "$shlibpath_var"; then
        eval "\$ECHO \"\$shlibpath_var=\$$shlibpath_var\""
        echo "export $shlibpath_var"
    fi
    $ECHO "$cmd$args"
    exit $EXIT_SUCCESS
fi
}

test "$opt_mode" = execute && func_mode_execute ${1+"$@"}

# func_mode_finish arg...
func_mode_finish ()
{
    $opt_debug
    libs=
    libdirs=
    admincmds=

    for opt in "$nonopt" ${1+"$@"}
    do

```

```

if test -d "$opt"; then
libdirs+=" $opt"

elif test -f "$opt"; then
if func_lalib_unsafe_p "$opt"; then
libs+=" $opt"
else
func_warning "`$opt' is not a valid libtool archive"
fi

else
func_fatal_error "invalid argument `'$opt'"
fi
done

if test -n "$libs"; then
if test -n "$lt_sysroot"; then
sysroot_regex=`$ECHO "$lt_sysroot" | $SED
"$sed_make_literal_regex"`
sysroot_cmd="s/\(['\])$sysroot_regex/\1/g;"
else
sysroot_cmd=
fi

# Remove sysroot references
if $opt_dry_run; then
for lib in $libs; do
echo "removing references to $lt_sysroot and `=' prefixes
from $lib"
done
else
tmpdir=`func_mkdir`
for lib in $libs; do
sed -e "${sysroot_cmd} s/\(['\]-[LR]\)=/\1/g; s/\(['\])=\1/g"
$lib \
> $tmpdir/tmp-la
mv -f $tmpdir/tmp-la $lib
done
${RM}r "$tmpdir"
fi
fi

if test -n "$finish_cmds$finish_eval" && test -n "$libdirs"; then
for libdir in $libdirs; do
if test -n "$finish_cmds"; then
# Do each command in the finish commands.
func_execute_cmds "$finish_cmds" 'admincmds="$admincmds
'"$cmd"''
fi
if test -n "$finish_eval"; then
# Do the single finish_eval.
eval cmds=\"$finish_eval\"

```

```

    $opt_dry_run || eval "$cmds" || admincmds+="
    $cmds"
fi
done
fi

# Exit here if they wanted silent mode.
$opt_silent && exit $EXIT_SUCCESS

if test -n "$finish_cmds$finish_eval" && test -n "$libdirs"; then
    echo "-----"
    echo "Libraries have been installed in:"
    for libdir in $libdirs; do
        $ECHO "    $libdir"
    done
    echo
    echo "If you ever happen to want to link against installed
libraries"
    echo "in a given directory, LIBDIR, you must either use libtool,
and"
    echo "specify the full pathname of the library, or use the \`-
LLIBDIR'"
    echo "flag during linking and do at least one of the following:"
    if test -n "$shlibpath_var"; then
        echo "    - add LIBDIR to the \`${shlibpath_var}' environment
variable"
        echo "        during execution"
    fi
    if test -n "$runpath_var"; then
        echo "    - add LIBDIR to the \`${runpath_var}' environment
variable"
        echo "        during linking"
    fi
    if test -n "$hardcode_libdir_flag_spec"; then
        libdir=LIBDIR
        eval flag=\`${hardcode_libdir_flag_spec}\`

        $ECHO "    - use the \`${flag}' linker flag"
    fi
    if test -n "$admincmds"; then
        $ECHO "    - have your system administrator run these
commands:$admincmds"
    fi
    if test -f /etc/ld.so.conf; then
        echo "    - have your system administrator add LIBDIR to
\`/etc/ld.so.conf'"
    fi
    echo

    echo "See any operating system documentation about shared
libraries for"

```

```

        case $host in
        solaris2.[6789]|solaris2.1[0-9])
            echo "more information, such as the ld(1), crle(1) and ld.so(8)
manual"
            echo "pages."
            ;;
        *)
            echo "more information, such as the ld(1) and ld.so(8) manual
pages."
            ;;
        esac
        echo "-----"
-----"
    fi
    exit $EXIT_SUCCESS
}

```

```
test "$opt_mode" = finish && func_mode_finish ${1+"$@"}
```

```
# func_mode_install arg...
```

```
func_mode_install ()
```

```
{
    $opt_debug
    # There may be an optional sh(1) argument at the beginning of
    # install_prog (especially on Windows NT).
    if test "$nonopt" = "$SHELL" || test "$nonopt" = /bin/sh ||
        # Allow the use of GNU shtool's install command.
        case $nonopt in *shtool*) ;; *) false;; esac; then
        # Aesthetically quote it.
        func_quote_for_eval "$nonopt"
        install_prog="$func_quote_for_eval_result "
        arg=$1
        shift
    else
        install_prog=
        arg=$nonopt
    fi

    # The real first argument should be the name of the installation
    program.
    # Aesthetically quote it.
    func_quote_for_eval "$arg"
    install_prog+="$func_quote_for_eval_result"
    install_shared_prog=$install_prog
    case " $install_prog " in
        *[\ \ /]cp\ *) install_cp=: ;;
        *) install_cp=false ;;
    esac

    # We need to accept at least all the BSD install flags.
    dest=

```



```

files=
opts=
prev=
install_type=
isdir=no
stripme=
no_mode=:
for arg
do
    arg2=
    if test -n "$dest"; then
files+=" $dest"
dest=$arg
continue
    fi

    case $arg in
    -d) isdir=yes ;;
    -f)
if $install_cp; then ;; else
    prev=$arg
    fi
    ;;
    -g | -m | -o)
prev=$arg
    ;;
    -s)
stripme=" -s"
continue
    ;;
    -*)
    ;;
    *)
# If the previous option needed an argument, then skip it.
if test -n "$prev"; then
    if test "x$prev" = x-m && test -n "$install_override_mode";
then
        arg2=$install_override_mode
        no_mode=false
    fi
    prev=
else
    dest=$arg
    continue
fi
    ;;
    esac

# Aesthetically quote the argument.
func_quote_for_eval "$arg"
install_prog+=" $func_quote_for_eval_result"
if test -n "$arg2"; then

```

```

    func_quote_for_eval "$arg2"
    fi
    install_shared_prog+=" $func_quote_for_eval_result"
done

test -z "$install_prog" && \
    func_fatal_help "you must specify an install program"

test -n "$prev" && \
    func_fatal_help "the '\`$prev' option requires an argument"

if test -n "$install_override_mode" && $no_mode; then
    if $install_cp; then ;; else
        func_quote_for_eval "$install_override_mode"
        install_shared_prog+=" -m $func_quote_for_eval_result"
    fi
fi

if test -z "$files"; then
    if test -z "$dest"; then
        func_fatal_help "no file or destination specified"
    else
        func_fatal_help "you must specify a destination"
    fi
fi

# Strip any trailing slash from the destination.
func_stripname ' ' '/' "$dest"
dest=$func_stripname_result

# Check to see that the destination is a directory.
test -d "$dest" && isdir=yes
if test "$isdir" = yes; then
    destdir="$dest"
    destname=
else
    func_dirname_and_basename "$dest" "" "."
    destdir="$func_dirname_result"
    destname="$func_basename_result"

    # Not a directory, so check to see that there is only one file
    specified.
    set dummy $files; shift
    test "$#" -gt 1 && \
        func_fatal_help "\`$dest' is not a directory"
fi
case $destdir in
[\\/] * | [A-Za-z]:[\\/] *) ;;
*)
    for file in $files; do
        case $file in
*.lo) ;;

```

```

*)
  func_fatal_help "\`$destdir' must be an absolute directory
name"
  ;;
esac
done
;;
esac

# This variable tells wrapper scripts just to set variables rather
# than running their programs.
libtool_install_magic="$magic"

staticlibs=
future_libdirs=
current_libdirs=
for file in $files; do

  # Do each installation.
  case $file in
  *.$libext)
    # Do the static libraries later.
    staticlibs+="$ file"
    ;;

  *.la)
    func_resolve_sysroot "$file"
    file=$func_resolve_sysroot_result

    # Check to see that this really is a libtool archive.
    func_lalib_unsafe_p "$file" \
      || func_fatal_help "\`$file' is not a valid libtool archive"

    library_names=
    old_library=
    relink_command=
    func_source "$file"

    # Add the libdir to current_libdirs if it is the destination.
    if test "X$destdir" = "X$libdir"; then
      case "$current_libdirs " in
      *" $libdir ") ;;
      *) current_libdirs+="$ libdir" ;;
      esac
    else
      # Note the libdir as a future libdir.
      case "$future_libdirs " in
      *" $libdir ") ;;
      *) future_libdirs+="$ libdir" ;;
      esac
    fi
  fi

```

```

func_dirname "$file" "/" ""
dir="$func_dirname_result"
dir+="$objdir"

if test "$fast_install" = no && test -n "$relink_command"; then
# Strip any trailing slash from the destination.
func_stripname '' '/' "$libdir"
destlibdir=$func_stripname_result

func_stripname '' '/' "$destdir"
s_destdir=$func_stripname_result

# Determine the prefix the user has applied to our future dir.
inst_prefix_dir=`$ECHO "X$s_destdir" | $Xsed -e
"s%$destlibdir\$%"`

# Don't allow the user to place us outside of our expected
# location b/c this prevents finding dependent libraries that
# are installed to the same prefix.
# At present, this check doesn't affect windows .dll's that
# are installed into $libdir/./bin (currently, that works
fine)
# but it's something to keep an eye on.
test "$inst_prefix_dir" = "$destdir" && \
func_fatal_error "error: cannot install `'$file'` to a
directory not ending in $libdir"

if test -n "$inst_prefix_dir"; then
# Stick the inst_prefix_dir data into the link command.
relink_command=`$ECHO "$relink_command" | $SED
"s%@inst_prefix_dir@%-inst-prefix-dir $inst_prefix_dir%"`
else
relink_command=`$ECHO "$relink_command" | $SED
"s%@inst_prefix_dir@%"`
fi

func_warning "relinking `'$file'`"
func_show_eval "$relink_command" \
'func_fatal_error "error: relink `'$file'` with the above
command before installing it"'
fi

# See the names of the shared library.
set dummy $library_names; shift
if test -n "$1"; then
realname="$1"
shift

srcname="$realname"
test "$fast_install" = no && test -n "$relink_command" &&
srcname="$realname"

```

```

    # Install the shared library and build the symlinks.
    func_show_eval "$install_shared_prog $dir/$srcname
$destdir/$realname" \
        'exit $?'
    tstripme="$stripme"
    case $host_os in
    cygwin* | mingw* | pw32* | cegcc*)
        case $realname in
        *.dll.a)
            tstripme=""
            ;;
        esac
        ;;
    esac
    if test -n "$tstripme" && test -n "$striplib"; then
        func_show_eval "$striplib $destdir/$realname" 'exit $?'
    fi

    if test "$#" -gt 0; then
        # Delete the old symlinks, and create new ones.
        # Try `ln -sf' first, because the `ln' binary might depend on
        # the symlink we replace! Solaris /bin/ln does not
understand -f,
        # so we also need to try rm && ln -s.
        for linkname
        do
            test "$linkname" != "$realname" \
                && func_show_eval "(cd $destdir && { $LN_S -f $realname
$linkname || { $RM $linkname && $LN_S $realname $linkname; }; })"
        done
    fi

    # Do each command in the postinstall commands.
    lib="$destdir/$realname"
    func_execute_cmds "$postinstall_cmds" 'exit $?'
fi

# Install the pseudo-library for information purposes.
func_basename "$file"
name="$func_basename_result"
instname="$dir/$name"i
func_show_eval "$install_prog $instname $destdir/$name" 'exit $?'

# Maybe install the static library, too.
test -n "$old_library" && staticlibs+=" $dir/$old_library"
;;

*.lo)
# Install (i.e. copy) a libtool object.

# Figure out destination file name, if it wasn't already
specified.

```

```

if test -n "$destname"; then
    destfile="$destdir/$destname"
else
    func_basename "$file"
    destfile="$func_basename_result"
    destfile="$destdir/$destfile"
fi

# Deduce the name of the destination old-style object file.
case $destfile in
*.lo)
    func_lo2o "$destfile"
    staticdest=$func_lo2o_result
    ;;
*.$objext)
    staticdest="$destfile"
    destfile=
    ;;
*)
    func_fatal_help "cannot copy a libtool object to `\$destfile'"
    ;;
esac

# Install the libtool object if requested.
test -n "$destfile" && \
    func_show_eval "$install_prog $file $destfile" 'exit $?'

# Install the old object if enabled.
if test "$build_old_libs" = yes; then
    # Deduce the name of the old-style object file.
    func_lo2o "$file"
    staticobj=$func_lo2o_result
    func_show_eval "$install_prog \$staticobj \$staticdest" 'exit
$?'
fi
exit $EXIT_SUCCESS
;;

*)
# Figure out destination file name, if it wasn't already
specified.
if test -n "$destname"; then
    destfile="$destdir/$destname"
else
    func_basename "$file"
    destfile="$func_basename_result"
    destfile="$destdir/$destfile"
fi

# If the file is missing, and there is a .exe on the end, strip
it
# because it is most likely a libtool script we actually want to

```

```

# install
stripped_ext=""
case $file in
*.exe)
    if test ! -f "$file"; then
        func_stripname ' ' '.exe' "$file"
        file=$func_stripname_result
        stripped_ext=".exe"
    fi
    ;;
esac

# Do a test to see if this is really a libtool program.
case $host in
*cygwin* | *mingw*)
    if func_ltwrapper_executable_p "$file"; then
        func_ltwrapper_scriptname "$file"
        wrapper=$func_ltwrapper_scriptname_result
    else
        func_stripname ' ' '.exe' "$file"
        wrapper=$func_stripname_result
    fi
    ;;
*)
    wrapper=$file
    ;;
esac
if func_ltwrapper_script_p "$wrapper"; then
    notinst_deplibs=
    relink_command=

    func_source "$wrapper"

    # Check the variables that should have been set.
    test -z "$generated_by_libtool_version" && \
        func_fatal_error "invalid libtool wrapper script \`$wrapper'"

    finalize=yes
    for lib in $notinst_deplibs; do
        # Check to see that each library is installed.
        libdir=
        if test -f "$lib"; then
            func_source "$lib"
        fi
        libfile="$libdir/"`$ECHO "$lib" | $SED 's%^.*/%%g'`###
testsuite: skip nested quoting test
        if test -n "$libdir" && test ! -f "$libfile"; then
            func_warning "`$lib' has not been installed in \`$libdir'"
            finalize=no
        fi
    done

```

```

relink_command=
func_source "$wrapper"

outputname=
if test "$fast_install" = no && test -n "$relink_command"; then
  $opt_dry_run || {
    if test "$finalize" = yes; then
      tmpdir=`func_mkdir`
      func_basename "$file$stripped_ext"
      file="$func_basename_result"
      outputname="$tmpdir/$file"
      # Replace the output file specification.
      relink_command=`ECHO "$relink_command" | $SED
's%@OUTPUT@%' "$outputname" '%g'`

      $opt_silent || {
        func_quote_for_expand "$relink_command"
        eval "func_echo $func_quote_for_expand_result"
      }
      if eval "$relink_command"; then :
        else
          func_error "error: relink ``$file' with the above command
before installing it"
          $opt_dry_run || ${RM}r "$tmpdir"
          continue
        fi
        file="$outputname"
      else
        func_warning "cannot relink ``$file'"
      fi
    }
  else
    # Install the binary that we compiled earlier.
    file=`ECHO "$file$stripped_ext" | $SED
"s%\([^/]*\)%%$objdir/\1%"`
    fi
  fi

# remove .exe since cygwin /usr/bin/install will append another
# one anyway
case $install_prog,$host in
*/usr/bin/install*,*cygwin*)
  case $file:$destfile in
*.exe:*.exe)
  # this is ok
  ;;
*.exe:*)
  destfile=$destfile.exe
  ;;
*:*.*.exe)
  func_stripname '' '.exe' "$destfile"
  destfile=$func_stripname_result

```



```

        ;;
    esac
    ;;
esac
func_show_eval "$install_prog\$stripme \$file \$destfile" 'exit
$?'
$opt_dry_run || if test -n "$outputname"; then
    ${RM}r "$tmpdir"
fi
;;
esac
done

for file in $staticlibs; do
    func_basename "$file"
    name="$func_basename_result"

    # Set up the ranlib parameters.
    oldlib="$destdir/$name"
    func_to_tool_file "$oldlib" func_convert_file_msys_to_w32
    tool_oldlib=$func_to_tool_file_result

    func_show_eval "$install_prog \$file \$oldlib" 'exit $?'

    if test -n "$stripme" && test -n "$old_striplib"; then
        func_show_eval "$old_striplib $tool_oldlib" 'exit $?'
    fi

    # Do each command in the postinstall commands.
    func_execute_cmds "$old_postinstall_cmds" 'exit $?'
done

test -n "$future_libdirs" && \
    func_warning "remember to run \`${progname} --
finish$future_libdirs'"

if test -n "$current_libdirs"; then
    # Maybe just do a dry run.
    $opt_dry_run && current_libdirs=" -n$current_libdirs"
    exec_cmd='$SHELL $proppath $preserve_args --
finish$current_libdirs'
else
    exit $EXIT_SUCCESS
fi
}

test "$opt_mode" = install && func_mode_install ${1+"$@"}

# func_generate_dlsyms outputname originator pic_p
# Extract symbols from dlprefiles and create ${outputname}.S.o with
# a dlpreopen symbol table.

```

```

func_generate_dlsyms ()
{
    $opt_debug
    my_outputname="$1"
    my_originator="$2"
    my_pic_p="{3-no}"
    my_prefix=`$ECHO "$my_originator" | sed 's%[^a-zA-Z0-9]%%_g'`
    my_dlsyms=

    if test -n "$dlfiles$dlprefiles" || test "$dlself" != no; then
        if test -n "$NM" && test -n "$global_symbol_pipe"; then
            my_dlsyms="{my_outputname}.S.c"
        else
            func_error "not configured to extract global symbols from
dlpreopened files"
        fi
    fi

    if test -n "$my_dlsyms"; then
        case $my_dlsyms in
            "") ;;
            *.c)
                # Discover the nlist of each of the dlfiles.
                nlist="$output_objdir/${my_outputname}.nm"

                func_show_eval "$RM $nlist ${nlist}S ${nlist}T"

                # Parse the name list into a source file.
                func_verbose "creating $output_objdir/$my_dlsyms"

                $opt_dry_run || $ECHO > "$output_objdir/$my_dlsyms" "\
/* $my_dlsyms - symbol resolution table for \`$my_outputname' dlsym
emulation. */
/* Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION */

#ifdef __cplusplus
extern \"C\" {
#endif

#ifdef __GNUC__ && (((__GNUC__ == 4) && (__GNUC_MINOR__ >= 4))
|| (__GNUC__ > 4))
#pragma GCC diagnostic ignored \"-Wstrict-prototypes\"
#endif

/* Keep this code in sync between libtool.m4, ltmain, lt_system.h, and
tests. */
#ifdef __WIN32__ || defined(__CYGWIN__) || defined(_WIN32_WCE)
/* DATA imports from DLLs on WIN32 can't be const, because runtime
relocations are performed -- see ld's documentation on pseudo-
relocs. */
# define LT_DLSYM_CONST
#elif defined(__osf__)

```

```

/* This system does not cope well with relocations in const data. */
# define LT_DLSYM_CONST
#else
# define LT_DLSYM_CONST const
#endif

/* External symbol declarations for the compiler. */\
"

    if test "$dlsself" = yes; then
        func_verbose "generating symbol list for \`${output}'"

        $opt_dry_run || echo ': @PROGRAM@ ' > "$nlist"

        # Add our own program objects to the symbol list.
        progfiles=`$ECHO "$objs$old_deplibs" | $SP2NL | $SED "$lo2o" |
$NL2SP`
        for progfile in $progfiles; do
            func_to_tool_file "$progfile" func_convert_file_msys_to_w32
            func_verbose "extracting global C symbols from
\`${func_to_tool_file_result}'"
            $opt_dry_run || eval "$NM $func_to_tool_file_result |
$global_symbol_pipe >> '$nlist'"
            done

            if test -n "$exclude_expsyms"; then
                $opt_dry_run || {
                    eval '$EGREP -v " ($exclude_expsyms)$" "$nlist" >
"$nlist.T"
                    eval '$MV "$nlist.T" "$nlist"'
                }
            fi

            if test -n "$export_symbols_regex"; then
                $opt_dry_run || {
                    eval '$EGREP -e "$export_symbols_regex" "$nlist" >
"$nlist.T"
                    eval '$MV "$nlist.T" "$nlist"'
                }
            fi

            # Prepare the list of exported symbols
            if test -z "$export_symbols"; then
                export_symbols="$output_objdir/$outputname.exp"
                $opt_dry_run || {
                    $RM $export_symbols
                    eval "${SED} -n -e '/^: @PROGRAM@ \$/d' -e 's/^.*
\(.*\)\$/\1/p' "' < "$nlist" > "$export_symbols"'
                    case $host in
                        *cygwin* | *mingw* | *cegcc* )
                            eval "echo EXPORTS "' >
"$output_objdir/$outputname.def"'

```

```

        eval 'cat "$export_symbols" >>
"$output_objdir/$outputname.def"
        ;;
    esac
}
else
    $opt_dry_run || {
        eval "${SED} -e 's/\([[:^$]]\)/\\\\\\\\1/g' -e 's/^/ /' -e
's/$/$/'" < "$export_symbols" > "$output_objdir/$outputname.exp"
        eval '$GREP -f "$output_objdir/$outputname.exp" < "$nlist"
> "$nlist"T'
        eval '$MV "$nlist"T "$nlist"'
        case $host in
            *cygwin* | *mingw* | *cegcc* )
                eval "echo EXPORTS "'>
"$output_objdir/$outputname.def"
                eval 'cat "$nlist" >> "$output_objdir/$outputname.def"'
            ;;
        esac
    }
fi
fi

for dlprefile in $dlprefiles; do
    func_verbose "extracting global C symbols from `"$dlprefile`"
    func_basename "$dlprefile"
    name="$func_basename_result"
    case $host in
        *cygwin* | *mingw* | *cegcc* )
            # if an import library, we need to obtain dlname
            if func_win32_import_lib_p "$dlprefile"; then
                func_tr_sh "$dlprefile"
                eval "curr_lafile=\$libfile_$func_tr_sh_result"
                dlprefile_dlbasename=""
                if test -n "$curr_lafile" && func_lalib_p "$curr_lafile";
then
                    # Use subshell, to avoid clobbering current variable
values
                    dlprefile_dlname=`source "$curr_lafile" && echo
"$dlname"`
                    if test -n "$dlprefile_dlname" ; then
                        func_basename "$dlprefile_dlname"
                        dlprefile_dlbasename="$func_basename_result"
                    else
                        # no lafile. user explicitly requested -dlpreopen
<import library>.
                        $sharedlib_from_linklib_cmd "$dlprefile"
                        dlprefile_dlbasename=$sharedlib_from_linklib_result
                    fi
                fi
            fi
            $opt_dry_run || {
                if test -n "$dlprefile_dlbasename" ; then

```

```

        eval '$ECHO ": $dlprefile_dlbasename" >> "$nlist"'
    else
        func_warning "Could not compute DLL name from $name"
        eval '$ECHO ": $name " >> "$nlist"'
    fi
    func_to_tool_file "$dlprefile"
func_convert_file_msys_to_w32
    eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe |
    $SED -e '/I __imp/d' -e 's/I __nm_/D /;s/_nm_/' >>
'$nlist'"
    }
    else # not an import lib
        $opt_dry_run || {
            eval '$ECHO ": $name " >> "$nlist"'
            func_to_tool_file "$dlprefile"
func_convert_file_msys_to_w32
            eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe >> '$nlist'"
        }
    fi
;;
*)
    $opt_dry_run || {
        eval '$ECHO ": $name " >> "$nlist"'
        func_to_tool_file "$dlprefile"
func_convert_file_msys_to_w32
        eval "$NM \"\$func_to_tool_file_result\" 2>/dev/null |
$global_symbol_pipe >> '$nlist'"
    }
;;
esac
done

$opt_dry_run || {
# Make sure we have at least an empty file.
test -f "$nlist" || : > "$nlist"

if test -n "$exclude_expsyms"; then
    $EGREP -v " ($exclude_expsyms)$" "$nlist" > "$nlist.T"
    $MV "$nlist.T" "$nlist"
fi

# Try sorting and uniquifying the output.
if $GREP -v "^: " < "$nlist" |
    if sort -k 3 </dev/null >/dev/null 2>&1; then
        sort -k 3
    else
        sort +2
    fi |
    uniq > "$nlist.S"; then
:

```

```

else
    $GREP -v "^: " < "$nlist" > "$nlist"S
fi

if test -f "$nlist"S; then
    eval "$global_symbol_to_cdecl" < "$nlist"S >>
"$output_objdir/$my_dlsyms"
else
    echo '/* NONE */' >> "$output_objdir/$my_dlsyms"
fi

echo >> "$output_objdir/$my_dlsyms" "\

/* The mapping between symbol names and symbols. */
typedef struct {
    const char *name;
    void *address;
} lt_dlsymlist;
extern LT_DLSYM_CONST lt_dlsymlist
lt_${my_prefix}_LTX_preloaded_symbols[];
LT_DLSYM_CONST lt_dlsymlist
lt_${my_prefix}_LTX_preloaded_symbols[] =
{\
    { \ "$my_originator\ ", (void *) 0 },"

    case $need_lib_prefix in
no)
    eval "$global_symbol_to_c_name_address" < "$nlist" >>
"$output_objdir/$my_dlsyms"
    ;;
*)
    eval "$global_symbol_to_c_name_address_lib_prefix" < "$nlist"
>> "$output_objdir/$my_dlsyms"
    ;;
esac
    echo >> "$output_objdir/$my_dlsyms" "\
{0, (void *) 0}
};

/* This works around a problem in FreeBSD linker */
#ifdef FREEBSD_WORKAROUND
static const void *lt_preloaded_setup() {
    return lt_${my_prefix}_LTX_preloaded_symbols;
}
#endif

#ifdef __cplusplus
}
#endif\
"

    } # !$opt_dry_run

```

```

pic_flag_for_syntable=
case "$compile_command " in
*" -static ") ;;
*)
  case $host in
  # compiling the symbol table file with pic_flag works around
  # a FreeBSD bug that causes programs to crash when -lm is
  # linked before any other PIC object. But we must not use
  # pic_flag when linking with -static. The problem exists in
  # FreeBSD 2.2.6 and is fixed in FreeBSD 3.1.
  *-*-freebsd2.*|*-*-freebsd3.0*|*-*-freebsdelf3.0*)
    pic_flag_for_syntable=" $pic_flag -DFREEBSD_WORKAROUND" ;;
  *-*-hpux*)
    pic_flag_for_syntable=" $pic_flag" ;;
  *)
    if test "X$my_pic_p" != Xno; then
      pic_flag_for_syntable=" $pic_flag"
    fi
    ;;
  esac
  ;;
esac
symtab_cflags=
for arg in $LTCFLAGS; do
  case $arg in
  -pie | -fpie | -fPIE) ;;
  *) symtab_cflags+=" $arg" ;;
  esac
done

# Now compile the dynamic symbol file.
func_show_eval '(cd $output_objdir && $LTCC$symtab_cflags -
c$no_builtin_flag$pic_flag_for_syntable "$my_dlsyms")' 'exit $?'

# Clean up the generated files.
func_show_eval '$RM "$output_objdir/$my_dlsyms" "$nlist"
"${nlist}S" "${nlist}T"'

# Transform the symbol file into the correct name.
symfileobj="$output_objdir/${my_outputname}S.$objext"
case $host in
*cygwin* | *mingw* | *cegcc* )
  if test -f "$output_objdir/$my_outputname.def"; then
    compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$output_objdir/$my_outputname.def $symfileobj%"`
    finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$output_objdir/$my_outputname.def $symfileobj%"`
  else
    compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
    finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
  fi

```

```

        fi
        ;;
    *)
        compile_command=`$ECHO "$compile_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
        finalize_command=`$ECHO "$finalize_command" | $SED
"s%@SYMFIL@%$symfileobj%"`
        ;;
    esac
    ;;
    *)
        func_fatal_error "unknown suffix for \`${my_dlsyms}'"
    ;;
    esac
else
    # We keep going just in case the user didn't refer to
    # lt_preloaded_symbols. The linker will fail if
global_symbol_pipe
    # really was required.

    # Nullify the symbol file.
    compile_command=`$ECHO "$compile_command" | $SED "s%
@SYMFIL@%%"`
    finalize_command=`$ECHO "$finalize_command" | $SED "s%
@SYMFIL@%%"`
    fi
}

# func_win32_libid arg
# return the library type of file 'arg'
#
# Need a lot of goo to handle *both* DLLs and import libs
# Has to be a shell function in order to 'eat' the argument
# that is supplied when $file_magic_command is called.
# Despite the name, also deal with 64 bit binaries.
func_win32_libid ()
{
    $opt_debug
    win32_libid_type="unknown"
    win32_fileres=`file -L $1 2>/dev/null`
    case $win32_fileres in
    *ar\ archive\ import\ library*) # definitely import
        win32_libid_type="x86 archive import"
        ;;
    *ar\ archive*) # could be an import, or static
        # Keep the egrep pattern in sync with the one in
_LT_CHECK_MAGIC_METHOD.
        if eval $OBJDUMP -f $1 | $SED -e '10q' 2>/dev/null |
            $EGREP 'file format (pei*-i386(.?architecture: i386)?|pe-arm-
wince|pe-x86-64)' >/dev/null; then
            func_to_tool_file "$1" func_convert_file_msys_to_w32

```



```

win32_nmres=`eval $NM -f posix -A \"\$func_to_tool_file_result\"
|
$SED -n -e '
    1,100{
        / I /{
            s,.*,import,
            p
            q
        }
    }'`
case $win32_nmres in
import*) win32_libid_type="x86 archive import";;
*) win32_libid_type="x86 archive static";;
esac
fi
;;
*DLL*)
win32_libid_type="x86 DLL"
;;
*executable*) # but shell scripts are "executable" too...
case $win32_fileres in
*MS\ Windows\ PE\ Intel*)
win32_libid_type="x86 DLL"
;;
esac
;;
esac
$ECHO "$win32_libid_type"
}

# func_cygming_dll_for_implib ARG
#
# Platform-specific function to extract the
# name of the DLL associated with the specified
# import library ARG.
# Invoked by eval'ing the libtool variable
# $sharedlib_from_linklib_cmd
# Result is available in the variable
# $sharedlib_from_linklib_result
func_cygming_dll_for_implib ()
{
    $opt_debug
    sharedlib_from_linklib_result=`$DLLTOOL --identify-strict --identify
"$1"`
}

# func_cygming_dll_for_implib_fallback_core SECTION_NAME LIBNAMEs
#
# This is the core of a fallback implementation of a
# platform-specific function to extract the name of the
# DLL associated with the specified import library LIBNAME.
#

```

```

# SECTION_NAME is either .idata$6 or .idata$7, depending
# on the platform and compiler that created the implib.
#
# Echoes the name of the DLL associated with the
# specified import library.
func_cygmimg_dll_for_implib_fallback_core ()
{
    $opt_debug
    match_literal=`$ECHO "$1" | $SED "$sed_make_literal_regex"`
    $OBJDUMP -s --section "$1" "$2" 2>/dev/null |
        $SED '/^Contents of section "$match_literal":/{
            # Place marker at beginning of archive member dllname section
            s/./====MARK====/
            p
            d
        }
        # These lines can sometimes be longer than 43 characters, but
        # are always uninteresting
        /:[ ]*file format pe[i]\{,1\}-/d
        /^In archive [^:]*:/d
        # Ensure marker is printed
        /^====MARK====/p
        # Remove all lines with less than 43 characters
        /^.\{43\}/!d
        # From remaining lines, remove first 43 characters
        s/^.\{43\}//' |
        $SED -n '
            # Join marker and all lines until next marker into a single line
            /^====MARK====/ b para
            H
            $ b para
            b
            :para
            x
            s/\n//g
            # Remove the marker
            s/^====MARK====//
            # Remove trailing dots and whitespace
            s/[\. \t]*$//
            # Print
            /./p' |
        # we now have a list, one entry per line, of the stringified
        # contents of the appropriate section of all members of the
        # archive which possess that section. Heuristic: eliminate
        # all those which have a first or second character that is
        # a '.' (that is, objdump's representation of an unprintable
        # character.) This should work for all archives with less than
        # 0x302f exports -- but will fail for DLLs whose name actually
        # begins with a literal '.' or a single character followed by
        # a '.'.
        #
        # Of those that remain, print the first one.

```

```

    $SED -e '/^\./d;/^\./d;q'
}

# func_cygming_gnu_implib_p ARG
# This predicate returns with zero status (TRUE) if
# ARG is a GNU/binutils-style import library. Returns
# with nonzero status (FALSE) otherwise.
func_cygming_gnu_implib_p ()
{
    $opt_debug
    func_to_tool_file "$1" func_convert_file_msys_to_w32
    func_cygming_gnu_implib_tmp=`$NM "$func_to_tool_file_result" | eval
"$global_symbol_pipe" | $EGREP ' (_head_[A-Za-z0-9_]+_[ad]l*|[A-Za-z0-
9_]+_[ad]l*_iname)$'`
    test -n "$func_cygming_gnu_implib_tmp"
}

# func_cygming_ms_implib_p ARG
# This predicate returns with zero status (TRUE) if
# ARG is an MS-style import library. Returns
# with nonzero status (FALSE) otherwise.
func_cygming_ms_implib_p ()
{
    $opt_debug
    func_to_tool_file "$1" func_convert_file_msys_to_w32
    func_cygming_ms_implib_tmp=`$NM "$func_to_tool_file_result" | eval
"$global_symbol_pipe" | $GREP '_NULL_IMPORT_DESCRIPTOR'`
    test -n "$func_cygming_ms_implib_tmp"
}

# func_cygming_dll_for_implib_fallback ARG
# Platform-specific function to extract the
# name of the DLL associated with the specified
# import library ARG.
#
# This fallback implementation is for use when $DLLTOOL
# does not support the --identify-strict option.
# Invoked by eval'ing the libtool variable
#   $sharedlib_from_linklib_cmd
# Result is available in the variable
#   $sharedlib_from_linklib_result
func_cygming_dll_for_implib_fallback ()
{
    $opt_debug
    if func_cygming_gnu_implib_p "$1" ; then
        # binutils import library

sharedlib_from_linklib_result=`func_cygming_dll_for_implib_fallback_co
re '.idata$7' "$1"`
    elif func_cygming_ms_implib_p "$1" ; then
        # ms-generated import library

```

```

sharedlib_from_linklib_result=`func_cygmimg_dll_for_implib_fallback_co
re '.idata$6' "$1"`
else
    # unknown
    sharedlib_from_linklib_result=""
fi
}

# func_extract_an_archive dir oldlib
func_extract_an_archive ()
{
    $opt_debug
    f_ex_an_ar_dir="$1"; shift
    f_ex_an_ar_oldlib="$1"
    if test "$lock_old_archive_extraction" = yes; then
        lockfile=$f_ex_an_ar_oldlib.lock
        until $opt_dry_run || ln "$proppath" "$lockfile" 2>/dev/null; do
            func_echo "Waiting for $lockfile to be removed"
            sleep 2
        done
    fi
    func_show_eval "(cd \"$f_ex_an_ar_dir\" && $AR x
\"$f_ex_an_ar_oldlib\")" \
        'stat=$?; rm -f "$lockfile"; exit $stat'
    if test "$lock_old_archive_extraction" = yes; then
        $opt_dry_run || rm -f "$lockfile"
    fi
    if ($AR t "$f_ex_an_ar_oldlib" | sort | sort -uc >/dev/null 2>&1);
then
    :
else
    func_fatal_error "object name conflicts in archive:
$f_ex_an_ar_dir/$f_ex_an_ar_oldlib"
fi
}

# func_extract_archives gentop oldlib ...
func_extract_archives ()
{
    $opt_debug
    my_gentop="$1"; shift
    my_oldlibs=${1+"$@"}
    my_oldobjs=""
    my_xlib=""
    my_xabs=""
    my_xdir=""

    for my_xlib in $my_oldlibs; do
        # Extract the objects.

```

```

case $my_xlib in
[\\/] * | [A-Za-z]:[\\/]*) my_xabs="$my_xlib" ;;
*) my_xabs=`pwd`"/$my_xlib" ;;
esac
func_basename "$my_xlib"
my_xlib="$func_basename_result"
my_xlib_u=$my_xlib
while ;; do
  case " $extracted_archives " in
*" $my_xlib_u "*)
  func_arith $extracted_serial + 1
  extracted_serial=$func_arith_result
  my_xlib_u=lt$extracted_serial-$my_xlib ;;
*) break ;;
esac
done
extracted_archives="$extracted_archives $my_xlib_u"
my_xdir="$my_gentop/$my_xlib_u"

func_mkdir_p "$my_xdir"

case $host in
*-darwin*)
func_verbose "Extracting $my_xabs"
# Do not bother doing anything if just a dry run
$opt_dry_run || {
  darwin_orig_dir=`pwd`
  cd $my_xdir || exit $?
  darwin_archive=$my_xabs
  darwin_curdir=`pwd`
  darwin_base_archive=`basename "$darwin_archive"`
  darwin_arches=`$LIPO -info "$darwin_archive" 2>/dev/null |
$GREP Architectures 2>/dev/null || true`
  if test -n "$darwin_arches"; then
    darwin_arches=`$ECHO "$darwin_arches" | $SED -e 's/.*are:/'`
    darwin_arch=
    func_verbose "$darwin_base_archive has multiple architectures
$darwin_arches"
    for darwin_arch in $darwin_arches ; do
      func_mkdir_p "unfat-$$/${darwin_base_archive}-
${darwin_arch}"
      $LIPO -thin $darwin_arch -output "unfat-
$$/${darwin_base_archive}-${darwin_arch}/${darwin_base_archive}"
      "${darwin_archive}"
      cd "unfat-$$/${darwin_base_archive}-${darwin_arch}"
      func_extract_an_archive "`pwd`" "${darwin_base_archive}"
      cd "$darwin_curdir"
      $RM "unfat-$$/${darwin_base_archive}-
${darwin_arch}/${darwin_base_archive}"
    done # $darwin_arches
    ## Okay now we've a bunch of thin objects, gotta fatten
them up :)

```

```

        darwin_filelist=`find unfat-$$ -type f -name \*.o -print -o -
name \*.lo -print | $SED -e "$basename" | sort -u`
        darwin_file=
        darwin_files=
        for darwin_file in $darwin_filelist; do
            darwin_files=`find unfat-$$ -name $darwin_file -print |
sort | $NL2SP`
            $LIPO -create -output "$darwin_file" $darwin_files
            done # $darwin_filelist
            $RM -rf unfat-$$
            cd "$darwin_orig_dir"
        else
            cd $darwin_orig_dir
            func_extract_an_archive "$my_xdir" "$my_xabs"
            fi # $darwin_arches
        } # !$opt_dry_run
        ;;
        *)
            func_extract_an_archive "$my_xdir" "$my_xabs"
            ;;
        esac
        my_oldobjs="$my_oldobjs "`find $my_xdir -name \*.$objext -print
-o -name \*.lo -print | sort | $NL2SP`
        done

        func_extract_archives_result="$my_oldobjs"
    }

# func_emit_wrapper [arg=no]
#
# Emit a libtool wrapper script on stdout.
# Don't directly open a file because we may want to
# incorporate the script contents within a cygwin/mingw
# wrapper executable.  Must ONLY be called from within
# func_mode_link because it depends on a number of variables
# set therein.
#
# ARG is the value that the WRAPPER_SCRIPT_BELONGS_IN_OBJDIR
# variable will take.  If 'yes', then the emitted script
# will assume that the directory in which it is stored is
# the $objdir directory.  This is a cygwin/mingw-specific
# behavior.
func_emit_wrapper ()
{
    func_emit_wrapper_arg1=${1-no}

    $ECHO "\
#! $SHELL

# $output - temporary wrapper script for $objdir/$outputname
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION

```

```

#
# The $output program cannot be directly executed until all the
libtool
# libraries that it depends on are installed.
#
# This wrapper script should never be moved out of the build
directory.
# If it is, it will not operate correctly.

# Sed substitution that helps us do robust quoting. It backslashifies
# metacharacters that are still active within double-quoted strings.
sed_quote_subst='$sed_quote_subst'

# Be Bourne compatible
if test -n \"\${ZSH_VERSION+set}\" && (emulate sh) >/dev/null 2>&1;
then
  emulate sh
  NULLCMD=:
  # Zsh 3.x and 4.x performs word splitting on \"${1+\"$@}\" , which
  # is contrary to our usage. Disable this feature.
  alias -g \"\${1+\"$@}\"='\"$@\"'
  setopt NO_GLOB_SUBST
else
  case \"(set -o) 2>/dev/null\" in *posix*) set -o posix;; esac
fi
BIN_SH=xpg4; export BIN_SH # for Tru64
DUALCASE=1; export DUALCASE # for MKS sh

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

relink_command=\"\${relink_command}\"

# This environment variable determines our operation mode.
if test \"\${libtool_install_magic}\" = \"\${magic}\"; then
  # install mode needs the following variables:
  generated_by_libtool_version='\${macro_version}'
  notinst_deplibs='\${notinst_deplibs}'
else
  # When we are sourced in execute mode, \${file} and \${ECHO} are already
  set.
  if test \"\${libtool_execute_magic}\" != \"\${magic}\"; then
    file=\"\${0}\"

    qECHO=`ECHO \"$ECHO\" | SED \"$sed_quote_subst\"`
    ECHO \"\

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
  eval 'cat <<_LTECHO_EOF'

```

```

\${1}
_LTECHO_EOF'
}
    ECHO="\${q}ECHO\"
fi

# Very basic option parsing. These options are (a) specific to
# the libtool wrapper, (b) are identical between the wrapper
# /script/ and the wrapper /executable/ which is used only on
# windows platforms, and (c) all begin with the string "--lt-"
# (application programs are unlikely to have options which match
# this pattern).
#
# There are only two supported options: --lt-debug and
# --lt-dump-script. There is, deliberately, no --lt-help.
#
# The first argument to this parsing function should be the
# script's $0 value, followed by "$@".
lt_option_debug=
func_parse_lt_options ()
{
    lt_script_arg0=\$0
    shift
    for lt_opt
    do
        case "\${lt_opt}" in
            --lt-debug) lt_option_debug=1 ;;
            --lt-dump-script)
                lt_dump_D="\${ECHO} \"X\${lt_script_arg0}\" | $SED -e 's/^X//' -e
's%/[^/]*$%'`
                test \"X\${lt_dump_D}\" = \"X\${lt_script_arg0}\" && lt_dump_D=.
                lt_dump_F="\${ECHO} \"X\${lt_script_arg0}\" | $SED -e 's/^X//' -e
's%^.*/%%'`
                cat \"\${lt_dump_D}/\${lt_dump_F}\"
                exit 0
                ;;
            --lt-*)
                \${ECHO} \"Unrecognized --lt- option: '\${lt_opt}'\" 1>&2
                exit 1
                ;;
        esac
    done

    # Print the debug banner immediately:
    if test -n "\${lt_option_debug}"; then
        echo "\${outputname}:\${output}:\${LINENO}: libtool wrapper (GNU
$PACKAGE$TIMESTAMP) $VERSION\" 1>&2
    fi
}

# Used when --lt-debug. Prints its arguments to stdout
# (redirection is the responsibility of the caller)

```



```

func_lt_dump_args ()
{
    lt_dump_args_N=1;
    for lt_arg
    do
        \${ECHO} "\${outputname}:${output}:\${LINENO}:
newargv[\$lt_dump_args_N]: \$lt_arg\"
        lt_dump_args_N=\`expr \$lt_dump_args_N + 1\`
    done
}

# Core function for launching the target application
func_exec_program_core ()
{
"
    case $host in
        # Backslashes separate directories on plain windows
        *-*-mingw | *-*-os2* | *-cegcc*)
            \${ECHO} "\
                if test -n \"\$lt_option_debug\"; then
                    \${ECHO} "\${outputname}:${output}:\${LINENO}: newargv[0]:
\${progdir}\\\\\\\\\${program}\" 1>&2
                    func_lt_dump_args \${1+\"\\$@\"} 1>&2
                fi
                exec \"\$progdir\\\\\\\\\${program}\" \${1+\"\\$@\"}
"
            ;;
        *)
            \${ECHO} "\
                if test -n \"\$lt_option_debug\"; then
                    \${ECHO} "\${outputname}:${output}:\${LINENO}: newargv[0]:
\${progdir}/\${program}\" 1>&2
                    func_lt_dump_args \${1+\"\\$@\"} 1>&2
                fi
                exec \"\$progdir/\${program}\" \${1+\"\\$@\"}
"
            ;;
    esac
    \${ECHO} "\
        \${ECHO} \"\$0: cannot exec \${program} \${*}\" 1>&2
        exit 1
"
}

# A function to encapsulate launching the target application
# Strips options in the --lt-* namespace from \@ and
# launches target application with the remaining arguments.
func_exec_program ()
{
    case \" \$* \" in
        *\\ --lt-*)
            for lt_wr_arg

```

```

do
    case \${lt_wr_arg} in
        --lt-*) ;;
        *) set x "\"\$@" "\"\${lt_wr_arg}\"; shift;;
    esac
    shift
done ;;
esac
func_exec_program_core \${1+\\"$@"}
}

# Parse options
func_parse_lt_options "\"$@" \${1+\\"$@"}

# Find the directory that this script lives in.
thisdir=\`$ECHO "\"$file\" | $SED 's%/[^/]*$%%'\`
test \"x$thisdir\" = \"x$file\" && thisdir=.

# Follow symbolic links until we get to the real thisdir.
file=\`ls -ld "\"$file\" | $SED -n 's/.*-> //p'\`
while test -n "\"$file\"; do
    destdir=\`$ECHO "\"$file\" | $SED 's%/[^/]*$%%'\`

    # If there was a directory component, then change thisdir.
    if test \"x$destdir\" != \"x$file\"; then
        case "\"$destdir\" in
            [\\\\/]* | [A-Za-z]:[\\\\/]* ) thisdir=\"\"$destdir\" ;;
            *) thisdir=\"\"$thisdir/$destdir\" ;;
        esac
    fi

    file=\`$ECHO "\"$file\" | $SED 's%^.*/%%%'\`
    file=\`ls -ld "\"$thisdir/$file\" | $SED -n 's/.*-> //p'\`
done

# Usually 'no', except on cygwin/mingw when embedded into
# the wrapper.
WRAPPER_SCRIPT_BELONGS_IN_OBJDIR=$func_emit_wrapper_arg1
if test "\"$WRAPPER_SCRIPT_BELONGS_IN_OBJDIR\" = \"yes\"; then
    # special case for '.'
    if test "\"$thisdir\" = \".\"; then
        thisdir=\`pwd\`
    fi
    # remove .libs from thisdir
    case "\"$thisdir\" in
        *[[\\\\/]]$objdir ) thisdir=\`$ECHO "\"$thisdir\" | $SED
's%[[\\\\/]][^\\\\/]*$%%'\` ;;
        $objdir ) thisdir=. ;;
    esac
fi

# Try to get the absolute directory name.

```

```

absdir=`cd `"\$thisdir\" && pwd`\`
test -n `"\$absdir\" && thisdir=`"\$absdir\"
"

    if test "$fast_install" = yes; then
        $ECHO "\
program=lt-'$outputname'$exeext
progdir=`"\$thisdir/$objdir\"

if test ! -f `"\$progdir/\$program\" ||
{ file=`ls -ldt `"\$progdir/\$program\"
`"\$progdir/./\$program\" 2>/dev/null | ${SED} lq\`; \
test `X$file\" != `X\$progdir/\$program\"; }; then

file=`"\$-\$program\"

if test ! -d `"\$progdir\"; then
    $MKDIR `"\$progdir\"
else
    $RM `"\$progdir/\$file\"
fi"

    $ECHO "\

# relink executable if necessary
if test -n `"\$relink_command\"; then
    if relink_command_output=`eval `\$relink_command 2>&1\`; then :
    else
    $ECHO `"\$relink_command_output\" >&2
    $RM `"\$progdir/\$file\"
    exit 1
    fi
fi

$MV `"\$progdir/\$file\" `"\$progdir/\$program\" 2>/dev/null ||
{ $RM `"\$progdir/\$program\";
  $MV `"\$progdir/\$file\" `"\$progdir/\$program\"; }
$RM `"\$progdir/\$file\"
fi"
    else
        $ECHO "\
program='$outputname'
progdir=`"\$thisdir/$objdir\"
"

    fi

    $ECHO "\

if test -f `"\$progdir/\$program\"; then"

    # fixup the dll searchpath if we need to.
    #

```

```

        # Fix the DLL searchpath if we need to.  Do this before
prepending
        # to shlibpath, because on Windows, both are PATH and uninstalled
        # libraries must come first.
        if test -n "$dllsearchpath"; then
            $ECHO "\
        # Add the dll search path components to the executable PATH
        PATH=$dllsearchpath:\$PATH
"
            fi

            # Export our shlibpath_var if we have one.
            if test "$shlibpath_overrides_runpath" = yes && test -n
"$shlibpath_var" && test -n "$temp_rpath"; then
                $ECHO "\
            # Add our own library path to $shlibpath_var
            $shlibpath_var="\$temp_rpath\$$shlibpath_var\"

            # Some systems cannot cope with colon-terminated $shlibpath_var
            # The second colon is a workaround for a bug in BeOS R4 sed
            $shlibpath_var=\`$ECHO "\$shlibpath_var" | $SED 's/::*\$//'\`

            export $shlibpath_var
"
                fi

                $ECHO "\
            if test \"\$libtool_execute_magic\" != \"\$magic\"; then
                # Run the actual program with our arguments.
                func_exec_program \${1+\\"$@"}
            fi
        else
            # The program doesn't exist.
            \$ECHO \"\$0: error: \\\"$progdir/\$program' does not exist\"
1>&2
            \$ECHO \"This script is just a wrapper for \$program.\" 1>&2
            \$ECHO \"See the $PACKAGE documentation for more information.\"
1>&2
            exit 1
        fi
    fi\
"
}

# func_emit_cwrappers_src
# emit the source code for a wrapper executable on stdout
# Must ONLY be called from within func_mode_link because
# it depends on a number of variable set therein.
func_emit_cwrappers_src ()
{
    cat <<EOF

```

```

/* $cwrappersource - temporary wrapper executable for
$objdir/$outputname
Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION

The $output program cannot be directly executed until all the
libtool
libraries that it depends on are installed.

This wrapper executable should never be moved out of the build
directory.
If it is, it will not operate correctly.
*/
EOF
        cat <<"EOF"
#ifdef _MSC_VER
# define _CRT_SECURE_NO_DEPRECATED 1
#endif
#include <stdio.h>
#include <stdlib.h>
#ifdef _MSC_VER
# include <direct.h>
# include <process.h>
# include <io.h>
#else
# include <unistd.h>
# include <stdint.h>
# ifdef __CYGWIN__
# include <io.h>
# endif
#endif
#include <malloc.h>
#include <stdarg.h>
#include <assert.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/stat.h>

/* declarations of non-ANSI functions */
#ifdef __MINGW32__
# ifdef __STRICT_ANSI__
int _putenv (const char *);
# endif
#elif defined(__CYGWIN__)
# ifdef __STRICT_ANSI__
char *realpath (const char *, char *);
int putenv (char *);
int setenv (const char *, const char *, int);
# endif
/* #elif defined (other platforms) ... */

```

```

#endif

/* portability defines, excluding path handling macros */
#if defined(_MSC_VER)
# define setmode _setmode
# define stat _stat
# define chmod _chmod
# define getcwd _getcwd
# define putenv _putenv
# define S_IXUSR _S_IXUSR
# define S_IXGRP _S_IXGRP
# define S_IXOTH _S_IXOTH
# ifndef _INTPTR_T_DEFINED
# define _INTPTR_T_DEFINED
# define intptr_t int
# endif
#elif defined(__MINGW32__)
# define setmode _setmode
# define stat _stat
# define chmod _chmod
# define getcwd _getcwd
# define putenv _putenv
#elif defined(__CYGWIN__)
# define HAVE_SETENV
# define FOPEN_WB "wb"
/* #elif defined (other platforms) ... */
#endif

#if defined(PATH_MAX)
# define LT_PATHMAX PATH_MAX
#elif defined(MAXPATHLEN)
# define LT_PATHMAX MAXPATHLEN
#else
# define LT_PATHMAX 1024
#endif

#ifndef S_IXOTH
# define S_IXOTH 0
#endif
#ifndef S_IXGRP
# define S_IXGRP 0
#endif

/* path handling portability macros */
#ifndef DIR_SEPARATOR
# define DIR_SEPARATOR '/'
# define PATH_SEPARATOR ':'
#endif

#if defined (__WIN32) || defined (__MSDOS__) || defined (__DJGPP__) ||
\
defined (__OS2__)
# define HAVE_DOS_BASED_FILE_SYSTEM
# define FOPEN_WB "wb"

```

```

# ifndef DIR_SEPARATOR_2
#  define DIR_SEPARATOR_2 '\\\
# endif
# ifndef PATH_SEPARATOR_2
#  define PATH_SEPARATOR_2 ';'
# endif
#endif

#ifndef DIR_SEPARATOR_2
# define IS_DIR_SEPARATOR(ch) ((ch) == DIR_SEPARATOR)
#else /* DIR_SEPARATOR_2 */
# define IS_DIR_SEPARATOR(ch) \
    (((ch) == DIR_SEPARATOR) || ((ch) == DIR_SEPARATOR_2))
#endif /* DIR_SEPARATOR_2 */

#ifndef PATH_SEPARATOR_2
# define IS_PATH_SEPARATOR(ch) ((ch) == PATH_SEPARATOR)
#else /* PATH_SEPARATOR_2 */
# define IS_PATH_SEPARATOR(ch) ((ch) == PATH_SEPARATOR_2)
#endif /* PATH_SEPARATOR_2 */

#ifndef FOPEN_WB
# define FOPEN_WB "w"
#endif
#ifndef _O_BINARY
# define _O_BINARY 0
#endif

#define XMALLOC(type, num)      ((type *) xmalloc ((num) *
sizeof(type)))
#define XFREE(stale) do { \
    if (stale) { free ((void *) stale); stale = 0; } \
} while (0)

#if defined(LT_DEBUGWRAPPER)
static int lt_debug = 1;
#else
static int lt_debug = 0;
#endif

const char *program_name = "libtool-wrapper"; /* in case xstrdup fails
*/

void *xmalloc (size_t num);
char *xstrdup (const char *string);
const char *base_name (const char *name);
char *find_executable (const char *wrapper);
char *chase_symlinks (const char *pathspec);
int make_executable (const char *path);
int check_executable (const char *path);
char *strendzap (char *str, const char *pat);

```

```

void lt_debugprintf (const char *file, int line, const char *fmt,
...);
void lt_fatal (const char *file, int line, const char *message, ...);
static const char *nonnull (const char *s);
static const char *nonempty (const char *s);
void lt_setenv (const char *name, const char *value);
char *lt_extend_str (const char *orig_value, const char *add, int
to_end);
void lt_update_exe_path (const char *name, const char *value);
void lt_update_lib_path (const char *name, const char *value);
char **prepare_spawn (char **argv);
void lt_dump_script (FILE *f);
EOF

```

```

        cat <<EOF
volatile const char * MAGIC_EXE = "$magic_exe";
const char * LIB_PATH_VARNAME = "$shlibpath_var";
EOF

        if test "$shlibpath_overrides_runpath" = yes && test -n
"$shlibpath_var" && test -n "$temp_rpath"; then
            func_to_host_path "$temp_rpath"
            cat <<EOF
const char * LIB_PATH_VALUE = "$func_to_host_path_result";
EOF
        else
            cat <<"EOF"
const char * LIB_PATH_VALUE = "";
EOF
        fi

        if test -n "$dllsearchpath"; then
            func_to_host_path "$dllsearchpath:"
            cat <<EOF
const char * EXE_PATH_VARNAME = "PATH";
const char * EXE_PATH_VALUE = "$func_to_host_path_result";
EOF
        else
            cat <<"EOF"
const char * EXE_PATH_VARNAME = "";
const char * EXE_PATH_VALUE = "";
EOF
        fi

        if test "$fast_install" = yes; then
            cat <<EOF
const char * TARGET_PROGRAM_NAME = "lt-$outputname"; /* hopefully, no
.exe */
EOF
        else
            cat <<EOF

```



```

const char * TARGET_PROGRAM_NAME = "$outputname"; /* hopefully, no
.exe */
EOF
    fi

    cat <<"EOF"

#define LTWRAPPER_OPTION_PREFIX          "--lt-"

static const char *ltwrapper_option_prefix = LTWRAPPER_OPTION_PREFIX;
static const char *dumpscrip_opt         = LTWRAPPER_OPTION_PREFIX
"dump-script";
static const char *debug_opt             = LTWRAPPER_OPTION_PREFIX
"debug";

int
main (int argc, char *argv[])
{
    char **newargz;
    int newargc;
    char *tmp_pathspec;
    char *actual_cwrapper_path;
    char *actual_cwrapper_name;
    char *target_name;
    char *lt_argv_zero;
    intptr_t rval = 127;

    int i;

    program_name = (char *) xstrdup (base_name (argv[0]));
    newargz = XMALLOC (char *, argc + 1);

    /* very simple arg parsing; don't want to rely on getopt
     * also, copy all non cwrapper options to newargz, except
     * argz[0], which is handled differently
     */
    newargc=0;
    for (i = 1; i < argc; i++)
    {
        if (strcmp (argv[i], dumpscrip_opt) == 0)
        {
EOF
            case "$host" in
                *mingw* | *cygwin* )
                    # make stdout use "unix" line endings
                    echo "          setmode(1, _O_BINARY);"
                    ;;
                esac

            cat <<"EOF"
            lt_dump_script (stdout);

```

```

    return 0;
}
if (strcmp (argv[i], debug_opt) == 0)
{
    lt_debug = 1;
    continue;
}
if (strcmp (argv[i], ltwrapper_option_prefix) == 0)
{
    /* however, if there is an option in the
LTWRAPPER_OPTION_PREFIX
    namespace, but it is not one of the ones we know about
and
    have already dealt with, above (including dump-script),
then
    report an error. Otherwise, targets might begin to
believe
    they are allowed to use options in the
LTWRAPPER_OPTION_PREFIX
    namespace. The first time any user complains about this,
we'll
    need to make LTWRAPPER_OPTION_PREFIX a configure-time
option
    or a configure.ac-settable value.
    */
    lt_fatal (__FILE__, __LINE__,
              "unrecognized %s option: '%s'",
              ltwrapper_option_prefix, argv[i]);
}
/* otherwise ... */
newargz[++newargc] = xstrdup (argv[i]);
}
newargz[++newargc] = NULL;

EOF
    cat <<EOF
    /* The GNU banner must be the first non-error debug message */
    lt_debugprintf (__FILE__, __LINE__, "libtool wrapper (GNU
PACKAGE$TIMESTAMP) $VERSION\n");
EOF
    cat <<"EOF"
    lt_debugprintf (__FILE__, __LINE__, "(main) argv[0]: %s\n",
argv[0]);
    lt_debugprintf (__FILE__, __LINE__, "(main) program_name: %s\n",
program_name);

    tmp_pathspec = find_executable (argv[0]);
    if (tmp_pathspec == NULL)
        lt_fatal (__FILE__, __LINE__, "couldn't find %s", argv[0]);
    lt_debugprintf (__FILE__, __LINE__,
                    "(main) found exe (before symlink chase) at: %s\n",
                    tmp_pathspec);

```

```

actual_cwrapper_path = chase_symlinks (tmp_pathspec);
lt_debugprintf (__FILE__, __LINE__,
                "(main) found exe (after symlink chase) at: %s\n",
                actual_cwrapper_path);
XFREE (tmp_pathspec);

actual_cwrapper_name = xstrdup (base_name (actual_cwrapper_path));
strendzap (actual_cwrapper_path, actual_cwrapper_name);

/* wrapper name transforms */
strendzap (actual_cwrapper_name, ".exe");
tmp_pathspec = lt_extend_str (actual_cwrapper_name, ".exe", 1);
XFREE (actual_cwrapper_name);
actual_cwrapper_name = tmp_pathspec;
tmp_pathspec = 0;

/* target_name transforms -- use actual target program name; might
have lt- prefix */
target_name = xstrdup (base_name (TARGET_PROGRAM_NAME));
strendzap (target_name, ".exe");
tmp_pathspec = lt_extend_str (target_name, ".exe", 1);
XFREE (target_name);
target_name = tmp_pathspec;
tmp_pathspec = 0;

lt_debugprintf (__FILE__, __LINE__,
                "(main) libtool target name: %s\n",
                target_name);
EOF

    cat <<EOF
newargz[0] =
    XMALLOC (char, (strlen (actual_cwrapper_path) +
                    strlen ("$_objdir") + 1 + strlen (actual_cwrapper_name)
+ 1));
strcpy (newargz[0], actual_cwrapper_path);
strcat (newargz[0], "$objdir");
strcat (newargz[0], "/");
EOF

    cat <<"EOF"
/* stop here, and copy so we don't have to do this twice */
tmp_pathspec = xstrdup (newargz[0]);

/* do NOT want the lt- prefix here, so use actual_cwrapper_name */
strcat (newargz[0], actual_cwrapper_name);

/* DO want the lt- prefix here if it exists, so use target_name */
lt_argv_zero = lt_extend_str (tmp_pathspec, target_name, 1);
XFREE (tmp_pathspec);
tmp_pathspec = NULL;

```

EOF

```
        case $host_os in
            mingw*)
                cat <<"EOF"
{
    char* p;
    while ((p = strchr (newargz[0], '\\')) != NULL)
        {
            *p = '/';
        }
    while ((p = strchr (lt_argv_zero, '\\')) != NULL)
        {
            *p = '/';
        }
}
EOF

        ;;
    esac

        cat <<"EOF"
XFREE (target_name);
XFREE (actual_cwrapper_path);
XFREE (actual_cwrapper_name);

    lt_setenv ("BIN_SH", "xpg4"); /* for Tru64 */
    lt_setenv ("DUALCASE", "1"); /* for MSK sh */
    /* Update the DLL searchpath.  EXE_PATH_VALUE ($dllsearchpath) must
       be prepended before (that is, appear after) LIB_PATH_VALUE
       ($temp_rpath)
       because on Windows, both *_VARNAMES are PATH but uninstalled
       libraries must come first. */
    lt_update_exe_path (EXE_PATH_VARNAME, EXE_PATH_VALUE);
    lt_update_lib_path (LIB_PATH_VARNAME, LIB_PATH_VALUE);

    lt_debugprintf (__FILE__, __LINE__, "(main) lt_argv_zero: %s\n",
                    nonnull (lt_argv_zero));
    for (i = 0; i < newargc; i++)
        {
            lt_debugprintf (__FILE__, __LINE__, "(main) newargz[%d]: %s\n",
                            i, nonnull (newargz[i]));
        }
}
EOF
```

EOF

```
        case $host_os in
            mingw*)
                cat <<"EOF"
/* execv doesn't actually work on mingw as expected on unix */
newargz = prepare_spawn (newargz);
rval = _spawnv (_P_WAIT, lt_argv_zero, (const char * const *)
newargz);
```

```

if (rval == -1)
{
    /* failed to start process */
    lt_debugprintf (__FILE__, __LINE__,
        "(main) failed to launch target \"%s\": %s\n",
        lt_argv_zero, nonnull (strerror (errno)));
    return 127;
}
return rval;
EOF
    ;;
    *)
        cat <<"EOF"
        execv (lt_argv_zero, newargz);
        return rval; /* =127, but avoids unused variable warning */
EOF
    ;;
    esac

    cat <<"EOF"
}

void *
xmalloc (size_t num)
{
    void *p = (void *) malloc (num);
    if (!p)
        lt_fatal (__FILE__, __LINE__, "memory exhausted");

    return p;
}

char *
xstrdup (const char *string)
{
    return string ? strcpy ((char *) xmalloc (strlen (string) + 1),
        string) : NULL;
}

const char *
base_name (const char *name)
{
    const char *base;

#ifdef HAVE_DOS_BASED_FILE_SYSTEM
    /* Skip over the disk name in MSDOS pathnames. */
    if (isalpha ((unsigned char) name[0]) && name[1] == ':')
        name += 2;
#endif

    for (base = name; *name; name++)
        if (IS_DIR_SEPARATOR (*name))

```

```

        base = name + 1;
    return base;
}

int
check_executable (const char *path)
{
    struct stat st;

    lt_debugprintf (__FILE__, __LINE__, "(check_executable): %s\n",
                    nonempty (path));
    if ((!path) || (!*path))
        return 0;

    if ((stat (path, &st) >= 0)
        && (st.st_mode & (S_IXUSR | S_IXGRP | S_IXOTH)))
        return 1;
    else
        return 0;
}

int
make_executable (const char *path)
{
    int rval = 0;
    struct stat st;

    lt_debugprintf (__FILE__, __LINE__, "(make_executable): %s\n",
                    nonempty (path));
    if ((!path) || (!*path))
        return 0;

    if (stat (path, &st) >= 0)
    {
        rval = chmod (path, st.st_mode | S_IXOTH | S_IXGRP | S_IXUSR);
    }
    return rval;
}

/* Searches for the full path of the wrapper. Returns
   newly allocated full path name if found, NULL otherwise
   Does not chase symlinks, even on platforms that support them.
*/
char *
find_executable (const char *wrapper)
{
    int has_slash = 0;
    const char *p;
    const char *p_next;
    /* static buffer for getcwd */
    char tmp[LT_PATHMAX + 1];
    int tmp_len;

```

```

char *concat_name;

lt_debugprintf (__FILE__, __LINE__, "(find_executable): %s\n",
                nonempty (wrapper));

if ((wrapper == NULL) || (*wrapper == '\\0'))
    return NULL;

/* Absolute path? */
#if defined (HAVE_DOS_BASED_FILE_SYSTEM)
    if (isalpha ((unsigned char) wrapper[0]) && wrapper[1] == ':')
    {
        concat_name = xstrdup (wrapper);
        if (check_executable (concat_name))
            return concat_name;
        XFREE (concat_name);
    }
else
    {
#endif
        if (IS_DIR_SEPARATOR (wrapper[0]))
        {
            concat_name = xstrdup (wrapper);
            if (check_executable (concat_name))
                return concat_name;
            XFREE (concat_name);
        }
#if defined (HAVE_DOS_BASED_FILE_SYSTEM)
    }
#endif

for (p = wrapper; *p; p++)
    if (*p == '/')
    {
        has_slash = 1;
        break;
    }
if (!has_slash)
    {
        /* no slashes; search PATH */
        const char *path = getenv ("PATH");
        if (path != NULL)
        {
            for (p = path; *p; p = p_next)
            {
                const char *q;
                size_t p_len;
                for (q = p; *q; q++)
                    if (IS_PATH_SEPARATOR (*q))
                        break;
                p_len = q - p;
                p_next = (*q == '\\0' ? q : q + 1);
            }
        }
    }

```

```

    if (p_len == 0)
    {
        /* empty path: current directory */
        if (getcwd (tmp, LT_PATHMAX) == NULL)
            lt_fatal (__FILE__, __LINE__, "getcwd failed: %s",
                    nonnull (strerror (errno)));
        tmp_len = strlen (tmp);
        concat_name =
            XMALLOC (char, tmp_len + 1 + strlen (wrapper) + 1);
        memcpy (concat_name, tmp, tmp_len);
        concat_name[tmp_len] = '/';
        strcpy (concat_name + tmp_len + 1, wrapper);
    }
    else
    {
        concat_name =
            XMALLOC (char, p_len + 1 + strlen (wrapper) + 1);
        memcpy (concat_name, p, p_len);
        concat_name[p_len] = '/';
        strcpy (concat_name + p_len + 1, wrapper);
    }
    if (check_executable (concat_name))
        return concat_name;
    XFREE (concat_name);
}
}
/* not found in PATH; assume curdir */
}
/* Relative path | not found in path: prepend cwd */
if (getcwd (tmp, LT_PATHMAX) == NULL)
    lt_fatal (__FILE__, __LINE__, "getcwd failed: %s",
            nonnull (strerror (errno)));
tmp_len = strlen (tmp);
concat_name = XMALLOC (char, tmp_len + 1 + strlen (wrapper) + 1);
memcpy (concat_name, tmp, tmp_len);
concat_name[tmp_len] = '/';
strcpy (concat_name + tmp_len + 1, wrapper);

if (check_executable (concat_name))
    return concat_name;
XFREE (concat_name);
return NULL;
}

char *
chase_symlinks (const char *pathspec)
{
#ifdef S_ISLNK
    return xstrdup (pathspec);
#else
    char buf[LT_PATHMAX];
    struct stat s;

```



```

char *tmp_pathspec = xstrdup (pathspec);
char *p;
int has_symlinks = 0;
while (strlen (tmp_pathspec) && !has_symlinks)
{
    lt_debugprintf (__FILE__, __LINE__,
                    "checking path component for symlinks: %s\n",
                    tmp_pathspec);
    if (lstat (tmp_pathspec, &s) == 0)
    {
        if (S_ISLNK (s.st_mode) != 0)
        {
            has_symlinks = 1;
            break;
        }

        /* search backwards for last DIR_SEPARATOR */
        p = tmp_pathspec + strlen (tmp_pathspec) - 1;
        while ((p > tmp_pathspec) && (!IS_DIR_SEPARATOR (*p)))
            p--;
        if ((p == tmp_pathspec) && (!IS_DIR_SEPARATOR (*p)))
        {
            /* no more DIR_SEPARATORS left */
            break;
        }
        *p = '\0';
    }
    else
    {
        lt_fatal (__FILE__, __LINE__,
                  "error accessing file \"%s\": %s",
                  tmp_pathspec, nonnull (strerror (errno)));
    }
}
XFREE (tmp_pathspec);

if (!has_symlinks)
{
    return xstrdup (pathspec);
}

tmp_pathspec = realpath (pathspec, buf);
if (tmp_pathspec == 0)
{
    lt_fatal (__FILE__, __LINE__,
              "could not follow symlinks for %s", pathspec);
}
return xstrdup (tmp_pathspec);
#endif
}

char *

```

```

strendzap (char *str, const char *pat)
{
    size_t len, patlen;

    assert (str != NULL);
    assert (pat != NULL);

    len = strlen (str);
    patlen = strlen (pat);

    if (patlen <= len)
    {
        str += len - patlen;
        if (strcmp (str, pat) == 0)
            *str = '\0';
    }
    return str;
}

void
lt_debugprintf (const char *file, int line, const char *fmt, ...)
{
    va_list args;
    if (lt_debug)
    {
        (void) fprintf (stderr, "%s:%s:%d: ", program_name, file, line);
        va_start (args, fmt);
        (void) vfprintf (stderr, fmt, args);
        va_end (args);
    }
}

static void
lt_error_core (int exit_status, const char *file,
               int line, const char *mode,
               const char *message, va_list ap)
{
    fprintf (stderr, "%s:%s:%d: %s: ", program_name, file, line, mode);
    vfprintf (stderr, message, ap);
    fprintf (stderr, ".\n");

    if (exit_status >= 0)
        exit (exit_status);
}

void
lt_fatal (const char *file, int line, const char *message, ...)
{
    va_list ap;
    va_start (ap, message);
    lt_error_core (EXIT_FAILURE, file, line, "FATAL", message, ap);
    va_end (ap);
}

```

```

}

static const char *
nonnull (const char *s)
{
    return s ? s : "(null)";
}

static const char *
nonempty (const char *s)
{
    return (s && !*s) ? "(empty)" : nonnull (s);
}

void
lt_setenv (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
                    "(lt_setenv) setting '%s' to '%s'\n",
                    nonnull (name), nonnull (value));
    {
#ifdef HAVE_SETENV
        /* always make a copy, for consistency with !HAVE_SETENV */
        char *str = xstrdup (value);
        setenv (name, str, 1);
#else
        int len = strlen (name) + 1 + strlen (value) + 1;
        char *str = XMALLOC (char, len);
        sprintf (str, "%s=%s", name, value);
        if (putenv (str) != EXIT_SUCCESS)
            {
                XFREE (str);
            }
#endif
    }
}

char *
lt_extend_str (const char *orig_value, const char *add, int to_end)
{
    char *new_value;
    if (orig_value && *orig_value)
        {
            int orig_value_len = strlen (orig_value);
            int add_len = strlen (add);
            new_value = XMALLOC (char, add_len + orig_value_len + 1);
            if (to_end)
                {
                    strcpy (new_value, orig_value);
                    strcpy (new_value + orig_value_len, add);
                }
            else

```

```

        {
            strcpy (new_value, add);
            strcpy (new_value + add_len, orig_value);
        }
    }
else
    {
        new_value = xstrdup (add);
    }
return new_value;
}

void
lt_update_exe_path (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
        "(lt_update_exe_path) modifying '%s' by prepending
'%s'\n",
        nonnull (name), nonnull (value));

    if (name && *name && value && *value)
        {
            char *new_value = lt_extend_str (getenv (name), value, 0);
            /* some systems can't cope with a ':'-terminated path '#' */
            int len = strlen (new_value);
            while (((len = strlen (new_value)) > 0) && IS_PATH_SEPARATOR
(new_value[len-1]))
                {
                    new_value[len-1] = '\0';
                }
            lt_setenv (name, new_value);
            XFREE (new_value);
        }
}

void
lt_update_lib_path (const char *name, const char *value)
{
    lt_debugprintf (__FILE__, __LINE__,
        "(lt_update_lib_path) modifying '%s' by prepending
'%s'\n",
        nonnull (name), nonnull (value));

    if (name && *name && value && *value)
        {
            char *new_value = lt_extend_str (getenv (name), value, 0);
            lt_setenv (name, new_value);
            XFREE (new_value);
        }
}

EOF
```

```

        case $host_os in
            mingw*)
                cat <<"EOF"

/* Prepares an argument vector before calling spawn().
   Note that spawn() does not by itself call the command interpreter
   (getenv ("COMSPEC") != NULL ? getenv ("COMSPEC") :
    ({ OSVERSIONINFO v; v.dwOSVersionInfoSize =
sizeof(OSVERSIONINFO);
    GetVersionEx(&v);
    v.dwPlatformId == VER_PLATFORM_WIN32_NT;
    }) ? "cmd.exe" : "command.com").
   Instead it simply concatenates the arguments, separated by ' ', and
   calls
   CreateProcess(). We must quote the arguments since Win32
   CreateProcess()
   interprets characters like ' ', '\t', '\\', '"' (but not '<' and
   '>') in a
   special way:
   - Space and tab are interpreted as delimiters. They are not treated
   as
   delimiters if they are surrounded by double quotes: "...".
   - Unescaped double quotes are removed from the input. Their only
   effect is
   that within double quotes, space and tab are treated like normal
   characters.
   - Backslashes not followed by double quotes are not special.
   - But 2*n+1 backslashes followed by a double quote become
   n backslashes followed by a double quote (n >= 0):
    \" -> "
    \\\" -> \"
    \\\\" -> \\\"
*/
#define SHELL_SPECIAL_CHARS "\\\"\\
\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017\020\021\0
22\023\024\025\026\027\030\031\032\033\034\035\036\037"
#define SHELL_SPACE_CHARS "
\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017\020\021\0
22\023\024\025\026\027\030\031\032\033\034\035\036\037"
char **
prepare_spawn (char **argv)
{
    size_t argc;
    char **new_argv;
    size_t i;

    /* Count number of arguments. */
    for (argc = 0; argv[argc] != NULL; argc++)
        ;

    /* Allocate new argument vector. */
    new_argv = XMALLOC (char *, argc + 1);

```

```

/* Put quoted arguments into the new argument vector. */
for (i = 0; i < argc; i++)
{
    const char *string = argv[i];

    if (string[0] == '\\0')
new_argv[i] = xstrdup ("\\0");
    else if (strpbrk (string, SHELL_SPECIAL_CHARS) != NULL)
    {
        int quote_around = (strpbrk (string, SHELL_SPACE_CHARS) !=
NULL);
        size_t length;
        unsigned int backslashes;
        const char *s;
        char *quoted_string;
        char *p;

        length = 0;
        backslashes = 0;
        if (quote_around)
            length++;
        for (s = string; *s != '\\0'; s++)
        {
            char c = *s;
            if (c == '"')
                length += backslashes + 1;
            length++;
            if (c == '\\')
                backslashes++;
            else
                backslashes = 0;
        }
        if (quote_around)
            length += backslashes + 1;

        quoted_string = XMALLOC (char, length + 1);

        p = quoted_string;
        backslashes = 0;
        if (quote_around)
            *p++ = '"';
        for (s = string; *s != '\\0'; s++)
        {
            char c = *s;
            if (c == '"')
            {
                unsigned int j;
                for (j = backslashes + 1; j > 0; j--)
                    *p++ = '\\';
            }
            *p++ = c;
        }
    }
}

```

```

        if (c == '\\')
            backslashes++;
        else
            backslashes = 0;
    }
    if (quote_around)
    {
        unsigned int j;
        for (j = backslashes; j > 0; j--)
            *p++ = '\\';
        *p++ = '"';
    }
    *p = '\\0';

    new_argv[i] = quoted_string;
}
else
    new_argv[i] = (char *) string;
}
new_argv[argc] = NULL;

return new_argv;
}
EOF

        ;;
    esac

        cat <<"EOF"
void lt_dump_script (FILE* f)
{
EOF
        func_emit_wrapper yes |
            $SED -n -e '
s/^\(.\{79\}\)\(.*\)/\1\
\2/
h
s/\([\\" ]\)/\\\1/g
s/\$/\\n/
s/\([\^\\n]*\).*/ fputs ("\1", f);/p
g
D'
        cat <<"EOF"
}
EOF
}
# end: func_emit_cwrapperexe_src

# func_win32_import_lib_p ARG
# True if ARG is an import lib, as indicated by $file_magic_cmd
func_win32_import_lib_p ()
{
    $opt_debug

```

```

    case `eval $file_magic_cmd "\${1}" 2>/dev/null | $SED -e 10q` in
    *import*) : ;;
    *) false ;;
    esac
}

# func_mode_link arg...
func_mode_link ()
{
    $opt_debug
    case $host in
    *-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-cegcc*)
        # It is impossible to link a dll without this setting, and
        # we shouldn't force the makefile maintainer to figure out
        # which system we are compiling for in order to pass an extra
        # flag for every libtool invocation.
        # allow_undefined=no

        # FIXME: Unfortunately, there are problems with the above when
trying
        # to make a dll which has undefined symbols, in which case not
        # even a static library is built. For now, we need to specify
        # -no-undefined on the libtool link line when we can be certain
        # that all symbols are satisfied, otherwise we get a static
library.
        allow_undefined=yes
        ;;
    *)
        allow_undefined=yes
        ;;
    esac
    libtool_args=$nonopt
    base_compile="$nonopt $@"
    compile_command=$nonopt
    finalize_command=$nonopt

    compile_rpath=
    finalize_rpath=
    compile_shlibpath=
    finalize_shlibpath=
    convenience=
    old_convenience=
    deplibs=
    old_deplibs=
    compiler_flags=
    linker_flags=
    dllsearchpath=
    lib_search_path=`pwd`
    inst_prefix_dir=
    new_inherited_linker_flags=

    avoid_version=no

```



```

bindir=
dlfiles=
dlprefiles=
dlself=no
export_dynamic=no
export_symbols=
export_symbols_regex=
generated=
libobjs=
ltlibs=
module=no
no_install=no
objs=
non_pic_objects=
precious_files_regex=
prefer_static_libs=no
preload=no
prev=
prevarg=
release=
rpath=
xrpath=
perm_rpath=
temp_rpath=
thread_safe=no
vinfo=
vinfo_number=no
weak_libs=
single_module="{wl}-single_module"
func_infer_tag $base_compile

# We need to know -static, to get the right output filenames.
for arg
do
    case $arg in
        -shared)
            test "$build_libtool_libs" != yes && \
                func_fatal_configuration "can not build a shared library"
            build_old_libs=no
            break
        ;;
        -all-static | -static | -static-libtool-libs)
            case $arg in
                -all-static)
                    if test "$build_libtool_libs" = yes && test -z
"$link_static_flag"; then
                        func_warning "complete static linking is impossible in this
configuration"
                    fi
                    if test -n "$link_static_flag"; then
                        dlopen_self=$dlopen_self_static
                    fi
                fi
            fi
        ;;
    esac
done

```

```

    prefer_static_libs=yes
    ;;
-static)
    if test -z "$pic_flag" && test -n "$link_static_flag"; then
        dlopen_self=$dlopen_self_static
    fi
    prefer_static_libs=built
    ;;
-static-libtool-libs)
    if test -z "$pic_flag" && test -n "$link_static_flag"; then
        dlopen_self=$dlopen_self_static
    fi
    prefer_static_libs=yes
    ;;
esac
build_libtool_libs=no
build_old_libs=yes
break
;;
esac
done

# See if our shared archives depend on static archives.
test -n "$old_archive_from_new_cmds" && build_old_libs=yes

# Go through the arguments, transforming them on the way.
while test "$#" -gt 0; do
    arg="$1"
    shift
    func_quote_for_eval "$arg"
    qarg=$func_quote_for_eval_unquoted_result
    libtool_args+=" $func_quote_for_eval_result"

    # If the previous option needs an argument, assign it.
    if test -n "$prev"; then
    case $prev in
output)
        compile_command+=" @OUTPUT@"
        finalize_command+=" @OUTPUT@"
        ;;
esac

case $prev in
bindir)
    bindir="$arg"
    prev=
    continue
    ;;
dlfiles|dlprefiles)
    if test "$preload" = no; then
        # Add the symbol object into the linking commands.
        compile_command+=" @SYMFIL@"

```

```

        finalize_command+=" @SYMFILE@"
        preload=yes
    fi
    case $arg in
        *.la | *.lo) ;; # We handle these cases below.
    force)
        if test "$dlsself" = no; then
            dlsself=needless
            export_dynamic=yes
        fi
        prev=
        continue
        ;;
    self)
        if test "$prev" = dlprefiles; then
            dlsself=yes
        elif test "$prev" = dlfiles && test "$dlopen_self" != yes;
then
            dlsself=yes
        else
            dlsself=needless
            export_dynamic=yes
        fi
        prev=
        continue
        ;;
    *)
        if test "$prev" = dlfiles; then
            dlfiles+=" $arg"
        else
            dlprefiles+=" $arg"
        fi
        prev=
        continue
        ;;
    esac
    expsyms)
        export_symbols="$arg"
        test -f "$arg" \
            || func_fatal_error "symbol file ``$arg' does not exist"
        prev=
        continue
        ;;
    expsyms_regex)
        export_symbols_regex="$arg"
        prev=
        continue
        ;;
    framework)
        case $host in
            *-*-darwin*)

```

```

        case "$deplibs " in
        *" $qarg.ltframework ") ;;
        *) deplibs+=" $qarg.ltframework" # this is fixed later
           ;;
        esac
        ;;
    esac
    prev=
    continue
    ;;
inst_prefix)
    inst_prefix_dir="$arg"
    prev=
    continue
    ;;
objectlist)
    if test -f "$arg"; then
        save_arg=$arg
        moreargs=
        for fil in `cat "$save_arg"`
        do
#           moreargs+=" $fil"
           arg=$fil
           # A libtool-controlled object.

           # Check to see that this really is a libtool object.
           if func_lalib_unsafe_p "$arg"; then
               pic_object=
               non_pic_object=

               # Read the .lo file
               func_source "$arg"

               if test -z "$pic_object" ||
                  test -z "$non_pic_object" ||
                  test "$pic_object" = none &&
                  test "$non_pic_object" = none; then
                   func_fatal_error "cannot find name of object for `\$arg'"
               fi

               # Extract subdirectory from the argument.
               func_dirname "$arg" "/" ""
               xdir="$func_dirname_result"

               if test "$pic_object" != none; then
                   # Prepend the subdirectory the object is found in.
                   pic_object="$xdir$pic_object"

                   if test "$prev" = dlfiles; then
                       if test "$build_libtool_libs" = yes && test
"$dlopen_support" = yes; then
                           dlfiles+=" $pic_object"

```

```

        prev=
        continue
    else
        # If libtool objects are unsupported, then we need to
preload.
        prev=dlprefiles
    fi
fi

# CHECK ME: I think I busted this. -Ossama
if test "$prev" = dlprefiles; then
    # Preload the old-style object.
    dlprefiles+=" $pic_object"
    prev=
fi

# A PIC object.
libobjs+=" $pic_object"
arg="$pic_object"
fi

# Non-PIC object.
if test "$non_pic_object" != none; then
    # Prepend the subdirectory the object is found in.
    non_pic_object="$xdir$non_pic_object"

    # A standard non-PIC object
    non_pic_objects+=" $non_pic_object"
    if test -z "$pic_object" || test "$pic_object" = none ;
then
        arg="$non_pic_object"
    fi
else
    # If the PIC object exists, use it instead.
    # $xdir was prepended to $pic_object above.
    non_pic_object="$pic_object"
    non_pic_objects+=" $non_pic_object"
fi
else
    # Only an error if not doing a dry-run.
    if $opt_dry_run; then
        # Extract subdirectory from the argument.
        func_dirname "$arg" "/" ""
        xdir="$func_dirname_result"

        func_lo2o "$arg"
        pic_object=$xdir$objdir/$func_lo2o_result
        non_pic_object=$xdir$func_lo2o_result
        libobjs+=" $pic_object"
        non_pic_objects+=" $non_pic_object"
    else
        func_fatal_error "`$arg' is not a valid libtool object"
    fi
fi

```

```

        fi
        fi
    done
else
    func_fatal_error "link input file ``$arg' does not exist"
fi
arg=$save_arg
prev=
continue
;;
precious_regex)
    precious_files_regex="$arg"
    prev=
    continue
    ;;
release)
    release="- $arg"
    prev=
    continue
    ;;
rpath | xrpath)
    # We need an absolute path.
    case $arg in
    [\\/*]* | [A-Za-z]:[\\/*]*) ;;
    *)
        func_fatal_error "only absolute run-paths are allowed"
        ;;
    esac
    if test "$prev" = rpath; then
        case "$rpath " in
        *" $arg "*) ;;
        *) rpath+=" $arg" ;;
        esac
    else
        case "$xrpath " in
        *" $arg "*) ;;
        *) xrpath+=" $arg" ;;
        esac
    fi
    prev=
    continue
    ;;
shrext)
    shrext_cmds="$arg"
    prev=
    continue
    ;;
weak)
    weak_libs+=" $arg"
    prev=
    continue
    ;;

```

```

xcclinker)
  linker_flags+=" $qarg"
  compiler_flags+=" $qarg"
  prev=
  compile_command+=" $qarg"
  finalize_command+=" $qarg"
  continue
  ;;
xcompiler)
  compiler_flags+=" $qarg"
  prev=
  compile_command+=" $qarg"
  finalize_command+=" $qarg"
  continue
  ;;
xlinker)
  linker_flags+=" $qarg"
  compiler_flags+=" $wl$qarg"
  prev=
  compile_command+=" $wl$qarg"
  finalize_command+=" $wl$qarg"
  continue
  ;;
*)
  eval "$prev=\"\$arg\""
  prev=
  continue
  ;;
esac
fi # test -n "$prev"

prevarg="$arg"

case $arg in
  -all-static)
if test -n "$link_static_flag"; then
  # See comment for -static flag below, for more details.
  compile_command+=" $link_static_flag"
  finalize_command+=" $link_static_flag"
fi
  continue
  ;;

  -allow-undefined)
  # FIXME: remove this flag sometime in the future.
  func_fatal_error "\`-allow-undefined' must not be used because it
is the default"
  ;;

  -avoid-version)
  avoid_version=yes
  continue

```

```

;;

-bindir)
prev=bindir
continue
;;

-dlopen)
prev=dlfiles
continue
;;

-dlpreopen)
prev=dlprefiles
continue
;;

-export-dynamic)
export_dynamic=yes
continue
;;

-export-symbols | -export-symbols-regex)
if test -n "$export_symbols" || test -n "$export_symbols_regex";
then
    func_fatal_error "more than one -exported-symbols argument is
not allowed"
fi
if test "X$arg" = "X-export-symbols"; then
    prev=expsyms
else
    prev=expsyms_regex
fi
continue
;;

-framework)
prev=framework
continue
;;

-inst-prefix-dir)
prev=inst_prefix
continue
;;

# The native IRIX linker understands -LANG:*, -LIST:* and -LNO:*
# so, if we see these flags be careful not to treat them like -L
-L[A-Z][A-Z]*:*)
case $with_gcc/$host in
no/*-*-irix* | /*-*-irix*)
    compile_command+=" $arg"

```



```

        finalize_command+=" $arg"
        ;;
    esac
    continue
    ;;

-L*)
func_stripname "-L" '' "$arg"
if test -z "$func_stripname_result"; then
    if test "$#" -gt 0; then
        func_fatal_error "require no space between \`-L' and \`${1}'"
    else
        func_fatal_error "need path for \`-L' option"
    fi
fi
func_resolve_sysroot "$func_stripname_result"
dir=$func_resolve_sysroot_result
# We need an absolute path.
case $dir in
[\\/] * | [A-Za-z]:[\\/] *) ;;
*)
    absdir=`cd "$dir" && pwd`
    test -z "$absdir" && \
        func_fatal_error "cannot determine absolute directory name of
\`${dir}'"
    dir="$absdir"
    ;;
esac
case "$deplibs " in
*" -L$dir " * | *" $arg " *)
    # Will only happen for absolute or sysroot arguments
    ;;
*)
    # Preserve sysroot, but never include relative directories
    case $dir in
        [\\/] * | [A-Za-z]:[\\/] * | =*) deplibs+=" $arg" ;;
        *) deplibs+=" -L$dir" ;;
    esac
    lib_search_path+=" $dir"
    ;;
esac
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-cegcc*)
    testbindir=`$ECHO "$dir" | $SED 's*/lib$*/bin*'`
    case :$dllsearchpath: in
        *:$dir:*) ;;
        : :) dllsearchpath=$dir;;
        *) dllsearchpath+=":$dir";;
    esac
    case :$dllsearchpath: in
        *:$testbindir:*) ;;
        : :) dllsearchpath=$testbindir;;
    esac

```

```

    *) dllsearchpath+=":$testbindir";;
    esac
    ;;
esac
continue
;;

-l*)
if test "X$arg" = "X-lc" || test "X$arg" = "X-lm"; then
    case $host in
        *-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-beos* | *-cegcc* |
*-*-haiku*)
            # These systems don't actually have a C or math library (as
such)
                continue
                ;;
        *-*-os2*)
            # These systems don't actually have a C library (as such)
            test "X$arg" = "X-lc" && continue
            ;;
        *-*-openbsd* | *-*-freebsd* | *-*-dragonfly*)
            # Do not include libc due to us having libc/libc_r.
            test "X$arg" = "X-lc" && continue
            ;;
        *-*-rhapsody* | *-*-darwin1.[012])
            # Rhapsody C and math libraries are in the System framework
            deplibs+=" System.ltframework"
            continue
            ;;
        *-*-sco3.2v5* | *-*-sco5v6*)
            # Causes problems with __ctype
            test "X$arg" = "X-lc" && continue
            ;;
        *-*-sysv4.2uw2* | *-*-sysv5* | *-*-unixware* | *-*-OpenUNIX*)
            # Compiler inserts libc in the correct place for threads to
work
                test "X$arg" = "X-lc" && continue
                ;;
    esac
elif test "X$arg" = "X-lc_r"; then
    case $host in
        *-*-openbsd* | *-*-freebsd* | *-*-dragonfly*)
            # Do not include libc_r directly, use -pthread flag.
            continue
            ;;
    esac
fi
deplibs+=" $arg"
continue
;;

-module)

```

```

module=yes
continue
;;

# Tru64 UNIX uses -model [arg] to determine the layout of C++
# classes, name mangling, and exception handling.
# Darwin uses the -arch flag to determine output architecture.
-model|-arch|-isysroot|--sysroot)
compiler_flags+=" $arg"
compile_command+=" $arg"
finalize_command+=" $arg"
prev=xcompiler
continue
;;

-mt|-mthreads|-kthread|-Kthread|-pthread|-pthreads|--thread-safe
\
|-threads|-fopenmp|-openmp|-mp|-xopenmp|-omp|-qomp=*)
compiler_flags+=" $arg"
compile_command+=" $arg"
finalize_command+=" $arg"
case "$new_inherited_linker_flags " in
    *" $arg "*) ;;
    * ) new_inherited_linker_flags+=" $arg" ;;
esac
continue
;;

-multi_module)
single_module="${wl}-multi_module"
continue
;;

-no-fast-install)
fast_install=no
continue
;;

-no-install)
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-darwin* |
*-cegcc*)
    # The PATH hackery in wrapper scripts is required on Windows
    # and Darwin in order for the loader to find any dlls it needs.
    func_warning "\`-no-install' is ignored for $host"
    func_warning "assuming \`-no-fast-install' instead"
    fast_install=no
    ;;
*) no_install=yes ;;
esac
continue
;;

```

```

-no-undefined)
allow_undefined=no
continue
;;

-objectlist)
prev=objectlist
continue
;;

-o) prev=output ;;

-precious-files-regex)
prev=precious_regex
continue
;;

-release)
prev=release
continue
;;

-rpath)
prev=rpath
continue
;;

-R)
prev=xrpath
continue
;;

-R*)
func_stripname '-R' '' "$arg"
dir=${func_stripname_result}
# We need an absolute path.
case $dir in
[\\/] * | [A-Za-z]:[\\/] *) ;;
=*)
    func_stripname '=' '' "$dir"
    dir=${lt_sysroot}${func_stripname_result}
    ;;
*)
    func_fatal_error "only absolute run-paths are allowed"
    ;;
esac
case "$xrpath " in
*" $dir "*) ;;
*) xrpath+="$ $dir" ;;
esac
continue

```

```

;;

-shared)
# The effects of -shared are defined in a previous loop.
continue
;;

-shrext)
prev=shrext
continue
;;

-static | -static-libtool-libs)
# The effects of -static are defined in a previous loop.
# We used to do the same as -all-static on platforms that
# didn't have a PIC flag, but the assumption that the effects
# would be equivalent was wrong. It would break on at least
# Digital Unix and AIX.
continue
;;

-thread-safe)
thread_safe=yes
continue
;;

-version-info)
prev=vinfo
continue
;;

-version-number)
prev=vinfo
vinfo_number=yes
continue
;;

-weak)
prev=weak
continue
;;

-Wc,*)
func_stripname '-Wc,' '' "$arg"
args=$func_stripname_result
arg=
save_ifs="$IFS"; IFS=', '
for flag in $args; do
IFS="$save_ifs"
func_quote_for_eval "$flag"
arg+=" $func_quote_for_eval_result"
compiler_flags+=" $func_quote_for_eval_result"

```

```

done
IFS="$save_ifs"
func_stripname ' ' ' "$arg"
arg=${func_stripname_result}
;;

-Wl,*)
func_stripname '-Wl,' ' ' "$arg"
args=${func_stripname_result}
arg=
save_ifs="$IFS"; IFS=', '
for flag in $args; do
    IFS="$save_ifs"
    func_quote_for_eval "$flag"
    arg+="${flag}"
    compiler_flags+="${flag}"
    linker_flags+="${flag}"
done
IFS="$save_ifs"
func_stripname ' ' ' "$arg"
arg=${func_stripname_result}
;;

-Xcompiler)
prev=xcompiler
continue
;;

-Xlinker)
prev=xlinker
continue
;;

-XCCLinker)
prev=xcclinker
continue
;;

# -msg_* for osf cc
-msg_*)
func_quote_for_eval "$arg"
arg="${func_quote_for_eval_result}"
;;

# Flags to be passed through unchanged, with rationale:
# -64, -mips[0-9]      enable 64-bit mode for the SGI compiler
# -r[0-9][0-9]*      specify processor for the SGI compiler
# -xarch=*, -xtarget=* enable 64-bit mode for the Sun compiler
# +DA*, +DD*          enable 64-bit mode for the HP compiler
# -q*                  compiler args for the IBM compiler
# -m*, -t[45]*, -txscale* architecture-specific flags for GCC

```

```

    # -F/path                path to uninstalled frameworks, gcc on
darwin
    # -p, -pg, --coverage, -fprofile-* profiling flags for GCC
    # -fstack-protector*    stack protector flags for GCC
    # @file                  GCC response files
    # -tp=*                  Portland pgcc target processor selection
    # --sysroot=*           for sysroot support
    # -O*, -flto*, -fwhopr*, -fuse-linker-plugin GCC link-time
optimization
    -64|-mips[0-9]|-r[0-9][0-9]*|-xarch=*|-xtarget=*|+DA*|+DD*|-q*|-
m*| \
    -t[45]*|-txscale*|-p|-pg|--coverage|-fprofile-*|-F*|@*|-tp=*|--
sysroot=*| \
    -O*|-flto*|-fwhopr*|-fuse-linker-plugin|-fstack-protector*)
    func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
    compile_command+=" $arg"
    finalize_command+=" $arg"
    compiler_flags+=" $arg"
    continue
    ;;

    # Some other compiler flag.
    -* | +*)
        func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
    ;;

    *.$objext)
    # A standard object.
objs+=" $arg"
    ;;

    *.lo)
    # A libtool-controlled object.

    # Check to see that this really is a libtool object.
if func_lalib_unsafe_p "$arg"; then
    pic_object=
    non_pic_object=

    # Read the .lo file
    func_source "$arg"

    if test -z "$pic_object" ||
        test -z "$non_pic_object" ||
        test "$pic_object" = none &&
        test "$non_pic_object" = none; then
        func_fatal_error "cannot find name of object for `\$arg'"
    fi

    # Extract subdirectory from the argument.

```

```

func_dirname "$arg" "/" ""
xdir="$func_dirname_result"

if test "$pic_object" != none; then
  # Prepend the subdirectory the object is found in.
  pic_object="$xdir$pic_object"

  if test "$prev" = dlfiles; then
    if test "$build_libtool_libs" = yes && test
"$dlopen_support" = yes; then
      dlfiles+="$pic_object"
      prev=
      continue
    else
      # If libtool objects are unsupported, then we need to
preload.
      prev=dlprefiles
    fi
  fi

  # CHECK ME: I think I busted this. -Ossama
  if test "$prev" = dlprefiles; then
    # Preload the old-style object.
    dlprefiles+="$pic_object"
    prev=
  fi

  # A PIC object.
  libobjs+="$pic_object"
  arg="$pic_object"
fi

# Non-PIC object.
if test "$non_pic_object" != none; then
  # Prepend the subdirectory the object is found in.
  non_pic_object="$xdir$non_pic_object"

  # A standard non-PIC object
  non_pic_objects+="$non_pic_object"
  if test -z "$pic_object" || test "$pic_object" = none ; then
    arg="$non_pic_object"
  fi
else
  # If the PIC object exists, use it instead.
  # $xdir was prepended to $pic_object above.
  non_pic_object="$pic_object"
  non_pic_objects+="$non_pic_object"
fi
else
  # Only an error if not doing a dry-run.
  if $opt_dry_run; then
    # Extract subdirectory from the argument.

```



```

func_dirname "$arg" "/" ""
xdir="$func_dirname_result"

func_lo2o "$arg"
pic_object=$xdir$objdir/$func_lo2o_result
non_pic_object=$xdir$func_lo2o_result
libobjs+=" $pic_object"
non_pic_objects+=" $non_pic_object"
else
func_fatal_error "\`$arg' is not a valid libtool object"
fi
fi
;;

*.$libext)
# An archive.
deplibs+=" $arg"
old_deplibs+=" $arg"
continue
;;

*.la)
# A libtool-controlled library.

func_resolve_sysroot "$arg"
if test "$prev" = dlfiles; then
# This library was specified with -dlopen.
dlfiles+=" $func_resolve_sysroot_result"
prev=
elif test "$prev" = dlprefiles; then
# The library was specified with -dlpreopen.
dlprefiles+=" $func_resolve_sysroot_result"
prev=
else
deplibs+=" $func_resolve_sysroot_result"
fi
continue
;;

# Some other compiler argument.
*)
# Unknown arguments in both finalize_command and compile_command
need
# to be aesthetically quoted because they are eval'd later.
func_quote_for_eval "$arg"
arg="$func_quote_for_eval_result"
;;
esac # arg

# Now actually substitute the argument into the commands.
if test -n "$arg"; then
compile_command+=" $arg"

```

```

        finalize_command+=" $arg"
    fi
done # argument parsing loop

test -n "$prev" && \
    func_fatal_help "the \`$prevarg' option requires an argument"

if test "$export_dynamic" = yes && test -n
"$export_dynamic_flag_spec"; then
    eval arg="$export_dynamic_flag_spec\"
    compile_command+=" $arg"
    finalize_command+=" $arg"
fi

oldlibs=
# calculate the name of the file, without its directory
func_basename "$output"
outputname="$func_basename_result"
libobjs_save="$libobjs"

if test -n "$shlibpath_var"; then
    # get the directories listed in $shlibpath_var
    eval shlib_search_path=\`$ECHO \`${$shlibpath_var} \| \| \`$SED
\`s/:// /g\`\'\'
else
    shlib_search_path=
fi
eval sys_lib_search_path=\`$sys_lib_search_path_spec\'
eval sys_lib_dlsearch_path=\`$sys_lib_dlsearch_path_spec\'

func_dirname "$output" "/" ""
output_objdir="$func_dirname_result$objdir"
func_to_tool_file "$output_objdir/"
tool_output_objdir=$func_to_tool_file_result
# Create the object directory.
func_mkdir_p "$output_objdir"

# Determine the type of output
case $output in
    "")
        func_fatal_help "you must specify an output file"
        ;;
    *.$libext) linkmode=oldlib ;;
    *.lo | *.$objext) linkmode=obj ;;
    *.la) linkmode=lib ;;
    *) linkmode=prog ;; # Anything else should be a program.
esac

specialdeplibs=

libs=
# Find all interdependent deplibs by searching for libraries

```

```

# that are linked more than once (e.g. -la -lb -la)
for deplib in $deplibs; do
  if $opt_preserve_dup_deps ; then
    case "$libs " in
      *" $deplib ") specialdeplibs+=" $deplib" ;;
    esac
  fi
  libs+=" $deplib"
done

if test "$linkmode" = lib; then
  libs="$predeps $libs $compiler_lib_search_path $postdeps"

  # Compute libraries that are listed more than once in $predeps
  # $postdeps and mark them as special (i.e., whose duplicates are
  # not to be eliminated).
  pre_post_deps=
  if $opt_duplicate_compiler_generated_deps; then
    for pre_post_dep in $predeps $postdeps; do
      case "$pre_post_deps " in
        *" $pre_post_dep ") specialdeplibs+=" $pre_post_deps" ;;
      esac
      pre_post_deps+=" $pre_post_dep"
    done
  fi
  pre_post_deps=
fi

deplibs=
newdependency_libs=
newlib_search_path=
need_relink=no # whether we're linking any uninstalled libtool
libraries
notinst_deplibs= # not-installed libtool libraries
notinst_path= # paths that contain not-installed libtool libraries

case $linkmode in
lib)
  passes="conv dlopen link"
  for file in $dlfiles $dlprefiles; do
    case $file in
      *.la) ;;
      *)
        func_fatal_help "libraries can \`-dlopen' only libtool
libraries: $file"
        ;;
    esac
  done
  ;;
prog)
  compile_deplibs=
  finalize_deplibs=

```

```

alldeplibs=no
newdlfiles=
newdlprefiles=
passes="conv scan dlopen dlpreopen link"
;;
*) passes="conv"
;;
esac

for pass in $passes; do
# The preopen pass in lib mode reverses $deplibs; put it back
here
# so that -L comes before libs that need it for instance...
if test "$linkmode,$pass" = "lib,link"; then
## FIXME: Find the place where the list is rebuilt in the wrong
##          order, and fix it there properly
tmp_deplibs=
for deplib in $deplibs; do
tmp_deplibs="$deplib $tmp_deplibs"
done
deplibs="$tmp_deplibs"
fi

if test "$linkmode,$pass" = "lib,link" ||
test "$linkmode,$pass" = "prog,scan"; then
libs="$deplibs"
deplibs=
fi
if test "$linkmode" = prog; then
case $pass in
dlopen) libs="$dlfiles" ;;
dlpreopen) libs="$dlprefiles" ;;
link) libs="$deplibs %DEPLIBS% $dependency_libs" ;;
esac
fi
if test "$linkmode,$pass" = "lib,dlpreopen"; then
# Collect and forward deplibs of preopened libtool libs
for lib in $dlprefiles; do
# Ignore non-libtool-libs
dependency_libs=
func_resolve_sysroot "$lib"
case $lib in
*.la) func_source "$func_resolve_sysroot_result" ;;
esac

# Collect preopened libtool deplibs, except any this library
# has declared as weak libs
for deplib in $dependency_libs; do
func_basename "$deplib"
deplib_base=$func_basename_result
case " $weak_libs " in
*" $deplib_base "*) ;;

```

```

        *) deplibs+=" $deplib" ;;
    esac
done
done
libs="$dlprefiles"
fi
if test "$pass" = dlopen; then
# Collect dlpreopened libraries
save_deplibs="$deplibs"
deplibs=
fi

for deplib in $libs; do
lib=
found=no
case $deplib in
-mt|-mthreads|-kthread|-Kthread|-pthread|-pthreads|--thread-safe
\
|-threads|-fopenmp|-openmp|-mp|-xopenmp|-omp|-qsmpr=*)
if test "$linkmode,$pass" = "prog,link"; then
compile_deplibs="$deplib $compile_deplibs"
finalize_deplibs="$deplib $finalize_deplibs"
else
compiler_flags+=" $deplib"
if test "$linkmode" = lib ; then
case "$new_inherited_linker_flags " in
*" $deplib ") ;;
* ) new_inherited_linker_flags+=" $deplib" ;;
esac
fi
fi
continue
;;
-l*)
if test "$linkmode" != lib && test "$linkmode" != prog; then
func_warning "`-l' is ignored for archives/objects"
continue
fi
func_stripname '-l' '' "$deplib"
name=$func_stripname_result
if test "$linkmode" = lib; then
searchdirs="$newlib_search_path $lib_search_path
$compiler_lib_search_dirs $sys_lib_search_path $shlib_search_path"
else
searchdirs="$newlib_search_path $lib_search_path
$sys_lib_search_path $shlib_search_path"
fi
for searchdir in $searchdirs; do
for search_ext in .la $std_shrext .so .a; do
# Search the libtool library
lib="$searchdir/lib${name}${search_ext}"
if test -f "$lib"; then

```

```

        if test "$search_ext" = ".la"; then
            found=yes
        else
            found=no
        fi
        break 2
    fi
done
done
if test "$found" != yes; then
    # deplib doesn't seem to be a libtool library
    if test "$linkmode,$pass" = "prog,link"; then
        compile_deplibs="$deplib $compile_deplibs"
        finalize_deplibs="$deplib $finalize_deplibs"
    else
        deplibs="$deplib $deplibs"
        test "$linkmode" = lib && newdependency_libs="$deplib
$newdependency_libs"
    fi
    continue
else # deplib is a libtool library
    # If $allow_libtool_libs_with_static_runtimes && $deplib is a
stdlib,
    # We need to do some special things here, and not later.
    if test "X$allow_libtool_libs_with_static_runtimes" = "Xyes"
; then
        case " $predeps $postdeps " in
        *" $deplib ")
            if func_lalib_p "$lib"; then
                library_names=
                old_library=
                func_source "$lib"
                for l in $old_library $library_names; do
                    ll="$l"
                done
                if test "X$ll" = "X$old_library" ; then # only static
version available
                    found=no
                    func_dirname "$lib" "" "."
                    ladir="$func_dirname_result"
                    lib=$ladir/$old_library
                    if test "$linkmode,$pass" = "prog,link"; then
                        compile_deplibs="$deplib $compile_deplibs"
                        finalize_deplibs="$deplib $finalize_deplibs"
                    else
                        deplibs="$deplib $deplibs"
                        test "$linkmode" = lib && newdependency_libs="$deplib
$newdependency_libs"
                    fi
                    continue
                fi
            fi
        fi
    fi
fi

```

```

        ;;
        *) ;;
        esac
    fi
fi
;; # -l
*.ltframework)
if test "$linkmode,$$pass" = "prog,link"; then
    compile_deplibs="$deplib $compile_deplibs"
    finalize_deplibs="$deplib $finalize_deplibs"
else
    deplibs="$deplib $deplibs"
    if test "$linkmode" = lib ; then
        case "$new_inherited_linker_flags " in
            *" $deplib "*) ;;
            * ) new_inherited_linker_flags+=" $deplib" ;;
        esac
    fi
fi
continue
;;
-L*)
case $linkmode in
lib)
    deplibs="$deplib $deplibs"
    test "$pass" = conv && continue
    newdependency_libs="$deplib $newdependency_libs"
    func_stripname '-L' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    newlib_search_path+=" $func_resolve_sysroot_result"
    ;;
prog)
    if test "$pass" = conv; then
        deplibs="$deplib $deplibs"
        continue
    fi
    if test "$pass" = scan; then
        deplibs="$deplib $deplibs"
    else
        compile_deplibs="$deplib $compile_deplibs"
        finalize_deplibs="$deplib $finalize_deplibs"
    fi
    func_stripname '-L' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    newlib_search_path+=" $func_resolve_sysroot_result"
    ;;
*)
    func_warning "\`-L' is ignored for archives/objects"
    ;;
esac # linkmode
continue
;; # -L

```

```

-R*)
  if test "$pass" = link; then
    func_stripname '-R' '' "$deplib"
    func_resolve_sysroot "$func_stripname_result"
    dir=$func_resolve_sysroot_result
    # Make sure the xrrpath contains only unique directories.
    case "$xrrpath " in
      *" $dir "*) ;;
      *) xrrpath+=" $dir" ;;
    esac
  fi
  deplibs="$deplib $deplibs"
  continue
;;
*.la)
  func_resolve_sysroot "$deplib"
  lib=$func_resolve_sysroot_result
;;
*.$libext)
  if test "$pass" = conv; then
    deplibs="$deplib $deplibs"
    continue
  fi
  case $linkmode in
  lib)
    # Linking convenience modules into shared libraries is
allowed,
    # but linking other static libraries is non-portable.
    case " $dlpreconveniencelibs " in
      *" $deplib "*) ;;
      *)
        valid_a_lib=no
        case $deplibs_check_method in
        match_pattern*)
          set dummy $deplibs_check_method; shift
          match_pattern_regex=`expr "$deplibs_check_method" : "$1
\(.*\)"`
          if eval "\$ECHO \"$deplib\" 2>/dev/null | $SED 10q \
| $EGREP "$match_pattern_regex" > /dev/null; then
            valid_a_lib=yes
          fi
        ;;
        pass_all)
          valid_a_lib=yes
        ;;
        esac
        if test "$valid_a_lib" != yes; then
          echo
          $ECHO "*** Warning: Trying to link with static lib archive
$deplib."
          echo "*** I have the capability to make that library
automatically link in when"

```



```

        echo "*** you link to this library. But I can only do this
if you have a"
        echo "*** shared version of the library, which you do not
appear to have"
        echo "*** because the file extensions .$libext of this
argument makes me believe"
        echo "*** that it is just a static archive that I should
not use here."
        else
        echo
        $ECHO "*** Warning: Linking the shared library $output
against the"
        $ECHO "*** static library $deplib is not portable!"
        deplibs="$deplib $deplibs"
        fi
        ;;
    esac
    continue
    ;;
prog)
    if test "$pass" != link; then
        deplibs="$deplib $deplibs"
    else
        compile_deplibs="$deplib $compile_deplibs"
        finalize_deplibs="$deplib $finalize_deplibs"
    fi
    continue
    ;;
esac # linkmode
;; # *.$libext
*.lo | *.$objext)
    if test "$pass" = conv; then
        deplibs="$deplib $deplibs"
    elif test "$linkmode" = prog; then
        if test "$pass" = dlpreopen || test "$dlopen_support" != yes
|| test "$build_libtool_libs" = no; then
            # If there is no dlopen support or we're linking
statically,
            # we need to preload.
            newdlprefiles+=" $deplib"
            compile_deplibs="$deplib $compile_deplibs"
            finalize_deplibs="$deplib $finalize_deplibs"
        else
            newdlfiles+=" $deplib"
        fi
    fi
    continue
    ;;
%DEPLIBS%)
    alldeplibs=yes
    continue
    ;;

```

```

esac # case $deplib

if test "$found" = yes || test -f "$lib"; then :
else
    func_fatal_error "cannot find the library \`$lib' or unhandled
argument \`$deplib'"
fi

# Check to see that this really is a libtool archive.
func_lalib_unsafe_p "$lib" \
    || func_fatal_error "\`$lib' is not a valid libtool archive"

func_dirname "$lib" "" "."
ladir="$func_dirname_result"

dlname=
dlopen=
dlpreopen=
libdir=
library_names=
old_library=
inherited_linker_flags=
# If the library was installed with an old release of libtool,
# it will not redefine variables installed, or shouldnotlink
installed=yes
shouldnotlink=no
avoidtempdirpath=

# Read the .la file
func_source "$lib"

# Convert "-framework foo" to "foo.ltframework"
if test -n "$inherited_linker_flags"; then
    tmp_inherited_linker_flags=`$ECHO "$inherited_linker_flags" |
$SED 's/-framework \([^ $]*\)\/\1.ltframework/g'`
    for tmp_inherited_linker_flag in $tmp_inherited_linker_flags;
do
        case " $new_inherited_linker_flags " in
            *" $tmp_inherited_linker_flag ") ;;
            *) new_inherited_linker_flags+="
$tmp_inherited_linker_flag";;
        esac
    done
fi
dependency_libs=`$ECHO " $dependency_libs" | $SED 's% \([^
$]*\)\.ltframework% -framework \1%g'`
if test "$linkmode,$pass" = "lib,link" ||
    test "$linkmode,$pass" = "prog,scan" ||
    { test "$linkmode" != prog && test "$linkmode" != lib; }; then
    test -n "$dlopen" && dlfiles+="$dlopen"
    test -n "$dlpreopen" && dlprefiles+="$dlpreopen"

```

```

fi

if test "$pass" = conv; then
  # Only check for convenience libraries
  deplibs="$lib $deplibs"
  if test -z "$libdir"; then
    if test -z "$old_library"; then
      func_fatal_error "cannot find name of link library for
\`$lib'"
    fi
    # It is a libtool convenience library, so add in its objects.
    convenience+=" $ladir/$objdir/$old_library"
    old_convenience+=" $ladir/$objdir/$old_library"
  elif test "$linkmode" != prog && test "$linkmode" != lib; then
    func_fatal_error "\`$lib' is not a convenience library"
  fi
  tmp_libs=
  for deplib in $dependency_libs; do
    deplibs="$deplib $deplibs"
    if $opt_preserve_dup_deps ; then
      case "$tmp_libs " in
        *" $deplib "*) specialdeplibs+=" $deplib" ;;
      esac
    fi
    tmp_libs+=" $deplib"
  done
  continue
fi # $pass = conv

# Get the name of the library we link against.
linklib=
if test -n "$old_library" &&
  { test "$prefer_static_libs" = yes ||
  test "$prefer_static_libs,$installed" = "built,no"; }; then
  linklib=$old_library
else
  for l in $old_library $library_names; do
    linklib="$l"
  done
fi
if test -z "$linklib"; then
  func_fatal_error "cannot find name of link library for \`$lib'"
fi

# This library was specified with -dlopen.
if test "$pass" = dlopen; then
  if test -z "$libdir"; then
    func_fatal_error "cannot -dlopen a convenience library:
\`$lib'"
  fi
  if test -z "$dlname" ||

```

```

        test "$dlopen_support" != yes ||
        test "$build_libtool_libs" = no; then
# If there is no dlname, no dlopen support or we're linking
# statically, we need to preload.  We also need to preload
any
# dependent libraries so libltdl's deplib preloader doesn't
# bomb out in the load deplibs phase.
        dlprefiles+=" $lib $dependency_libs"
    else
        newdlfiles+=" $lib"
    fi
    continue
fi # $pass = dlopen

# We need an absolute path.
case $ladir in
[\\/] * | [A-Za-z]:[\\/]*) abs_ladir="$ladir" ;;
*)
    abs_ladir=`cd "$ladir" && pwd`
    if test -z "$abs_ladir"; then
        func_warning "cannot determine absolute directory name of
\`$ladir'"
        func_warning "passing it literally to the linker, although it
might fail"
        abs_ladir="$ladir"
    fi
    ;;
esac
func_basename "$lib"
laname="$func_basename_result"

# Find the relevant object directory and library name.
if test "X$installed" = Xyes; then
    if test ! -f "$lt_sysroot$libdir/$linklib" && test -f
"$abs_ladir/$linklib"; then
        func_warning "library \`$lib' was moved."
        dir="$ladir"
        absdir="$abs_ladir"
        libdir="$abs_ladir"
    else
        dir="$lt_sysroot$libdir"
        absdir="$lt_sysroot$libdir"
    fi
    test "X$hardcode_automatic" = Xyes && avoidtemprpath=yes
else
    if test ! -f "$ladir/$objdir/$linklib" && test -f
"$abs_ladir/$linklib"; then
        dir="$ladir"
        absdir="$abs_ladir"
        # Remove this search path later
        notinst_path+=" $abs_ladir"
    else

```

```

    dir="$ladir/$objdir"
    absdir="$abs_ladir/$objdir"
    # Remove this search path later
    notinst_path+=" $abs_ladir"
fi
fi # $installed = yes
func_stripname 'lib' '.la' "$laname"
name=$func_stripname_result

# This library was specified with -dlpreopen.
if test "$pass" = dlpreopen; then
    if test -z "$libdir" && test "$linkmode" = prog; then
        func_fatal_error "only libraries may -dlpreopen a convenience
library: \`$lib'"
    fi
    case "$host" in
        # special handling for platforms with PE-DLLs.
        *cygwin* | *mingw* | *cegcc* )
            # Linker will automatically link against shared library if
both
            # static and shared are present.  Therefore, ensure we
extract
            # symbols from the import library if a shared library is
present
            # (otherwise, the dlopen module name will be incorrect).
We do
            # this by putting the import library name into
$newdlprefiles.
            # We recover the dlopen module name by 'saving' the la file
            # name in a special purpose variable, and (later)
extracting the
            # dlname from the la file.
            if test -n "$dlname"; then
                func_tr_sh "$dir/$linklib"
                eval "libfile_$func_tr_sh_result=\$abs_ladir/\$laname"
                newdlprefiles+=" $dir/$linklib"
            else
                newdlprefiles+=" $dir/$old_library"
                # Keep a list of preopened convenience libraries to check
                # that they are being used correctly in the link pass.
                test -z "$libdir" && \
                    dlpreconveniencelibs+=" $dir/$old_library"
            fi
        ;;
        * )
            # Prefer using a static library (so that no silly _DYNAMIC
symbols
            # are required to link).
            if test -n "$old_library"; then
                newdlprefiles+=" $dir/$old_library"
                # Keep a list of preopened convenience libraries to check
                # that they are being used correctly in the link pass.

```

```

        test -z "$libdir" && \
            dlpreconveniencelibs+=" $dir/$old_library"
# Otherwise, use the dlname, so that lt_dlopen finds it.
elif test -n "$dlname"; then
    newdlprefiles+=" $dir/$dlname"
else
    newdlprefiles+=" $dir/$linklib"
fi
;;
esac
fi # $pass = dlpreopen

if test -z "$libdir"; then
# Link the convenience library
if test "$linkmode" = lib; then
    deplibs="$dir/$old_library $deplibs"
elif test "$linkmode,$pass" = "prog,link"; then
    compile_deplibs="$dir/$old_library $compile_deplibs"
    finalize_deplibs="$dir/$old_library $finalize_deplibs"
else
    deplibs="$lib $deplibs" # used for prog,scan pass
fi
continue
fi

if test "$linkmode" = prog && test "$pass" != link; then
    newlib_search_path+=" $ladir"
    deplibs="$lib $deplibs"

    linkalldeplibs=no
    if test "$link_all_deplibs" != no || test -z "$library_names"
||
        test "$build_libtool_libs" = no; then
        linkalldeplibs=yes
    fi

    tmp_libs=
    for deplib in $dependency_libs; do
        case $deplib in
        -L*) func_stripname '-L' '' "$deplib"
            func_resolve_sysroot "$func_stripname_result"
            newlib_search_path+=" $func_resolve_sysroot_result"
            ;;
        esac
    # Need to link against all dependency_libs?
    if test "$linkalldeplibs" = yes; then
        deplibs="$deplib $deplibs"
    else
        # Need to hardcode shared library paths
        # or/and link against static libraries
        newdependency_libs="$deplib $newdependency_libs"
    fi

```

```

fi
if $opt_preserve_dup_deps ; then
  case "$tmp_libs " in
    *" $deplib ") specialdeplibs+=" $deplib" ;;
  esac
fi
tmp_libs+=" $deplib"
done # for deplib
continue
fi # $linkmode = prog...

if test "$linkmode,$pass" = "prog,link"; then
  if test -n "$library_names" &&
    { { test "$prefer_static_libs" = no ||
      test "$prefer_static_libs,$installed" = "built,yes"; }
  ||
    test -z "$old_library"; }; then
    # We need to hardcode the library path
    if test -n "$shlibpath_var" && test -z "$avoidtemp_rpath" ;
  then
    # Make sure the rpath contains only unique directories.
    case "$temp_rpath:" in
      *"$absdir:") ;;
      *) temp_rpath+=" $absdir:" ;;
    esac
  fi

  # Hardcode the library path.
  # Skip directories that are in the system default run-time
  # search path.
  #case " $sys_lib_dlsearch_path " in
  #*" $absdir ") ;;
  #*)
  # case "$compile_rpath " in
  # *" $absdir ") ;;
  # *) compile_rpath+=" $absdir" ;;
  # esac
  # ;;
  #esac
  case " $sys_lib_dlsearch_path " in
    *" $libdir ") ;;
    *)
      case "$finalize_rpath " in
        *" $libdir ") ;;
        *) finalize_rpath+=" $libdir" ;;
      esac
    ;;
  esac
fi # $linkmode,$pass = prog,link...

if test "$alldeplibs" = yes &&
  { test "$deplibs_check_method" = pass_all ||

```

```

        { test "$build_libtool_libs" = yes &&
          test -n "$library_names"; }; }; then
    # We only need to search for static libraries
    continue
  fi
fi
link_static=no # Whether the deplib will be linked statically
use_static_libs=$prefer_static_libs
if test "$use_static_libs" = built && test "$installed" = yes;
then
  use_static_libs=no
fi
if test -n "$library_names" &&
  { test "$use_static_libs" = no || test -z "$old_library"; };
then
  case $host in
    *cygwin* | *mingw* | *cegcc*)
      # No point in relinking DLLs because paths are not encoded
      notinst_deplibs+="$lib"
      need_relink=no
      ;;
    *)
      if test "$installed" = no; then
        notinst_deplibs+="$lib"
        need_relink=yes
      fi
      ;;
  esac
  # This is a shared library

  # Warn about portability, can't link against -module's on some
  # systems (darwin). Don't bleat about dlopened modules though!
  dlopenmodule=""
  for dlpremoduletest in $dlprefiles; do
    if test "X$dlpremoduletest" = "X$lib"; then
      dlopenmodule="$dlpremoduletest"
      break
    fi
  done
  if test -z "$dlopenmodule" && test "$shouldnotlink" = yes &&
test "$pass" = link; then
    echo
    if test "$linkmode" = prog; then
      $ECHO "*** Warning: Linking the executable $output against
the loadable module"
    else
      $ECHO "*** Warning: Linking the shared library $output
against the loadable module"
    fi
    $ECHO "*** $linklib is not portable!"
  fi
fi

```



```

if test "$linkmode" = lib &&
    test "$hardcode_into_libs" = yes; then
    # Hardcode the library path.
    # Skip directories that are in the system default run-time
    # search path.
    #case " $sys_lib_dlsearch_path " in
    #" $absdir ") ;;
    #*)
    # case "$compile_rpath " in
    # *) $absdir *) ;;
    # *) compile_rpath+=" $absdir" ;;
    # esac
    # ;;
    #esac
    case " $sys_lib_dlsearch_path " in
    *) $libdir *) ;;
    *)
        case "$finalize_rpath " in
        *) $libdir *) ;;
        *) finalize_rpath+=" $libdir" ;;
        esac
        ;;
    esac
fi

```

```

if test -n "$old_archive_from_expsyms_cmds"; then
    # figure out the soname
    set dummy $library_names
    shift
    realname="$1"
    shift
    libname=`eval "\\$ECHO \"\$libname_spec\""`
    # use dlname if we got it. it's perfectly good, no?
    if test -n "$dlname"; then
        soname="$dlname"
    elif test -n "$soname_spec"; then
        # bleh windows
        case $host in
        *cygwin* | mingw* | *cegcc*)
            func_arith $current - $age
            major=$func_arith_result
            versuffix="-${major}"
            ;;
        esac
        eval soname="\$soname_spec"
    else
        soname="$realname"
    fi

```

```

# Make a new name for the extract_expsyms_cmds to use
soroot="$soname"
func_basename "$soroot"

```

```

soname="$func_basename_result"
func_stripname 'lib' '.dll' "$soname"
newlib=libimp-$func_stripname_result.a

# If the library has no export list, then create one now
if test -f "$output_objdir/$soname-def"; then :
else
    func_verbose "extracting exported symbol list from
\`$soname'"
    func_execute_cmds "$extract_expsyms_cmds" 'exit $?'
fi

# Create $newlib
if test -f "$output_objdir/$newlib"; then ;; else
    func_verbose "generating import library for \`$soname'"
    func_execute_cmds "$old_archive_from_expsyms_cmds" 'exit
$?'
fi

# make sure the library variables are pointing to the new
library
dir=$output_objdir
linklib=$newlib
fi # test -n "$old_archive_from_expsyms_cmds"

if test "$linkmode" = prog || test "$opt_mode" != relink; then
add_shlibpath=
add_dir=
add=
lib_linked=yes
case $hardcode_action in
immediate | unsupported)
    if test "$hardcode_direct" = no; then
add="$dir/$linklib"
case $host in
*-*-sco3.2v5.0.[024]*) add_dir="-L$dir" ;;
*-*-sysv4*uw2*) add_dir="-L$dir" ;;
*-*-sysv5OpenUNIX* | *-*-sysv5UnixWare7.[01].[10]* | \
*-*-unixware7*) add_dir="-L$dir" ;;
*-*-darwin* )
    # if the lib is a (non-dlopened) module then we can not
    # link against it, someone is ignoring the earlier
warnings
    if /usr/bin/file -L $add 2> /dev/null |
    $GREP ": [^:]* bundle" >/dev/null ; then
    if test "X$dlopenmodule" != "X$lib"; then
    $ECHO "*** Warning: lib $linklib is a module, not a
shared library"
    if test -z "$old_library" ; then
    echo
    echo "*** And there doesn't seem to be a static
archive available"
    echo "*** The link will probably fail, sorry"

```

```

        else
            add="$dir/$old_library"
        fi
        elif test -n "$old_library"; then
            add="$dir/$old_library"
        fi
    fi
esac
elif test "$hardcode_minus_L" = no; then
case $host in
*-*-sunos*) add_shlibpath="$dir" ;;
esac
add_dir="-L$dir"
add="-l$name"
elif test "$hardcode_shlibpath_var" = no; then
add_shlibpath="$dir"
add="-l$name"
else
lib_linked=no
fi
;;
relink)
if test "$hardcode_direct" = yes &&
test "$hardcode_direct_absolute" = no; then
add="$dir/$linklib"
elif test "$hardcode_minus_L" = yes; then
add_dir="-L$absdir"
# Try looking first in the location we're being installed
to.
if test -n "$inst_prefix_dir"; then
case $libdir in
[\\/]*)
add_dir+=" -L$inst_prefix_dir$libdir"
;;
esac
fi
add="-l$name"
elif test "$hardcode_shlibpath_var" = yes; then
add_shlibpath="$dir"
add="-l$name"
else
lib_linked=no
fi
;;
*) lib_linked=no ;;
esac

if test "$lib_linked" != yes; then
func_fatal_configuration "unsupported hardcode properties"
fi

if test -n "$add_shlibpath"; then

```

```

        case :$compile_shlibpath: in
        *":$add_shlibpath:") ;;
        *) compile_shlibpath+=" $add_shlibpath:" ;;
        esac
    fi
    if test "$linkmode" = prog; then
        test -n "$add_dir" && compile_deplibs="$add_dir
$compile_deplibs"
        test -n "$add" && compile_deplibs="$add $compile_deplibs"
    else
        test -n "$add_dir" && deplibs="$add_dir $deplibs"
        test -n "$add" && deplibs="$add $deplibs"
        if test "$hardcode_direct" != yes &&
            test "$hardcode_minus_L" != yes &&
            test "$hardcode_shlibpath_var" = yes; then
            case :$finalize_shlibpath: in
            *":$libdir:") ;;
            *) finalize_shlibpath+=" $libdir:" ;;
            esac
        fi
    fi
fi

if test "$linkmode" = prog || test "$opt_mode" = relink; then
    add_shlibpath=
    add_dir=
    add=
    # Finalize command for both is simple: just hardcode it.
    if test "$hardcode_direct" = yes &&
        test "$hardcode_direct_absolute" = no; then
        add="$libdir/$linklib"
    elif test "$hardcode_minus_L" = yes; then
        add_dir="-L$libdir"
        add="-l$name"
    elif test "$hardcode_shlibpath_var" = yes; then
        case :$finalize_shlibpath: in
        *":$libdir:") ;;
        *) finalize_shlibpath+=" $libdir:" ;;
        esac
        add="-l$name"
    elif test "$hardcode_automatic" = yes; then
        if test -n "$inst_prefix_dir" &&
            test -f "$inst_prefix_dir$libdir/$linklib" ; then
            add="$inst_prefix_dir$libdir/$linklib"
        else
            add="$libdir/$linklib"
        fi
    else
        # We cannot seem to hardcode it, guess we'll fake it.
        add_dir="-L$lt_sysroot$libdir"
        # Try looking first in the location we're being installed
to.

```

```

        if test -n "$inst_prefix_dir"; then
        case $libdir in
            [\\/]*)
                add_dir+=" -L$inst_prefix_dir$libdir"
                ;;
        esac
        fi
        add="-l$name"
    fi

    if test "$linkmode" = prog; then
        test -n "$add_dir" && finalize_deplibs="$add_dir
$finalize_deplibs"
        test -n "$add" && finalize_deplibs="$add $finalize_deplibs"
    else
        test -n "$add_dir" && deplibs="$add_dir $deplibs"
        test -n "$add" && deplibs="$add $deplibs"
    fi
fi
elif test "$linkmode" = prog; then
# Here we assume that one of hardcode_direct or
hardcode_minus_L
# is not unsupported.  This is valid on all known static and
# shared platforms.
if test "$hardcode_direct" != unsupported; then
    test -n "$old_library" && linklib="$old_library"
    compile_deplibs="$dir/$linklib $compile_deplibs"
    finalize_deplibs="$dir/$linklib $finalize_deplibs"
else
    compile_deplibs="-l$name -L$dir $compile_deplibs"
    finalize_deplibs="-l$name -L$dir $finalize_deplibs"
fi
elif test "$build_libtool_libs" = yes; then
# Not a shared library
if test "$deplibs_check_method" != pass_all; then
# We're trying link a shared library against a static one
# but the system doesn't support it.

# Just print a warning and add the library to dependency_libs
so
# that the program can be linked against the static library.
echo
$ECHO "*** Warning: This system can not link to static lib
archive $lib."
    echo "*** I have the capability to make that library
automatically link in when"
    echo "*** you link to this library.  But I can only do this
if you have a"
    echo "*** shared version of the library, which you do not
appear to have."
    if test "$module" = yes; then

```

```

        echo "*** But as you try to build a module library, libtool
will still create "
        echo "*** a static module, that should work as long as the
dlopening application"
        echo "*** is linked with the -dlopen flag to resolve
symbols at runtime."
        if test -z "$global_symbol_pipe"; then
        echo
        echo "*** However, this would only work if libtool was able
to extract symbol"
        echo "*** lists from a program, using `nm' or equivalent,
but libtool could"
        echo "*** not find such a program. So, this module is
probably useless."
        echo "*** `nm' from GNU binutils and a full rebuild may
help."
        fi
        if test "$build_old_libs" = no; then
        build_libtool_libs=module
        build_old_libs=yes
        else
        build_libtool_libs=no
        fi
    fi
else
    deplibs="$dir/$old_library $deplibs"
    link_static=yes
fi
fi # link shared/static library?

if test "$linkmode" = lib; then
    if test -n "$dependency_libs" &&
    { test "$hardcode_into_libs" != yes ||
      test "$build_old_libs" = yes ||
      test "$link_static" = yes; }; then
    # Extract -R from dependency_libs
    temp_deplibs=
    for libdir in $dependency_libs; do
        case $libdir in
        -R*) func_stripname '-R' '' "$libdir"
            temp_xrpath=$func_stripname_result
            case " $xrpath " in
            *" $temp_xrpath ") ;;
            *) xrpath+=" $temp_xrpath";;
            esac;;
        *) temp_deplibs+=" $libdir";;
        esac
    done
    dependency_libs="$temp_deplibs"
fi

newlib_search_path+=" $absdir"

```

```

# Link against this library
test "$link_static" = no &&
newdependency_libs="$abs_ladir/$lname $newdependency_libs"
# ... and its dependency_libs
tmp_libs=
for deplib in $dependency_libs; do
  newdependency_libs="$deplib $newdependency_libs"
  case $deplib in
    -L*) func_stripname '-L' '' "$deplib"
          func_resolve_sysroot "$func_stripname_result";;
    *) func_resolve_sysroot "$deplib" ;;
  esac
  if $opt_preserve_dup_deps ; then
    case "$tmp_libs " in
      *" $func_resolve_sysroot_result "*)
        specialdeplibs+=" $func_resolve_sysroot_result" ;;
    esac
  fi
  tmp_libs+=" $func_resolve_sysroot_result"
done

if test "$link_all_deplibs" != no; then
# Add the search paths of all dependency libraries
for deplib in $dependency_libs; do
  path=
  case $deplib in
    -L*) path="$deplib" ;;
    *.la)
      func_resolve_sysroot "$deplib"
      deplib=$func_resolve_sysroot_result
      func_dirname "$deplib" "" "."
      dir=$func_dirname_result
      # We need an absolute path.
      case $dir in
        [\\/]*) | [A-Za-z]:[\\/]*) absdir="$dir" ;;
      *)
        absdir=`cd "$dir" && pwd`
        if test -z "$absdir"; then
          func_warning "cannot determine absolute directory name
of `\$dir'"
          absdir="$dir"
        fi
      ;;
    esac
  if $GREP "^installed=no" $deplib > /dev/null; then
    case $host in
      *-*-darwin*)
        depdepl=
        eval deplib_names=`${SED} -n -e
's/^library_names=(.*)$/\1/p' $deplib`
        if test -n "$deplib_names" ; then
          for tmp in $deplib_names ; do

```

```

        depdepl=$tmp
    done
    if test -f "$absdir/$objdir/$depdepl" ; then
        depdepl="$absdir/$objdir/$depdepl"
        darwin_install_name=`${OTOOL} -L $depdepl | awk '{if
(NR == 2) {print $1;exit}}'`
        if test -z "$darwin_install_name"; then
            darwin_install_name=`${OTOOL64} -L $depdepl
| awk '{if (NR == 2) {print $1;exit}}'`
        fi
        compiler_flags+=" ${wl}-dylib_file
${wl}${darwin_install_name}:${depdepl}"
        linker_flags+=" -dylib_file
${darwin_install_name}:${depdepl}"
        path=
    fi
    fi
    ;;
*)
    path="-L$absdir/$objdir"
    ;;
esac
else
    eval libdir=`${SED} -n -e 's/^libdir=\(.*\)$/\1/p'
$deplib`
    test -z "$libdir" && \
        func_fatal_error "`$deplib' is not a valid libtool
archive"
    #test "$absdir" != "$libdir" && \
    # func_warning "`$deplib' seems to be moved"

    path="-L$absdir"
    fi
    ;;
esac
case " $deplibs " in
*" $path ") ;;
*) deplibs="$path $deplibs" ;;
esac
done
fi # link_all_deplibs != no
fi # linkmode = lib
done # for deplib in $libs
if test "$pass" = link; then
if test "$linkmode" = "prog"; then
    compile_deplibs="$new_inherited_linker_flags $compile_deplibs"
    finalize_deplibs="$new_inherited_linker_flags
$finalize_deplibs"
else
    compiler_flags="$compiler_flags "`$ECHO "
$new_inherited_linker_flags" | $SED 's% \([^ $]*\).ltframework% -
framework \1%g'`

```



```

fi
fi
dependency_libs="$newdependency_libs"
if test "$pass" = dlpreopen; then
# Link the dlpreopened libraries before other libraries
for deplib in $save_deplibs; do
    deplibs="$deplib $deplibs"
done
fi
if test "$pass" != dlopen; then
if test "$pass" != conv; then
    # Make sure lib_search_path contains only unique directories.
    lib_search_path=
    for dir in $newlib_search_path; do
        case "$lib_search_path " in
            *" $dir "*) ;;
            *) lib_search_path+="$ dir" ;;
        esac
    done
    newlib_search_path=
fi

if test "$linkmode,$pass" != "prog,link"; then
    vars="deplibs"
else
    vars="compile_deplibs finalize_deplibs"
fi
for var in $vars dependency_libs; do
    # Add libraries to $var in reverse order
    eval tmp_libs="\${$var}"
    new_libs=
    for deplib in $tmp_libs; do
        # FIXME: Pedantically, this is the right thing to do, so
        #         that some nasty dependency loop isn't accidentally
        #         broken:
        #new_libs="$deplib $new_libs"
        # Pragmatically, this seems to cause very few problems in
        # practice:
        case $deplib in
            -L*) new_libs="$deplib $new_libs" ;;
            -R*) ;;
            *)
                # And here is the reason: when a library appears more
                # than once as an explicit dependence of a library, or
                # is implicitly linked in more than once by the
                # compiler, it is considered special, and multiple
                # occurrences thereof are not removed. Compare this
                # with having the same library being listed as a
                # dependency of multiple other libraries: in this case,
                # we know (pedantically, we assume) the library does not
                # need to be listed more than once, so we keep only the
                # last copy. This is not always right, but it is rare

```

```

# enough that we require users that really mean to play
# such unportable linking tricks to link the library
# using -Wl,-lname, so that libtool does not consider it
# for duplicate removal.
case " $specialdeplibs " in
*" $deplib "*) new_libs="$deplib $new_libs" ;;
*)
case " $new_libs " in
*" $deplib "*) ;;
*) new_libs="$deplib $new_libs" ;;
esac
;;
esac
;;
esac
done
tmp_libs=
for deplib in $new_libs; do
case $deplib in
-L*)
case " $tmp_libs " in
*" $deplib "*) ;;
*) tmp_libs+=" $deplib" ;;
esac
;;
*) tmp_libs+=" $deplib" ;;
esac
done
eval $var="\$tmp_libs\"
done # for var
fi
# Last step: remove runtime libs from dependency_libs
# (they stay in deplibs)
tmp_libs=
for i in $dependency_libs ; do
case " $predeps $postdeps $compiler_lib_search_path " in
*" $i "*)
i=""
;;
esac
if test -n "$i" ; then
tmp_libs+=" $i"
fi
done
dependency_libs=$tmp_libs
done # for pass
if test "$linkmode" = prog; then
dlfiles="$newdlfiles"
fi
if test "$linkmode" = prog || test "$linkmode" = lib; then
dlprefiles="$newdlprefiles"
fi

```

```

case $linkmode in
oldlib)
    if test -n "$dlfiles$dlpfiles" || test "$dlself" != no; then
        func_warning "\`-dlopen' is ignored for archives"
        fi

        case " $deplibs" in
        *\ -l* | *\ -L*)
            func_warning "\`-l' and \`-L' are ignored for archives" ;;
        esac

        test -n "$rpath" && \
            func_warning "\`-rpath' is ignored for archives"

        test -n "$xrpath" && \
            func_warning "\`-R' is ignored for archives"

        test -n "$vinfo" && \
            func_warning "\`-version-info/-version-number' is ignored for
archives"

        test -n "$release" && \
            func_warning "\`-release' is ignored for archives"

        test -n "$export_symbols$export_symbols_regex" && \
            func_warning "\`-export-symbols' is ignored for archives"

        # Now set the variables for building old libraries.
        build_libtool_libs=no
        oldlibs="$output"
        objs+="$old_deplibs"
        ;;

lib)
    # Make sure we only generate libraries of the form `libNAME.la'.
    case $outputname in
    lib*)
        func_stripname 'lib' '.la' "$outputname"
        name=$func_stripname_result
        eval shared_ext="\`$shrext_cmds\"
        eval libname="\`$libname_spec\"
        ;;
    *)
        test "$module" = no && \
            func_fatal_help "libtool library \`$output' must begin with
\`lib'"

        if test "$need_lib_prefix" != no; then
            # Add the "lib" prefix for modules if required
            func_stripname '' '.la' "$outputname"
            name=$func_stripname_result

```

```

    eval shared_ext="\$shrext_cmds\"
    eval libname="\$libname_spec\"
else
    func_stripname ' ' '.la' "$outputname"
    libname=$func_stripname_result
fi
;;
esac

if test -n "$objs"; then
if test "$deplibs_check_method" != pass_all; then
    func_fatal_error "cannot build libtool library \`$output' from
non-libtool objects on this host:$objs"
else
    echo
    $ECHO "*** Warning: Linking the shared library $output against
the non-libtool"
    $ECHO "*** objects $objs is not portable!"
    libobjs+=" $objs"
fi
fi

test "$dlsel" != no && \
func_warning "\`-dlopen self' is ignored for libtool libraries"

set dummy $rpath
shift
test "$#" -gt 1 && \
func_warning "ignoring multiple \`-rpath's for a libtool library"

install_libdir="$1"

oldlibs=
if test -z "$rpath"; then
if test "$build_libtool_libs" = yes; then
    # Building a libtool convenience library.
    # Some compilers have problems with a `.al' extension so
    # convenience libraries should have the same extension an
    # archive normally would.
    oldlibs="$output_objdir/$libname.$libext $oldlibs"
    build_libtool_libs=convenience
    build_old_libs=yes
fi
fi

test -n "$vinfo" && \
    func_warning "\`-version-info/-version-number' is ignored for
convenience libraries"

test -n "$release" && \
    func_warning "\`-release' is ignored for convenience libraries"
else

```

```

# Parse the version information argument.
save_ifs="$IFS"; IFS=':'
set dummy $vinfo 0 0 0
shift
IFS="$save_ifs"

test -n "$7" && \
    func_fatal_help "too many parameters to \`-version-info'"

# convert absolute version numbers to libtool ages
# this retains compatibility with .la files and attempts
# to make the code below a bit more comprehensible

case $vinfo_number in
yes)
    number_major="$1"
    number_minor="$2"
    number_revision="$3"
    #
    # There are really only two kinds -- those that
    # use the current revision as the major version
    # and those that subtract age and use age as
    # a minor version.  But, then there is irix
    # which has an extra 1 added just for fun
    #
    case $version_type in
    # correct linux to gnu/linux during the next big refactor
    darwin|linux|osf|windows|none)
        func_arith $number_major + $number_minor
        current=$func_arith_result
        age="$number_minor"
        revision="$number_revision"
        ;;
    freebsd-aout|freebsd-elf|qnx|sunos)
        current="$number_major"
        revision="$number_minor"
        age="0"
        ;;
    irix|nonstopux)
        func_arith $number_major + $number_minor
        current=$func_arith_result
        age="$number_minor"
        revision="$number_minor"
        lt_irix_increment=no
        ;;
    esac
    ;;
no)
    current="$1"
    revision="$2"
    age="$3"
    ;;

```

```

esac

# Check that each of the things are valid numbers.
case $current in
0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-9][0-9][0-9][0-9]) ;;
*)
    func_error "CURRENT \`${current}' must be a nonnegative integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

case $revision in
0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-9][0-9][0-9][0-9]) ;;
*)
    func_error "REVISION \`${revision}' must be a nonnegative
integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

case $age in
0|[1-9]|[1-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9][0-9][0-9]|[1-9][0-9][0-9][0-9][0-9]) ;;
*)
    func_error "AGE \`${age}' must be a nonnegative integer"
    func_fatal_error "\`${vinfo}' is not valid version information"
    ;;
esac

if test "$age" -gt "$current"; then
    func_error "AGE \`${age}' is greater than the current interface
number \`${current}'"
    func_fatal_error "\`${vinfo}' is not valid version information"
fi

# Calculate the version variables.
major=
versuffix=
verstring=
case $version_type in
none) ;;

darwin)
    # Like Linux, but with the current version available in
    # verstring for coding it into the library header
    func_arith $current - $age
    major=.$func_arith_result
    versuffix="$major.$age.$revision"
    # Darwin ld doesn't like 0 for these options...
    func_arith $current + 1

```

```

        minor_current=$func_arith_result
        xlcverstring="${wl}-compatibility_version ${wl}$minor_current
${wl}-current_version ${wl}$minor_current.$revision"
        verstring="-compatibility_version $minor_current -
current_version $minor_current.$revision"
        ;;

freebsd-aout)
    major=".$current"
    versuffix=".$current.$revision";
    ;;

freebsd-elf)
    major=".$current"
    versuffix=".$current"
    ;;

irix | nonstopux)
    if test "X$lt_irix_increment" = "Xno"; then
        func_arith $current - $age
    else
        func_arith $current - $age + 1
    fi
    major=$func_arith_result

    case $version_type in
        nonstopux) verstring_prefix=nonstopux ;;
        *)          verstring_prefix=sgi ;;
    esac
    verstring="$verstring_prefix$major.$revision"

    # Add in all the interfaces that we are compatible with.
    loop=$revision
    while test "$loop" -ne 0; do
        func_arith $revision - $loop
        iface=$func_arith_result
        func_arith $loop - 1
        loop=$func_arith_result
        verstring="$verstring_prefix$major.$iface:$verstring"
    done

    # Before this point, $major must not contain `.'.
    major=.$major
    versuffix="$major.$revision"
    ;;

linux) # correct to gnu/linux during the next big refactor
    func_arith $current - $age
    major=.$func_arith_result
    versuffix="$major.$age.$revision"
    ;;

```

```

osf)
  func_arith $current - $age
  major=.$func_arith_result
  versuffix=".$current.$age.$revision"
  verstring="$current.$age.$revision"

  # Add in all the interfaces that we are compatible with.
  loop=$age
  while test "$loop" -ne 0; do
    func_arith $current - $loop
    iface=$func_arith_result
    func_arith $loop - 1
    loop=$func_arith_result
    verstring="$verstring:${iface}.0"
  done

  # Make executables depend on our current version.
  verstring+=":${current}.0"
  ;;

qnx)
  major=".$current"
  versuffix=".$current"
  ;;

sunos)
  major=".$current"
  versuffix=".$current.$revision"
  ;;

windows)
  # Use '-' rather than '.', since we only want one
  # extension on DOS 8.3 filesystems.
  func_arith $current - $age
  major=$func_arith_result
  versuffix="-${major}"
  ;;

*)
  func_fatal_configuration "unknown library version type
\`$version_type'"
  ;;
esac

# Clear the version info if we defaulted, and they specified a
release.
if test -z "$vinfo" && test -n "$release"; then
  major=
  case $version_type in
  darwin)
    # we can't check for "0.0" in archive_cmds due to quoting
    # problems, so we reset it completely

```



```

        verstring=
        ;;
    *)
        verstring="0.0"
        ;;
    esac
    if test "$need_version" = no; then
        versuffix=
    else
        versuffix=".0.0"
    fi
fi

# Remove version info from name if versioning should be avoided
if test "$avoid_version" = yes && test "$need_version" = no; then
    major=
    versuffix=
    verstring=""
fi

# Check to see if the archive will have undefined symbols.
if test "$allow_undefined" = yes; then
    if test "$allow_undefined_flag" = unsupported; then
        func_warning "undefined symbols not allowed in $host shared
libraries"
        build_libtool_libs=no
        build_old_libs=yes
    fi
else
    # Don't allow undefined symbols.
    allow_undefined_flag="$no_undefined_flag"
fi

fi

func_generate_dlsyms "$libname" "$libname" "yes"
libobjs+=" $symfileobj"
test "X$libobjs" = "X " && libobjs=

if test "$opt_mode" != relink; then
# Remove our outputs, but don't remove object files since they
# may have been created when compiling PIC objects.
removelist=
tempremovelist=`$ECHO "$output_objdir/*"`
for p in $tempremovelist; do
    case $p in
        *.$objext | *.gcno)
            ;;
        $output_objdir/$outputname | $output_objdir/$libname.* |
        $output_objdir/${libname}${release}.* )
            if test "X$precious_files_regex" != "X"; then

```

```

        if $ECHO "$p" | $EGREP -e "$precious_files_regex"
>/dev/null 2>&1
        then
            continue
        fi
        fi
        removelist+=" $p"
        ;;
    *) ;;
esac
done
test -n "$removelist" && \
func_show_eval "${RM}r \${removelist}"
fi

# Now set the variables for building old libraries.
if test "$build_old_libs" = yes && test "$build_libtool_libs" !=
convenience ; then
    oldlibs+=" $output_objdir/$libname.$libext"

    # Transform .lo files to .o files.
    oldobjs="$objs "`$ECHO "$libobjs" | $SP2NL | $SED
"/\.${libext}$/d; $lo2o" | $NL2SP`
    fi

    # Eliminate all temporary directories.
    #for path in $notinst_path; do
    #    lib_search_path=`$ECHO "$lib_search_path " | $SED "s% $path
% %g"`
    #    deplibs=`$ECHO "$deplibs " | $SED "s% -L$path % %g"`
    #    dependency_libs=`$ECHO "$dependency_libs " | $SED "s% -
L$path % %g"`
    #done

    if test -n "$xrpath"; then
    # If the user specified any rpath flags, then add them.
    temp_xrpath=
    for libdir in $xrpath; do
        func_replace_sysroot "$libdir"
        temp_xrpath+="-R${func_replace_sysroot_result}"
        case "$finalize_rpath " in
        *" $libdir ") ;;
        *) finalize_rpath+="$libdir" ;;
        esac
    done
    if test "$hardcode_into_libs" != yes || test "$build_old_libs" =
yes; then
        dependency_libs="$temp_xrpath $dependency_libs"
    fi
    fi

```

```

# Make sure dlfiles contains only unique files that won't be
dlpreopened
old_dlfiles="$dlfiles"
dlfiles=
for lib in $old_dlfiles; do
case " $dlprefiles $dlfiles " in
*" $lib ") ;;
*) dlfiles+="$ lib" ;;
esac
done

# Make sure dlprefiles contains only unique files
old_dlprefiles="$dlprefiles"
dlprefiles=
for lib in $old_dlprefiles; do
case "$dlprefiles " in
*" $lib ") ;;
*) dlprefiles+="$ lib" ;;
esac
done

if test "$build_libtool_libs" = yes; then
if test -n "$rpath"; then
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-*-beos* |
*-cegcc* | *-*-haiku*)
# these systems don't actually have a c library (as such)!
;;
*-*-rhapsody* | *-*-darwin1.[012])
# Rhapsody C library is in the System framework
deplibs+=" System.ltframework"
;;
*-*-netbsd*)
# Don't link with libc until the a.out ld.so is fixed.
;;
*-*-openbsd* | *-*-freebsd* | *-*-dragonfly*)
# Do not include libc due to us having libc/libc_r.
;;
*-*-sco3.2v5* | *-*-sco5v6*)
# Causes problems with __ctype
;;
*-*-sysv4.2uw2* | *-*-sysv5* | *-*-unixware* | *-*-OpenUNIX*)
# Compiler inserts libc in the correct place for threads to
work
;;
*)
# Add libc to deplibs on all other systems if necessary.
if test "$build_libtool_need_lc" = "yes"; then
deplibs+=" -lc"
fi
;;
esac

```

```

fi

# Transform deplibs into only deplibs that can be linked in
shared.
name_save=$name
libname_save=$libname
release_save=$release
versuffix_save=$versuffix
major_save=$major
# I'm not sure if I'm treating the release correctly. I think
# release should show up in the -l (ie -lgmp5) so we don't want
to
# add it in twice. Is that correct?
release=""
versuffix=""
major=""
newdeplibs=
droppeddeps=no
case $deplibs_check_method in
pass_all)
# Don't check for shared/static. Everything works.
# This might be a little naive. We might want to check
# whether the library exists or not. But this is on
# osf3 & osf4 and I'm not really sure... Just
# implementing what was already the behavior.
newdeplibs=$deplibs
;;
test_compile)
# This code stresses the "libraries are programs" paradigm to
its
# limits. Maybe even breaks it. We compile a program, linking
it
# against the deplibs as a proxy for the library. Then we can
check
# whether they linked in statically or dynamically with ldd.
$opt_dry_run || $RM conftest.c
cat > conftest.c <<EOF
int main() { return 0; }
EOF
$opt_dry_run || $RM conftest
if $LTCC $LTCCFLAGS -o conftest conftest.c $deplibs; then
ldd_output=`ldd conftest`
for i in $deplibs; do
case $i in
-l*)
func_stripname -l '' "$i"
name=$func_stripname_result
if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
case " $predeps $postdeps " in
*" $i "*)
newdeplibs+="$ i"

```

```

        i=""
        ;;
    esac
fi
if test -n "$i" ; then
    libname=`eval "\\$ECHO \"$libname_spec\""`
    deplib_matches=`eval "\\$ECHO \"$library_names_spec\""`
    set dummy $deplib_matches; shift
    deplib_match=$1
    if test `expr "$ldd_output" : ".*$deplib_match"` -ne 0 ;
then
        newdeplibs+="$i"
    else
        droppeddeps=yes
        echo
        $ECHO "*** Warning: dynamic linker does not accept
needed library $i."
        echo "*** I have the capability to make that library
automatically link in when"
        echo "*** you link to this library.  But I can only do
this if you have a"
        echo "*** shared version of the library, which I
believe you do not have"
        echo "*** because a test_compile did reveal that the
linker did not use it for"
        echo "*** its dynamic dependency list that programs get
resolved with at runtime."
    fi
fi
    *)
    newdeplibs+="$i"
    ;;
esac
done
else
# Error occurred in the first compile.  Let's try to salvage
# the situation: Compile a separate program for each library.
for i in $deplibs; do
    case $i in
    -l*)
        func_stripname -l '' "$i"
        name=$func_stripname_result
        $opt_dry_run || $RM conftest
        if $LTCC $LTCCFLAGS -o conftest conftest.c $i; then
            ldd_output=`ldd conftest`
            if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
                case " $predeps $postdeps " in
                *" $i "*)
                    newdeplibs+="$i"
                    i=""

```

```

        ;;
    esac
fi
if test -n "$i" ; then
    libname=`eval "\\$ECHO \"\$libname_spec\""`
    deplib_matches=`eval "\\$ECHO \"\$library_names_spec\""`
    set dummy $deplib_matches; shift
    deplib_match=$1
    if test `expr "$ldd_output" : ".*$deplib_match" -ne 0`
; then
        newdeplibs+="$i"
    else
        droppeddeps=yes
        echo
        $ECHO "*** Warning: dynamic linker does not accept
needed library $i."
        echo "*** I have the capability to make that library
automatically link in when"
        echo "*** you link to this library. But I can only
do this if you have a"
        echo "*** shared version of the library, which you do
not appear to have"
        echo "*** because a test_compile did reveal that the
linker did not use this one"
        echo "*** as a dynamic dependency that programs can
get resolved with at runtime."
        fi
    fi
else
    droppeddeps=yes
    echo
    $ECHO "*** Warning! Library $i is needed by this library
but I was not able to"
    echo "*** make it link in! You will probably need to
install it or some"
    echo "*** library that it depends on before this library
will be fully"
    echo "*** functional. Installing it before continuing
would be even better."
    fi
    ;;
*)
    newdeplibs+="$i"
    ;;
esac
done
fi
;;
file_magic*)
    set dummy $deplibs_check_method; shift
    file_magic_regex=`expr "$deplibs_check_method" : "$1 \(.*\)"`
    for a_deplib in $deplibs; do

```

```

case $a_deplib in
-1*)
    func_stripname -1 '' "$a_deplib"
    name=$func_stripname_result
    if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then
        case " $predeps $postdeps " in
        *" $a_deplib ")
            newdeplibs+=" $a_deplib"
            a_deplib=""
            ;;
        esac
        fi
        if test -n "$a_deplib" ; then
            libname=`eval "\\$ECHO \"$libname_spec\""`
            if test -n "$file_magic_glob"; then
                libnameglob=`func_echo_all "$libname" | $SED -e
$file_magic_glob`
            else
                libnameglob=$libname
            fi
            test "$want_nocaseglob" = yes && nocaseglob=`shopt -p
nocaseglob`
            for i in $lib_search_path $sys_lib_search_path
$shlib_search_path; do
                if test "$want_nocaseglob" = yes; then
                    shopt -s nocaseglob
                    potential_libs=`ls $i/$libnameglob[.]* 2>/dev/null`
                    $nocaseglob
                else
                    potential_libs=`ls $i/$libnameglob[.]* 2>/dev/null`
                fi
                for potent_lib in $potential_libs; do
                    # Follow soft links.
                    if ls -lLd "$potent_lib" 2>/dev/null |
                    $GREP " -> " >/dev/null; then
                        continue
                    fi
                    # The statement above tries to avoid entering an
                    # endless loop below, in case of cyclic links.
                    # We might still enter an endless loop, since a link
                    # loop can be closed while we follow links,
                    # but so what?
                    potlib="$potent_lib"
                    while test -h "$potlib" 2>/dev/null; do
                        potliblink=`ls -ld $potlib | ${SED} 's/.* -> //'`
                        case $potliblink in
                        [\\/*]* | [A-Za-z]:[\\/*]*) potlib="$potliblink";;
                        *) potlib=`$ECHO "$potlib" | $SED
's,[^/]*$,,'`"$potliblink";;
                        esac
                    done
                done
            done

```

```

        if eval $file_magic_cmd "\${spotlib}" 2>/dev/null |
        $SED -e 10q |
        $EGREP "$file_magic_regex" > /dev/null; then
newdeplibs+=" $a_deplib"
a_deplib=""
break 2
fi
    done
done
fi
if test -n "$a_deplib" ; then
droppeddeps=yes
echo
$ECHO "*** Warning: linker path does not have real file for
library $a_deplib."
echo "*** I have the capability to make that library
automatically link in when"
echo "*** you link to this library. But I can only do this
if you have a"
echo "*** shared version of the library, which you do not
appear to have"
echo "*** because I did check the linker path looking for a
file starting"
if test -z "$spotlib" ; then
    $ECHO "*** with $libname but no candidates were found.
(...for file magic test)"
else
    $ECHO "*** with $libname and none of the candidates
passed a file format test"
    $ECHO "*** using a file magic. Last file checked:
$spotlib"
fi
fi
;;
*)
# Add a -L argument.
newdeplibs+=" $a_deplib"
;;
esac
done # Gone through all deplibs.
;;
match_pattern*)
set dummy $deplibs_check_method; shift
match_pattern_regex=`expr "$deplibs_check_method" : "$1
\(.*\)"`
for a_deplib in $deplibs; do
case $a_deplib in
-1*)
func_stripname -1 ' ' "$a_deplib"
name=$func_stripname_result
if test "X$allow_libtool_libs_with_static_runtimes" =
"Xyes" ; then

```



```

case " $predeps $postdeps " in
*" $a_deplib *)
    newdeplibs+=" $a_deplib"
    a_deplib=""
    ;;
esac
fi
if test -n "$a_deplib" ; then
libname=`eval "\\$ECHO \"$libname_spec\""`
for i in $lib_search_path $sys_lib_search_path
$shlib_search_path; do
    potential_libs=`ls $i/$libname[.]* 2>/dev/null`
    for potent_lib in $potential_libs; do
        potlib="$potent_lib" # see symlink-check above in
file_magic test
        if eval "\\$ECHO \"$potent_lib\" 2>/dev/null | $SED 10q
| \
            $EGREP "$match_pattern_regex" > /dev/null; then
            newdeplibs+=" $a_deplib"
            a_deplib=""
            break 2
        fi
    done
done
fi
if test -n "$a_deplib" ; then
droppeddeps=yes
echo
$ECHO "*** Warning: linker path does not have real file for
library $a_deplib."
echo "*** I have the capability to make that library
automatically link in when"
echo "*** you link to this library. But I can only do this
if you have a"
echo "*** shared version of the library, which you do not
appear to have"
echo "*** because I did check the linker path looking for a
file starting"
if test -z "$potlib" ; then
    $ECHO "*** with $libname but no candidates were found.
(...for regex pattern test)"
else
    $ECHO "*** with $libname and none of the candidates
passed a file format test"
    $ECHO "*** using a regex pattern. Last file checked:
$potlib"
fi
fi
;;
*)
    # Add a -L argument.
    newdeplibs+=" $a_deplib"

```

```

        ;;
    esac
done # Gone through all deplibs.
;;
none | unknown | *)
newdeplibs=""
tmp_deplibs=`$ECHO " $deplibs" | $SED 's/ -lc$//; s/ -[LR][^
]*/g'`
if test "X$allow_libtool_libs_with_static_runtimes" = "Xyes" ;
then
    for i in $predeps $postdeps ; do
        # can't use Xsed below, because $i might contain '/'
        tmp_deplibs=`$ECHO " $tmp_deplibs" | $SED "s,$i,,"`
    done
fi
case $tmp_deplibs in
*(!\      \ ]*)
    echo
    if test "X$deplibs_check_method" = "Xnone"; then
        echo "*** Warning: inter-library dependencies are not
supported in this platform."
    else
        echo "*** Warning: inter-library dependencies are not known
to be supported."
    fi
    echo "*** All declared inter-library dependencies are being
dropped."
    droppeddeps=yes
    ;;
esac
;;
esac
;;
esac
versuffix=$versuffix_save
major=$major_save
release=$release_save
libname=$libname_save
name=$name_save

case $host in
*-*-rhapsody* | *-*-darwin1.[012])
    # On Rhapsody replace the C library with the System framework
    newdeplibs=`$ECHO " $newdeplibs" | $SED 's/ -lc /
System.ltframework /'`
    ;;
esac

if test "$droppeddeps" = yes; then
    if test "$module" = yes; then
        echo
        echo "*** Warning: libtool could not satisfy all declared
inter-library"

```

```

        $ECHO "**** dependencies of module $libname.  Therefore,
libtool will create"
        echo "**** a static module, that should work as long as the
dlopening"
        echo "**** application is linked with the -dlopen flag."
        if test -z "$global_symbol_pipe"; then
            echo
            echo "**** However, this would only work if libtool was able
to extract symbol"
            echo "**** lists from a program, using `nm' or equivalent,
but libtool could"
            echo "**** not find such a program.  So, this module is
probably useless."
            echo "**** `nm' from GNU binutils and a full rebuild may
help."
        fi
        if test "$build_old_libs" = no; then
            oldlibs="$output_objdir/$libname.$libext"
            build_libtool_libs=module
            build_old_libs=yes
        else
            build_libtool_libs=no
        fi
    else
        echo "**** The inter-library dependencies that have been
dropped here will be"
        echo "**** automatically added whenever a program is linked
with this library"
        echo "**** or is declared to -dlopen it."

        if test "$allow_undefined" = no; then
            echo
            echo "**** Since this library must not contain undefined
symbols,"
            echo "**** because either the platform does not support them
or"
            echo "**** it was explicitly requested with -no-undefined,"
            echo "**** libtool will only create a static version of it."
            if test "$build_old_libs" = no; then
                oldlibs="$output_objdir/$libname.$libext"
                build_libtool_libs=module
                build_old_libs=yes
            else
                build_libtool_libs=no
            fi
        fi
    fi
fi
fi
# Done checking deplibs!
deplibs=$newdeplibs
fi

```

```

# Time to change all our "foo.ltframework" stuff back to "-
framework foo"
case $host in
*-*-darwin*)
    newdeplibs=`$ECHO " $newdeplibs" | $SED 's% \([^
$]*\).ltframework% -framework \1%g'`
    new_inherited_linker_flags=`$ECHO "
$new_inherited_linker_flags" | $SED 's% \([^ $]*\).ltframework% -
framework \1%g'`
    deplibs=`$ECHO " $deplibs" | $SED 's% \([^ $]*\).ltframework% -
framework \1%g'`
    ;;
esac

# move library search paths that coincide with paths to not yet
# installed libraries to the beginning of the library search
list
new_libs=
for path in $notinst_path; do
case " $new_libs " in
*" -L$path/$objdir "*) ;;
*)
    case " $deplibs " in
    *" -L$path/$objdir "*)
        new_libs+="-L$path/$objdir" ;;
    esac
    ;;
esac
done
for deplib in $deplibs; do
case $deplib in
-L*)
    case " $new_libs " in
    "$deplib "*) ;;
    *) new_libs+="$deplib" ;;
    esac
    ;;
*) new_libs+="$deplib" ;;
esac
done
deplibs="$new_libs"

# All the library-specific variables (install_libdir is set
above).
library_names=
old_library=
dlname=

# Test again, we may have decided not to build it any more
if test "$build_libtool_libs" = yes; then
# Remove ${wl} instances when linking with ld.
# FIXME: should test the right_cmds variable.

```

```

case $archive_cmds in
  *\$LD\ *) wl= ;;
  esac
if test "$hardcode_into_libs" = yes; then
  # Hardcode the library paths
  hardcode_libdirs=
  dep_rpath=
  rpath="$finalize_rpath"
  test "$opt_mode" != relink && rpath="$compile_rpath$rpath"
  for libdir in $rpath; do
    if test -n "$hardcode_libdir_flag_spec"; then
      func_replace_sysroot "$libdir"
      libdir=$func_replace_sysroot_result
      func_stripname '=' '' "$libdir"
      libdir=$func_stripname_result
      if test -n "$hardcode_libdir_separator"; then
        if test -z "$hardcode_libdirs"; then
          hardcode_libdirs="$libdir"
        else
          # Just accumulate the unique libdirs.
          case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
          *"$hardcode_libdir_separator$libdir$hardcode_libdir_separator"*)
            ;;
          *)
            hardcode_libdirs+="$hardcode_libdir_separator$libdir"
            ;;
          esac
        fi
      fi
    else
      # We only want to hardcode in an rpath if it isn't in
the
      # default dlsearch path.
      func_normal_abspath "$libdir"
      libdir_norm=$func_normal_abspath_result
      case "$sys_lib_dlsearch_path" in
        * $libdir_norm *) ;;
        *) eval flag=\"$hardcode_libdir_flag_spec\"
          dep_rpath+="$flag"
          ;;
      esac
    fi
  done
  elif test -n "$runpath_var"; then
    case "$perm_rpath" in
      * $libdir *) ;;
      *) perm_rpath+="$libdir" ;;
    esac
  fi
done
# Substitute the hardcoded libdirs into the rpath.

```

```

    if test -n "$hardcode_libdir_separator" &&
        test -n "$hardcode_libdirs"; then
        libdir="$hardcode_libdirs"
        eval "dep_rpath=\`"$hardcode_libdir_flag_spec`\`"
    fi
    if test -n "$runpath_var" && test -n "$perm_rpath"; then
        # We should set the runpath_var.
        rpath=
        for dir in $perm_rpath; do
            rpath+="$dir:"
        done
        eval "$runpath_var='`$rpath`'$runpath_var'; export
$runpath_var"
    fi
    test -n "$dep_rpath" && deplibs="$dep_rpath $deplibs"
fi

shlibpath="$finalize_shlibpath"
test "$opt_mode" != relink &&
shlibpath="$compile_shlibpath$shlibpath"
if test -n "$shlibpath"; then
    eval "$shlibpath_var='`$shlibpath`'$shlibpath_var'; export
$shlibpath_var"
fi

# Get the real and link names of the library.
eval shared_ext=``$shrext_cmds``
eval library_names=``$library_names_spec``
set dummy $library_names
shift
realname="$1"
shift

if test -n "$soname_spec"; then
    eval soname=``$soname_spec``
else
    soname="$realname"
fi
if test -z "$dlname"; then
    dlname=$soname
fi

lib="$output_objdir/$realname"
linknames=
for link
do
    linknames+="$link"
done

# Use standard objects if they are pic
test -z "$pic_flag" && libobjs=`$ECHO "$libobjs" | $SP2NL | $SED
"$1o2o" | $NL2SP`

```

```

test "X$libobjs" = "X " && libobjs=

delfiles=
if test -n "$export_symbols" && test -n "$include_expsyms"; then
  $opt_dry_run || cp "$export_symbols"
"$output_objdir/$libname.uexp"
  export_symbols="$output_objdir/$libname.uexp"
  delfiles+=" $export_symbols"
fi

orig_export_symbols=
case $host_os in
cygwin* | mingw* | cegcc*)
  if test -n "$export_symbols" && test -z
"$export_symbols_regex"; then
    # exporting using user supplied symfile
    if test "x`$SED 1q $export_symbols`" != xEXPORTS; then
      # and it's NOT already a .def file. Must figure out
      # which of the given symbols are data symbols and tag
      # them as such. So, trigger use of export_symbols_cmds.
      # export_symbols gets reassigned inside the "prepare
      # the list of exported symbols" if statement, so the
      # include_expsyms logic still works.
      orig_export_symbols="$export_symbols"
      export_symbols=
      always_export_symbols=yes
    fi
  fi
;;
esac

# Prepare the list of exported symbols
if test -z "$export_symbols"; then
  if test "$always_export_symbols" = yes || test -n
"$export_symbols_regex"; then
    func_verbose "generating symbol list for \`$libname.la'"
    export_symbols="$output_objdir/$libname.exp"
    $opt_dry_run || $RM $export_symbols
    cmds=$export_symbols_cmds
    save_ifs="$IFS"; IFS='~'
    for cmd1 in $cmds; do
      IFS="$save_ifs"
      # Take the normal branch if the nm_file_list_spec branch
      # doesn't work or if tool conversion is not needed.
      case $nm_file_list_spec~$to_tool_file_cmd in
*~func_convert_file_noop | *~func_convert_file_msys_to_w32
| ~*)
        try_normal_branch=yes
        eval cmd="\`$cmd1\`"
        func_len " $cmd"
        len=$func_len_result
        ;;

```

```

*)
    try_normal_branch=no
    ;;
esac
if test "$try_normal_branch" = yes \
  && { test "$len" -lt "$max_cmd_len" \
    || test "$max_cmd_len" -le -1; }
then
func_show_eval "$cmd" 'exit $?'
skipped_export=false
elif test -n "$nm_file_list_spec"; then
func_basename "$output"
output_la=$func_basename_result
save_libobjs=$libobjs
save_output=$output
output=${output_objdir}/${output_la}.nm
func_to_tool_file "$output"
libobjs=$nm_file_list_spec$func_to_tool_file_result
delfiles+=" $output"
func_verbose "creating $NM input file list: $output"
for obj in $save_libobjs; do
    func_to_tool_file "$obj"
    $ECHO "$func_to_tool_file_result"
done > "$output"
eval cmd="\$cmd1\"
func_show_eval "$cmd" 'exit $?'
output=$save_output
libobjs=$save_libobjs
skipped_export=false
else
# The command line is too long to execute in one step.
func_verbose "using reloadable object file for export
list..."
skipped_export=:
# Break out early, otherwise skipped_export may be
# set to false by a later but shorter cmd.
break
fi
done
IFS=$save_ifs"
if test -n "$export_symbols_regex" && test "X$skipped_export"
!= "X:"; then
    func_show_eval '$EGREP -e "$export_symbols_regex"
"$export_symbols" > "${export_symbols}T"'
    func_show_eval '$MV "${export_symbols}T" "$export_symbols"'
fi
fi
fi

if test -n "$export_symbols" && test -n "$include_expsyms"; then
    tmp_export_symbols="$export_symbols"

```



```

        test -n "$orig_export_symbols" &&
tmp_export_symbols="$orig_export_symbols"
        $opt_dry_run || eval '$ECHO "$include_expsyms" | $SP2NL >>
"$tmp_export_symbols"'
        fi

        if test "X$skipped_export" != "X:" && test -n
"$orig_export_symbols"; then
        # The given exports_symbols file has to be filtered, so filter
it.
        func_verbose "filter symbol list for \`$libname.la' to tag DATA
exports"
        # FIXME: $output_objdir/$libname.filter potentially contains
lots of
        # 's' commands which not all seds can handle. GNU sed should be
fine
        # though. Also, the filter scales superlinearly with the number
of
        # global variables. join(1) would be nice here, but
unfortunately
        # isn't a blessed tool.
        $opt_dry_run || $SED -e '/[ ,]DATA/!d;s,\(.*\)\[
\, \].*\),s|^\1$|\1\2|,' < $export_symbols >
$output_objdir/$libname.filter
        delfiles+=" $export_symbols $output_objdir/$libname.filter"
        export_symbols=$output_objdir/$libname.def
        $opt_dry_run || $SED -f $output_objdir/$libname.filter <
$orig_export_symbols > $export_symbols
        fi

tmp_deplibs=
for test_deplib in $deplibs; do
    case " $convenience " in
        *" $test_deplib ") ;;
        *)
            tmp_deplibs+=" $test_deplib"
            ;;
    esac
done
deplibs="$tmp_deplibs"

if test -n "$convenience"; then
    if test -n "$whole_archive_flag_spec" &&
        test "$compiler_needs_object" = yes &&
        test -z "$libobjs"; then
        # extract the archives, so we have objects to list.
        # TODO: could optimize this to just extract one archive.
        whole_archive_flag_spec=
    fi
    if test -n "$whole_archive_flag_spec"; then
        save_libobjs=$libobjs
        eval libobjs="\`$libobjs $whole_archive_flag_spec\`"
    fi
fi

```

```

    test "X$libobjs" = "X " && libobjs=
else
    gentop="$output_objdir/${outputname}x"
    generated+=" $gentop"

    func_extract_archives $gentop $convenience
    libobjs+=" $func_extract_archives_result"
    test "X$libobjs" = "X " && libobjs=
fi
fi

if test "$thread_safe" = yes && test -n "$thread_safe_flag_spec";
then
    eval flag=\ "$thread_safe_flag_spec"
    linker_flags+=" $flag"
fi

# Make a backup of the uninstalled library when relinking
if test "$opt_mode" = relink; then
    $opt_dry_run || eval '(cd $output_objdir && $RM ${realname}U &&
$MV $realname ${realname}U)' || exit $?
fi

# Do each of the archive commands.
if test "$module" = yes && test -n "$module_cmds" ; then
    if test -n "$export_symbols" && test -n "$module_expsym_cmds";
then
        eval test_cmds=\ "$module_expsym_cmds"
        cmds=$module_expsym_cmds
    else
        eval test_cmds=\ "$module_cmds"
        cmds=$module_cmds
    fi
else
    if test -n "$export_symbols" && test -n "$archive_expsym_cmds";
then
        eval test_cmds=\ "$archive_expsym_cmds"
        cmds=$archive_expsym_cmds
    else
        eval test_cmds=\ "$archive_cmds"
        cmds=$archive_cmds
    fi
fi

if test "X$skipped_export" != "X:" &&
    func_len " $test_cmds" &&
    len=$func_len_result &&
    test "$len" -lt "$max_cmd_len" || test "$max_cmd_len" -le -1;
then
    :
else

```

```

# The command line is too long to link in one step, link
piecewise
# or, if using GNU ld and skipped_export is not :, use a linker
# script.

# Save the value of $output and $libobjs because we want to
# use them later.  If we have whole_archive_flag_spec, we
# want to use save_libobjs as it was before
# whole_archive_flag_spec was expanded, because we can't
# assume the linker understands whole_archive_flag_spec.
# This may have to be revisited, in case too many
# convenience libraries get linked in and end up exceeding
# the spec.
if test -z "$convenience" || test -z
"$whole_archive_flag_spec"; then
    save_libobjs=$libobjs
fi
save_output=$output
func_basename "$output"
output_la=$func_basename_result

# Clear the reloadable object creation command queue and
# initialize k to one.
test_cmds=
concat_cmds=
objlist=
last_robj=
k=1

if test -n "$save_libobjs" && test "X$skipped_export" != "X:"
&& test "$with_gnu_ld" = yes; then
    output=${output_objdir}/${output_la}.lnkscript
    func_verbose "creating GNU ld script: $output"
    echo 'INPUT (' > $output
    for obj in $save_libobjs
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result" >> $output
    done
    echo ')' >> $output
    delfiles+=" $output"
    func_to_tool_file "$output"
    output=$func_to_tool_file_result
elif test -n "$save_libobjs" && test "X$skipped_export" != "X:"
&& test "X$file_list_spec" != X; then
    output=${output_objdir}/${output_la}.lnk
    func_verbose "creating linker input file list: $output"
    : > $output
    set x $save_libobjs
    shift
    firstobj=
    if test "$compiler_needs_object" = yes; then

```

```

        firstobj="$1 "
        shift
    fi
    for obj
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result" >> $output
    done
    delfiles+="$ $output"
    func_to_tool_file "$output"
    output=${firstobj}\"$file_list_spec$func_to_tool_file_result\"
else
    if test -n "$save_libobjs"; then
        func_verbose "creating reloadable object files..."
        output=$output_objdir/$output_la-${k}.$objext
        eval test_cmds=\"$reload_cmds\"
        func_len " $test_cmds"
        len0=$func_len_result
        len=$len0

        # Loop over the list of objects to be linked.
        for obj in $save_libobjs
        do
            func_len " $obj"
            func_arith $len + $func_len_result
            len=$func_arith_result
            if test "X$objlist" = X ||
                test "$len" -lt "$max_cmd_len"; then
                objlist+="$ obj"
            else
                # The command $test_cmds is almost too long, add a
                # command to the queue.
                if test "$k" -eq 1 ; then
                    # The first file doesn't have a previous command to
add.

                    reload_objs=$objlist
                    eval concat_cmds=\"$reload_cmds\"
                else
                    # All subsequent reloadable object files will link in
                    # the last one created.
                    reload_objs="$objlist $last_robj"
                    eval concat_cmds=\"\$concat_cmds~$reload_cmds~\$RM
$last_robj\"
                fi
            fi
            last_robj=$output_objdir/$output_la-${k}.$objext
            func_arith $k + 1
            k=$func_arith_result
            output=$output_objdir/$output_la-${k}.$objext
            objlist="$ obj"
            func_len " $last_robj"
            func_arith $len0 + $func_len_result
            len=$func_arith_result

```

```

fi
done
# Handle the remaining objects by creating one last
# reloadable object file. All subsequent reloadable object
# files will link in the last one created.
test -z "$concat_cmds" || concat_cmds=$concat_cmds~
reload_objs="$objlist $last_robj"
eval concat_cmds="\${concat_cmds}$reload_cmds\"
if test -n "$last_robj"; then
    eval concat_cmds="\${concat_cmds}~\${RM} $last_robj\"
fi
delfiles+=" $output"

else
    output=
fi

if ${skipped_export-false}; then
    func_verbose "generating symbol list for \`${libname}.la'"
    export_symbols="$output_objdir/${libname}.exp"
    $opt_dry_run || $RM $export_symbols
    libobjs=$output
    # Append the command to create the export file.
    test -z "$concat_cmds" || concat_cmds=$concat_cmds~
    eval concat_cmds="\${concat_cmds}$export_symbols_cmds\"
    if test -n "$last_robj"; then
        eval concat_cmds="\${concat_cmds}~\${RM} $last_robj\"
    fi
fi

test -n "$save_libobjs" &&
    func_verbose "creating a temporary reloadable object file:
$output"

# Loop through the commands generated above and execute them.
save_ifs="$IFS"; IFS='~'
for cmd in $concat_cmds; do
    IFS="$save_ifs"
    $opt_silent || {
        func_quote_for_expand "$cmd"
        eval "func_echo $func_quote_for_expand_result"
    }
    $opt_dry_run || eval "$cmd" || {
        lt_exit=$?
    }

# Restore the uninstalled library and exit
if test "$opt_mode" = relink; then
    ( cd "$output_objdir" && \
      $RM "${realname}T" && \
      $MV "${realname}U" "$realname" )
fi

```

```

        exit $lt_exit
    }
done
IFS="$save_ifs"

    if test -n "$export_symbols_regex" && ${skipped_export-
false}; then
        func_show_eval '$EGREP -e "$export_symbols_regex"
"$export_symbols" > "${export_symbols}T"'
        func_show_eval '$MV "${export_symbols}T" "$export_symbols"'
    fi
fi

    if ${skipped_export-false}; then
    if test -n "$export_symbols" && test -n "$include_expsyms";
then
        tmp_export_symbols="$export_symbols"
        test -n "$orig_export_symbols" &&
tmp_export_symbols="$orig_export_symbols"
        $opt_dry_run || eval '$ECHO "$include_expsyms" | $SP2NL >>
"$tmp_export_symbols"'
        fi

        if test -n "$orig_export_symbols"; then
            # The given exports_symbols file has to be filtered, so
filter it.
            func_verbose "filter symbol list for `"$libname.la" to tag
DATA exports"
            # FIXME: $output_objdir/$libname.filter potentially
contains lots of
            # 's' commands which not all sed's can handle. GNU sed
should be fine
            # though. Also, the filter scales superlinearly with the
number of
            # global variables. join(1) would be nice here, but
unfortunately
            # isn't a blessed tool.
            $opt_dry_run || $SED -e '/[ ,]DATA/!d;s,\(.*\) \([
\,].*\),s|^1$|\1\2|,' < $export_symbols >
$output_objdir/$libname.filter
            delfiles+=" $export_symbols $output_objdir/$libname.filter"
            export_symbols=$output_objdir/$libname.def
            $opt_dry_run || $SED -f $output_objdir/$libname.filter <
$orig_export_symbols > $export_symbols
        fi
    fi

    libobjs=$output
    # Restore the value of output.
    output=$save_output

```

```

    if test -n "$convenience" && test -n
"$whole_archive_flag_spec"; then
        eval libobjs="\${libobjs} $whole_archive_flag_spec\"
        test "X$libobjs" = "X " && libobjs=
    fi
    # Expand the library linking commands again to reset the
    # value of $libobjs for piecewise linking.

    # Do each of the archive commands.
    if test "$module" = yes && test -n "$module_cmds" ; then
        if test -n "$export_symbols" && test -n
"$module_expsym_cmds"; then
            cmds=$module_expsym_cmds
        else
            cmds=$module_cmds
        fi
    else
        if test -n "$export_symbols" && test -n
"$archive_expsym_cmds"; then
            cmds=$archive_expsym_cmds
        else
            cmds=$archive_cmds
        fi
    fi
fi

if test -n "$delfiles"; then
    # Append the command to remove temporary files to $cmds.
    eval cmds="\${cmds}~\${RM} $delfiles\"
fi

# Add any objects from preloaded convenience libraries
if test -n "$dlprefiles"; then
    gentop="$output_objdir/${outputname}x"
    generated+=" $gentop"

    func_extract_archives $gentop $dlprefiles
    libobjs+=" ${func_extract_archives_result}"
    test "X$libobjs" = "X " && libobjs=
fi

save_ifs="$IFS"; IFS='~'
for cmd in $cmds; do
    IFS="$save_ifs"
    eval cmd="\${cmd}"
    $opt_silent || {
        func_quote_for_expand "$cmd"
        eval "func_echo ${func_quote_for_expand_result}"
    }
    $opt_dry_run || eval "$cmd" || {
        lt_exit=$?
    }
done

```

```

    # Restore the uninstalled library and exit
    if test "$opt_mode" = relink; then
        ( cd "$output_objdir" && \
          $RM "${realname}T" && \
          $MV "${realname}U" "$realname" )
        fi

        exit $lt_exit
    }
done
IFS="$save_ifs"

# Restore the uninstalled library and exit
if test "$opt_mode" = relink; then
    $opt_dry_run || eval '(cd $output_objdir && $RM ${realname}T &&
$MV $realname ${realname}T && $MV ${realname}U $realname)' || exit $?

    if test -n "$convenience"; then
        if test -z "$whole_archive_flag_spec"; then
            func_show_eval '${RM}r "$gentop"'
        fi
    fi

    exit $EXIT_SUCCESS
fi

# Create links to the real library.
for linkname in $linknames; do
    if test "$realname" != "$linkname"; then
        func_show_eval '(cd "$output_objdir" && $RM "$linkname" &&
$LN_S "$realname" "$linkname")' 'exit $?'
    fi
done

# If -module or -export-dynamic was specified, set the dlname.
if test "$module" = yes || test "$export_dynamic" = yes; then
    # On all known operating systems, these are identical.
    dlname="$soname"
fi
fi
;;

obj)
    if test -n "$dlfiles$dlprefiles" || test "$dlself" != no; then
        func_warning "\`-dlopen' is ignored for objects"
    fi

    case " $deplibs" in
        *\ -l* | *\ -L*)
            func_warning "\`-l' and \`-L' are ignored for objects" ;;
    esac

```



```

test -n "$rpath" && \
func_warning "\`-rpath' is ignored for objects"

test -n "$xrpath" && \
func_warning "\`-R' is ignored for objects"

test -n "$vinfo" && \
func_warning "\`-version-info' is ignored for objects"

test -n "$release" && \
func_warning "\`-release' is ignored for objects"

case $output in
*.lo)
test -n "$objs$old_deplibs" && \
func_fatal_error "cannot build library object \`${output}' from
non-libtool objects"

libobj=$output
func_lo2o "$libobj"
obj=$func_lo2o_result
;;
*)
libobj=
obj="$output"
;;
esac

# Delete the old objects.
$opt_dry_run || $RM $obj $libobj

# Objects from convenience libraries. This assumes
# single-version convenience libraries. Whenever we create
# different ones for PIC/non-PIC, this we'll have to duplicate
# the extraction.
reload_conv_objs=
gentop=
# reload_cmds runs $LD directly, so let us get rid of
# -Wl from whole_archive_flag_spec and hope we can get by with
# turning comma into space..
wl=

if test -n "$convenience"; then
if test -n "$whole_archive_flag_spec"; then
eval tmp_whole_archive_flags="\`$whole_archive_flag_spec\`"
reload_conv_objs=$reload_objs\ ` $ECHO
"$tmp_whole_archive_flags" | $SED 's|,| |g'`
else
gentop="$output_objdir/${obj}x"
generated+=" $gentop"

func_extract_archives $gentop $convenience

```

```

    reload_conv_objs="$reload_objs $func_extract_archives_result"
fi
fi

# If we're not building shared, we need to use non_pic_objs
test "$build_libtool_libs" != yes && libobjs="$non_pic_objects"

# Create the old-style object.
reload_objs="$objs$old_deplibs "`$ECHO "$libobjs" | $SP2NL |
$SED "/\.${libext}$/d; /\.lib$/d; $lo2o" | $NL2SP`" $reload_conv_objs"
### testsuite: skip nested quoting test

output="$obj"
func_execute_cmds "$reload_cmds" 'exit $?'

# Exit if we aren't doing a library object file.
if test -z "$libobj"; then
if test -n "$gentop"; then
    func_show_eval '${RM}r "$gentop"'
fi

exit $EXIT_SUCCESS
fi

if test "$build_libtool_libs" != yes; then
if test -n "$gentop"; then
    func_show_eval '${RM}r "$gentop"'
fi

# Create an invalid libtool object if no PIC, so that we don't
# accidentally link it into a program.
# $show "echo timestamp > $libobj"
# $opt_dry_run || eval "echo timestamp > $libobj" || exit $?
exit $EXIT_SUCCESS
fi

if test -n "$pic_flag" || test "$pic_mode" != default; then
# Only do commands if we really have different PIC objects.
reload_objs="$libobjs $reload_conv_objs"
output="$libobj"
func_execute_cmds "$reload_cmds" 'exit $?'
fi

if test -n "$gentop"; then
func_show_eval '${RM}r "$gentop"'
fi

exit $EXIT_SUCCESS
;;

prog)
case $host in

```

```

*cygwin*) func_stripname ' ' '.exe' "$output"
            output=${func_stripname_result}.exe;;
esac
test -n "$vinfo" && \
func_warning "\`-version-info' is ignored for programs"

test -n "$release" && \
func_warning "\`-release' is ignored for programs"

test "$preload" = yes \
    && test "$dlopen_support" = unknown \
&& test "$dlopen_self" = unknown \
&& test "$dlopen_self_static" = unknown && \
    func_warning "\`LT_INIT([dlopen])' not used. Assuming no dlopen
support."

case $host in
*-*-rhapsody* | *-*-darwin1.[012])
# On Rhapsody replace the C library is the System framework
compile_deplibs=`$ECHO " $compile_deplibs" | $SED 's/ -lc /
System.ltframework /'`
finalize_deplibs=`$ECHO " $finalize_deplibs" | $SED 's/ -lc /
System.ltframework /'`
;;
esac

case $host in
*-*-darwin*)
# Don't allow lazy linking, it breaks C++ global constructors
# But is supposedly fixed on 10.4 or later (yay!).
if test "$tagname" = CXX ; then
    case ${MACOSX_DEPLOYMENT_TARGET-10.0} in
    10.[0123])
        compile_command+=" ${wl}-bind_at_load"
        finalize_command+=" ${wl}-bind_at_load"
        ;;
    esac
fi
# Time to change all our "foo.ltframework" stuff back to "-
framework foo"
compile_deplibs=`$ECHO " $compile_deplibs" | $SED 's% \([^
$]*\)\.ltframework% -framework \1%g'`
finalize_deplibs=`$ECHO " $finalize_deplibs" | $SED 's% \([^
$]*\)\.ltframework% -framework \1%g'`
;;
esac

# move library search paths that coincide with paths to not yet
# installed libraries to the beginning of the library search
list
new_libs=

```

```

for path in $notinst_path; do
case " $new_libs " in
*" -L$path/$objdir "*) ;;
*)
    case " $compile_deplibs " in
    *" -L$path/$objdir "*)
        new_libs+="-L$path/$objdir" ;;
    esac
    ;;
esac
done
for deplib in $compile_deplibs; do
case $deplib in
-L*)
    case " $new_libs " in
    *" $deplib "*) ;;
    *) new_libs+="$deplib" ;;
    esac
    ;;
*) new_libs+="$deplib" ;;
esac
done
compile_deplibs="$new_libs"

compile_command+=" $compile_deplibs"
finalize_command+=" $finalize_deplibs"

if test -n "$rpath$xrpath"; then
# If the user specified any rpath flags, then add them.
for libdir in $rpath $xrpath; do
    # This is the magic to use -rpath.
    case "$finalize_rpath " in
    *" $libdir "*) ;;
    *) finalize_rpath+="$libdir" ;;
    esac
done
fi

# Now hardcode the library paths
rpath=
hardcode_libdirs=
for libdir in $compile_rpath $finalize_rpath; do
if test -n "$hardcode_libdir_flag_spec"; then
    func_replace_sysroot "$libdir"
    libdir=$func_replace_sysroot_result
    func_stripname '=' ' ' "$libdir"
    libdir=$func_stripname_result
    if test -n "$hardcode_libdir_separator"; then
        if test -z "$hardcode_libdirs"; then
            hardcode_libdirs="$libdir"
        else

```

```

        # Just accumulate the unique libdirs.
        case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
*"${hardcode_libdir_separator}$libdir${hardcode_libdir_separator}"*)
    ;;
    *)
        hardcode_libdirs+="${hardcode_libdir_separator}$libdir"
        ;;
    esac
fi
else
    # We only want to hardcode in an rpath if it isn't in the
    # default dlsearch path.
    func_normal_abspath "$libdir"
    libdir_norm=${func_normal_abspath_result}
    case " $sys_lib_dlsearch_path " in
*" $libdir_norm "*) ;;
    *) eval flag="\${hardcode_libdir_flag_spec}"
        rpath+="${flag}"
        ;;
    esac
fi
elif test -n "$runpath_var"; then
    case "$perm_rpath " in
*" $libdir "*) ;;
    *) perm_rpath+="${libdir}" ;;
    esac
fi
case $host in
*-*-cygwin* | *-*-mingw* | *-*-pw32* | *-*-os2* | *-cegcc*)
    testbindir=`${ECHO} "$libdir" | ${SED} -e 's*/lib$*/bin*'`
    case :$dllsearchpath: in
*:$libdir:*) ;;
    : :) dllsearchpath=$libdir;;
    *) dllsearchpath+=":$libdir";;
    esac
case :$dllsearchpath: in
*:$testbindir:*) ;;
    : :) dllsearchpath=$testbindir;;
    *) dllsearchpath+=":$testbindir";;
    esac
    ;;
esac
done
# Substitute the hardcoded libdirs into the rpath.
if test -n "$hardcode_libdir_separator" &&
test -n "$hardcode_libdirs"; then
libdir="$hardcode_libdirs"
eval rpath="\${hardcode_libdir_flag_spec}"
fi

```

```

compile_rpath="$rpath"

rpath=
hardcode_libdirs=
for libdir in $finalize_rpath; do
if test -n "$hardcode_libdir_flag_spec"; then
  if test -n "$hardcode_libdir_separator"; then
    if test -z "$hardcode_libdirs"; then
      hardcode_libdirs="$libdir"
    else
      # Just accumulate the unique libdirs.
      case
$hardcode_libdir_separator$hardcode_libdirs$hardcode_libdir_separator
in
*"$hardcode_libdir_separator$libdir$hardcode_libdir_separator"*)
  ;;
  *)
    hardcode_libdirs+="$hardcode_libdir_separator$libdir"
  ;;
esac
fi
else
  # We only want to hardcode in an rpath if it isn't in the
  # default dlsearch path.
  case " $sys_lib_dlsearch_path " in
*" $libdir "*) ;;
*) eval flag=\"$hardcode_libdir_flag_spec\"
rpath+=" $flag"
  ;;
esac
fi
elif test -n "$runpath_var"; then
  case "$finalize_perm_rpath " in
*" $libdir "*) ;;
*) finalize_perm_rpath+=" $libdir" ;;
esac
fi
done
# Substitute the hardcoded libdirs into the rpath.
if test -n "$hardcode_libdir_separator" &&
test -n "$hardcode_libdirs"; then
libdir="$hardcode_libdirs"
eval rpath=\ "$hardcode_libdir_flag_spec\"
fi
finalize_rpath="$rpath"

if test -n "$libobjs" && test "$build_old_libs" = yes; then
# Transform all the library objects into standard objects.
compile_command=`ECHO "$compile_command" | $SP2NL | $SED "$lo2o"
| $NL2SP`

```

```

        finalize_command=`$ECHO "$finalize_command" | $SP2NL | $SED
"$lo2o" | $NL2SP`
        fi

        func_generate_dlsyms "$outputname" "@PROGRAM@" "no"

        # template prelinking step
        if test -n "$prelink_cmds"; then
        func_execute_cmds "$prelink_cmds" 'exit $?'
        fi

        wrappers_required=yes
        case $host in
        *cegcc* | *mingw32ce*)
            # Disable wrappers for cegcc and mingw32ce hosts, we are cross
            compiling anyway.
            wrappers_required=no
            ;;
        *cygwin* | *mingw* )
            if test "$build_libtool_libs" != yes; then
                wrappers_required=no
            fi
            ;;
        *)
            if test "$need_relink" = no || test "$build_libtool_libs" !=
yes; then
                wrappers_required=no
            fi
            ;;
        esac
        if test "$wrappers_required" = no; then
        # Replace the output file specification.
        compile_command=`$ECHO "$compile_command" | $SED
's%@OUTPUT@%' "$output" '%g'`
        link_command="$compile_command$compile_rpath"

        # We have no uninstalled library dependencies, so finalize right
        now.
        exit_status=0
        func_show_eval "$link_command" 'exit_status=$?'

        if test -n "$postlink_cmds"; then
            func_to_tool_file "$output"
            postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%' "$output" '%g' -e
's%@TOOL_OUTPUT@%' "$func_to_tool_file_result" '%g'`
            func_execute_cmds "$postlink_cmds" 'exit $?'
        fi

        # Delete the generated files.
        if test -f "$output_objdir/${outputname}S.${objext}"; then
            func_show_eval '$RM "$output_objdir/${outputname}S.${objext}"'

```

```

fi

exit $exit_status
fi

if test -n "$compile_shlibpath$finalize_shlibpath"; then
compile_command="$shlibpath_var=\"${compile_shlibpath$finalize_shlibpath}\${shlibpath_var}\" $compile_command"
fi
if test -n "$finalize_shlibpath"; then
finalize_command="$shlibpath_var=\"${finalize_shlibpath}\${shlibpath_var}\" $finalize_command"
fi

compile_var=
finalize_var=
if test -n "$runpath_var"; then
if test -n "$perm_rpath"; then
# We should set the runpath_var.
rpath=
for dir in $perm_rpath; do
rpath+="$dir:"
done
compile_var="$runpath_var=\"${rpath}\${runpath_var}\" "
fi
if test -n "$finalize_perm_rpath"; then
# We should set the runpath_var.
rpath=
for dir in $finalize_perm_rpath; do
rpath+="$dir:"
done
finalize_var="$runpath_var=\"${rpath}\${runpath_var}\" "
fi
fi

if test "$no_install" = yes; then
# We don't need to create a wrapper script.
link_command="$compile_var$compile_command$compile_rpath"
# Replace the output file specification.
link_command=`$ECHO "$link_command" | $SED
's%@OUTPUT@%' "$output"'%g'`
# Delete the old output file.
$opt_dry_run || $RM $output
# Link the executable and exit
func_show_eval "$link_command" 'exit $?'

if test -n "$postlink_cmds"; then
func_to_tool_file "$output"
postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%' "$output"'%g' -e
's%@TOOL_OUTPUT@%' "$func_to_tool_file_result"'%g'`
func_execute_cmds "$postlink_cmds" 'exit $?'

```



```

fi

exit $EXIT_SUCCESS
fi

if test "$hardcode_action" = relink; then
# Fast installation is not supported
link_command="$compile_var$compile_command$compile_rpath"
relink_command="$finalize_var$finalize_command$finalize_rpath"

func_warning "this platform does not like uninstalled shared
libraries"
func_warning "\`$output' will be relinked during installation"
else
if test "$fast_install" != no; then
link_command="$finalize_var$compile_command$finalize_rpath"
if test "$fast_install" = yes; then
relink_command=`$ECHO
"$compile_var$compile_command$compile_rpath" | $SED
's%@OUTPUT@%\$progdir/\$file%g'`
else
# fast_install is set to needless
relink_command=
fi
else
link_command="$compile_var$compile_command$compile_rpath"
relink_command="$finalize_var$finalize_command$finalize_rpath"
fi
fi

# Replace the output file specification.
link_command=`$ECHO "$link_command" | $SED
's%@OUTPUT@%'"$output_objdir/$outputname"'"%g'`

# Delete the old output files.
$opt_dry_run || $RM $output $output_objdir/$outputname
$output_objdir/lt-$outputname

func_show_eval "$link_command" 'exit $?'

if test -n "$postlink_cmds"; then
func_to_tool_file "$output_objdir/$outputname"
postlink_cmds=`func_echo_all "$postlink_cmds" | $SED -e
's%@OUTPUT@%'"$output_objdir/$outputname"'"%g' -e
's%@TOOL_OUTPUT@%'"$func_to_tool_file_result"'"%g'`
func_execute_cmds "$postlink_cmds" 'exit $?'
fi

# Now create the wrapper script.
func_verbose "creating $output"

# Quote the relink command for shipping.

```

```

if test -n "$relink_command"; then
# Preserve any variables that may affect compiler behavior
for var in $variables_saved_for_relink; do
  if eval test -z "\${$var+set}"; then
    relink_command="{ test -z "\${$var+set}" || $lt_unset $var
|| { $var=; export $var; }; }; $relink_command"
    elif eval var_value=\${$var}; test -z "$var_value"; then
      relink_command="$var=; export $var; $relink_command"
    else
      func_quote_for_eval "$var_value"
      relink_command="$var=${func_quote_for_eval_result}; export
$var; $relink_command"
    fi
  done
  relink_command="(cd `pwd`; $relink_command)"
  relink_command=`$ECHO "$relink_command" | $SED
"$sed_quote_subst"`
fi

```

```

# Only actually do things if not in dry run mode.
$opt_dry_run || {
# win32 will think the script is a binary if it has
# a .exe suffix, so we strip it off here.
case $output in
*.exe) func_stripname '' '.exe' "$output"
        output=${func_stripname_result} ;;
esac
# test for cygwin because mv fails w/o .exe extensions
case $host in
*cygwin*)
  exeext=.exe
  func_stripname '' '.exe' "$outputname"
  outputname=${func_stripname_result} ;;
*) exeext= ;;
esac
case $host in
*cygwin* | *mingw* )
  func_dirname_and_basename "$output" "" "."
  output_name=${func_basename_result}
  output_path=${func_dirname_result}
  cwrappersource="$output_path/$objdir/lt-$output_name.c"
  cwrapper="$output_path/$output_name.exe"
  $RM $cwrappersource $cwrapper
  trap "$RM $cwrappersource $cwrapper; exit $EXIT_FAILURE" 1 2

```

15

```

func_emit_cwrapperexe_src > $cwrappersource

```

```

# The wrapper executable is built using the $host compiler,
# because it contains $host paths and files. If cross-
# compiling, it, like the target executable, must be
# executed on the $host or under an emulation environment.

```

```

    $opt_dry_run || {
        $LTCC $LTCCFLAGS -o $cwrapper $cwrappersource
        $STRIP $cwrapper
    }

    # Now, create the wrapper script for func_source use:
    func_ltwrapper_scriptname $cwrapper
    $RM $func_ltwrapper_scriptname_result
    trap "$RM $func_ltwrapper_scriptname_result; exit
$EXIT_FAILURE" 1 2 15
    $opt_dry_run || {
        # note: this script will not be executed, so do not chmod.
        if test "x$build" = "x$host" ; then
            $cwrapper --lt-dump-script >
$func_ltwrapper_scriptname_result
        else
            func_emit_wrapper no > $func_ltwrapper_scriptname_result
        fi
    }
    ;;
* )
    $RM $output
    trap "$RM $output; exit $EXIT_FAILURE" 1 2 15

    func_emit_wrapper no > $output
    chmod +x $output
    ;;
esac
}
exit $EXIT_SUCCESS
;;
esac

# See if we need to build an old-fashioned archive.
for oldlib in $oldlibs; do

    if test "$build_libtool_libs" = convenience; then
        oldobjs="$libobjs_save $symfileobj"
        addlibs="$convenience"
        build_libtool_libs=no
    else
        if test "$build_libtool_libs" = module; then
            oldobjs="$libobjs_save"
            build_libtool_libs=no
        else
            oldobjs="$old_deplibs $non_pic_objects"
            if test "$preload" = yes && test -f "$symfileobj"; then
                oldobjs+="$symfileobj"
            fi
        fi
        addlibs="$old_convenience"
    fi

```

```

    if test -n "$addlibs"; then
gentop="$output_objdir/${outputname}x"
generated+=" $gentop"

func_extract_archives $gentop $addlibs
oldobjs+=" $func_extract_archives_result"
    fi

    # Do each command in the archive commands.
    if test -n "$old_archive_from_new_cmds" && test
"$build_libtool_libs" = yes; then
cmds=$old_archive_from_new_cmds
    else

# Add any objects from preloaded convenience libraries
if test -n "$dlprefiles"; then
gentop="$output_objdir/${outputname}x"
generated+=" $gentop"

func_extract_archives $gentop $dlprefiles
oldobjs+=" $func_extract_archives_result"
fi

# POSIX demands no paths to be encoded in archives. We have
# to avoid creating archives with duplicate basenames if we
# might have to extract them afterwards, e.g., when creating a
# static archive out of a convenience library, or when linking
# the entirety of a libtool archive into another (currently
# not supported by libtool).
if (for obj in $oldobjs
do
func_basename "$obj"
$ECHO "$func_basename_result"
done | sort | sort -uc >/dev/null 2>&1); then
:
else
echo "copying selected object files to avoid basename
conflicts..."
gentop="$output_objdir/${outputname}x"
generated+=" $gentop"
func_mkdir_p "$gentop"
save_oldobjs=$oldobjs
oldobjs=
counter=1
for obj in $save_oldobjs
do
func_basename "$obj"
objbase="$func_basename_result"
case " $oldobjs " in
" ") oldobjs=$obj ;;
*[\ /]"$objbase "*)

```

```

        while ;; do
        # Make sure we don't pick an alternate name that also
        # overlaps.
        newobj=lt$counter-$objbase
        func_arith $counter + 1
        counter=$func_arith_result
        case " $oldobjs " in
        *[\ /]"$newobj "*) ;;
        *) if test ! -f "$gentop/$newobj"; then break; fi ;;
        esac
        done
        func_show_eval "ln $obj $gentop/$newobj || cp $obj
$gentop/$newobj"
        oldobjs+=" $gentop/$newobj"
        ;;
        *) oldobjs+=" $obj" ;;
        esac
    done
fi
func_to_tool_file "$oldlib" func_convert_file_msys_to_w32
tool_oldlib=$func_to_tool_file_result
eval cmds="\$old_archive_cmds\"

func_len " $cmds"
len=$func_len_result
if test "$len" -lt "$max_cmd_len" || test "$max_cmd_len" -le -1;
then
    cmds=$old_archive_cmds
elif test -n "$archiver_list_spec"; then
    func_verbose "using command file archive linking..."
    for obj in $oldobjs
    do
        func_to_tool_file "$obj"
        $ECHO "$func_to_tool_file_result"
        done > $output_objdir/$libname.libcmd
        func_to_tool_file "$output_objdir/$libname.libcmd"
        oldobjs=" $archiver_list_spec$func_to_tool_file_result"
        cmds=$old_archive_cmds
    else
        # the command line is too long to link in one step, link in
parts
        func_verbose "using piecewise archive linking..."
        save_RANLIB=$RANLIB
        RANLIB=:
        objlist=
        concat_cmds=
        save_oldobjs=$oldobjs
        oldobjs=
        # Is there a better way of finding the last object in the list?
        for obj in $save_oldobjs
        do
            last_oldobj=$obj

```

```

done
eval test_cmds="\$old_archive_cmds\"
func_len " $test_cmds"
len0=$func_len_result
len=$len0
for obj in $save_oldobjs
do
    func_len " $obj"
    func_arith $len + $func_len_result
    len=$func_arith_result
    objlist+="$obj"
    if test "$len" -lt "$max_cmd_len"; then
        :
    else
        # the above command should be used before it gets too long
        oldobjs=$objlist
        if test "$obj" = "$last_oldobj" ; then
            RANLIB=$save_RANLIB
            fi
            test -z "$concat_cmds" || concat_cmds=$concat_cmds~
            eval concat_cmds="\${concat_cmds}$old_archive_cmds\"
            objlist=
            len=$len0
        fi
    done
    RANLIB=$save_RANLIB
    oldobjs=$objlist
    if test "X$oldobjs" = "X" ; then
        eval cmds="\${concat_cmds}\"
    else
        eval cmds="\${concat_cmds}~\${old_archive_cmds}\"
    fi
fi
fi
func_execute_cmds "$cmds" 'exit $?'
done

test -n "$generated" && \
    func_show_eval "${RM}r$generated"

# Now create the libtool archive.
case $output in
*.la)
    old_library=
    test "$build_old_libs" = yes && old_library="$libname.$libext"
    func_verbose "creating $output"

    # Preserve any variables that may affect compiler behavior
    for var in $variables_saved_for_relink; do
        if eval test -z \${$var+set}\; then
            relink_command="{ test -z \${$var+set}\ || $lt_unset $var ||
{ $var=; export $var; }; }; $relink_command"

```

```

elif eval var_value=\$$var; test -z "$var_value"; then
    relink_command="$var=; export $var; $relink_command"
else
    func_quote_for_eval "$var_value"
    relink_command="$var=${func_quote_for_eval_result}; export $var;
$relink_command"
fi
done
# Quote the link command for shipping.
relink_command="(cd `pwd`; $SHELL $progp_path $preserve_args --
mode=relink $libtool_args @inst_prefix_dir@)"
relink_command=`$ECHO "$relink_command" | $SED
"$sed_quote_subst"`
if test "$hardcode_automatic" = yes ; then
relink_command=
fi

# Only create the output if not a dry run.
$opt_dry_run || {
for installed in no yes; do
    if test "$installed" = yes; then
        if test -z "$install_libdir"; then
            break
        fi
        output="$output_objdir/$outputname"i
        # Replace all uninstalled libtool libraries with the
installed ones
        newdependency_libs=
        for deplib in $dependency_libs; do
            case $deplib in
                *.la)
                    func_basename "$deplib"
                    name="$func_basename_result"
                    func_resolve_sysroot "$deplib"
                    eval libdir=`${SED} -n -e 's/^libdir=\.*)$/\1/p'
$func_resolve_sysroot_result`
                    test -z "$libdir" && \
                        func_fatal_error "`$deplib' is not a valid libtool
archive"
                    newdependency_libs+="$ {lt_sysroot:+}$libdir/$name"
                    ;;
                -L*)
                    func_stripname -L '' "$deplib"
                    func_replace_sysroot "$func_stripname_result"
                    newdependency_libs+="$ -L$func_replace_sysroot_result"
                    ;;
                -R*)
                    func_stripname -R '' "$deplib"
                    func_replace_sysroot "$func_stripname_result"
                    newdependency_libs+="$ -R$func_replace_sysroot_result"
                    ;;
                *) newdependency_libs+="$ deplib" ;;
            esac
        done
    fi
done
}

```

```

    esac
done
dependency_libs="$newdependency_libs"
newdlfiles=

for lib in $dlfiles; do
    case $lib in
        *.la)
            func_basename "$lib"
            name="$func_basename_result"
            eval libdir=`${SED} -n -e 's/^libdir=\(.*\)$/\1/p' $lib`
            test -z "$libdir" && \
                func_fatal_error "`$lib' is not a valid libtool archive"
            newdlfiles+="${lt_sysroot:+}$libdir/$name"
            ;;
        *) newdlfiles+="$lib" ;;
    esac
done
dlfiles="$newdlfiles"
newdlprefiles=
for lib in $dlprefiles; do
    case $lib in
        *.la)
            # Only pass preopened files to the pseudo-archive (for
            # eventual linking with the app. that links it) if we
            # didn't already link the preopened objects directly into
            # the library:
            func_basename "$lib"
            name="$func_basename_result"
            eval libdir=`${SED} -n -e 's/^libdir=\(.*\)$/\1/p' $lib`
            test -z "$libdir" && \
                func_fatal_error "`$lib' is not a valid libtool archive"
            newdlprefiles+="${lt_sysroot:+}$libdir/$name"
            ;;
    esac
done
dlprefiles="$newdlprefiles"
else
newdlfiles=
for lib in $dlfiles; do
    case $lib in
        [\\/] * | [A-Za-z]:[\\/] *) abs="$lib" ;;
        *) abs=`pwd`"/$lib" ;;
    esac
    newdlfiles+="$abs"
done
dlfiles="$newdlfiles"
newdlprefiles=
for lib in $dlprefiles; do
    case $lib in
        [\\/] * | [A-Za-z]:[\\/] *) abs="$lib" ;;
        *) abs=`pwd`"/$lib" ;;
    esac
done

```



```

        esac
        newdlprefiles+=" $abs"
    done
    dlprefiles="$newdlprefiles"
fi
$RM $output
# place dlname in correct position for cygwin
# In fact, it would be nice if we could use this code for all
target
# systems that can't hard-code library paths into their
executables
# and that have no shared library path variable independent of
PATH,
# but it turns out we can't easily determine that from
inspecting
# libtool variables, so we have to hard-code the OSs to which
it
# applies here; at the moment, that means platforms that use
the PE
# object format with DLL files. See the long comment at the
top of
# tests/bindir.at for full details.
tdlname=$dlname
case $host,$output,$installed,$module,$dlname in
    *cygwin*,*lai,yes,no,*.dll | *mingw*,*lai,yes,no,*.dll |
*cegcc*,*lai,yes,no,*.dll)
    # If a -bindir argument was supplied, place the dll there.
    if test "x$bindir" != x ;
    then
        func_relative_path "$install_libdir" "$bindir"
        tdlname=$func_relative_path_result$dlname
    else
        # Otherwise fall back on heuristic.
        tdlname=../bin/$dlname
    fi
    ;;
esac
    $ECHO > $output "\
# $outputname - a libtool library file
# Generated by $PROGRAM (GNU $PACKAGE$TIMESTAMP) $VERSION
#
# Please DO NOT delete this file!
# It is necessary for linking the library.

# The name that we can dlopen(3).
dlname='$tdlname'

# Names of this library.
library_names='$library_names'

# The name of the static archive.
old_library='$old_library'

```

```

# Linker flags that can not go in dependency_libs.
inherited_linker_flags='$new_inherited_linker_flags'

# Libraries that this one depends upon.
dependency_libs='$dependency_libs'

# Names of additional weak libraries provided by this library
weak_library_names='$weak_libs'

# Version information for $libname.
current=$current
age=$age
revision=$revision

# Is this an already installed library?
installed=$installed

# Should we warn about portability when linking against -modules?
shouldnotlink=$module

# Files to dlopen/dlpreopen
dlopen='$dlfiles'
dlpreopen='$dlprefiles'

# Directory that this library needs to be installed in:
libdir='$install_libdir'
    if test "$installed" = no && test "$need_relink" = yes; then
        $ECHO >> $output "\
relink_command=\"\${relink_command}\""
        fi
    done
}

# Do a symbolic link so that the libtool archive can be found in
# LD_LIBRARY_PATH before the program is installed.
func_show_eval '( cd "$output_objdir" && $RM "$outputname" &&
$LN_S "../$outputname" "$outputname" )' 'exit $?'
;;
esac
exit $EXIT_SUCCESS
}

{ test "$opt_mode" = link || test "$opt_mode" = relink; } &&
  func_mode_link ${1+"$@"}

# func_mode_uninstall arg...
func_mode_uninstall ()
{
  $opt_debug
  RM="$nonopt"

```

```

files=
rmforce=
exit_status=0

# This variable tells wrapper scripts just to set variables rather
# than running their programs.
libtool_install_magic="$magic"

for arg
do
  case $arg in
    -f) RM+=" $arg"; rmforce=yes ;;
    -*) RM+=" $arg" ;;
    *) files+=" $arg" ;;
  esac
done

test -z "$RM" && \
  func_fatal_help "you must specify an RM program"

rmdirs=

for file in $files; do
  func_dirname "$file" "" "."
  dir="$func_dirname_result"
  if test "X$dir" = X.; then
    odir="$objdir"
  else
    odir="$dir/$objdir"
  fi
  func_basename "$file"
  name="$func_basename_result"
  test "$opt_mode" = uninstall && odir="$dir"

  # Remember odir for removal later, being careful to avoid
duplicates
  if test "$opt_mode" = clean; then
    case " $rmdirs " in
      *" $odir ") ;;
      *) rmdirs+=" $odir" ;;
    esac
  fi

  # Don't error if the file doesn't exist and rm -f was used.
  if { test -L "$file"; } >/dev/null 2>&1 ||
    { test -h "$file"; } >/dev/null 2>&1 ||
    test -f "$file"; then
    :
  elif test -d "$file"; then
    exit_status=1
    continue
  elif test "$rmforce" = yes; then

```

```

continue
fi

rmfiles="$file"

case $name in
*.la)
# Possibly a libtool archive, so verify it.
if func_lalib_p "$file"; then
    func_source $dir/$name

    # Delete the libtool libraries and symlinks.
    for n in $library_names; do
        rmfiles+=" $odir/$n"
    done
    test -n "$old_library" && rmfiles+=" $odir/$old_library"

    case "$opt_mode" in
clean)
        case " $library_names " in
        *" $dlname ") ;;
        *) test -n "$dlname" && rmfiles+=" $odir/$dlname" ;;
        esac
        test -n "$libdir" && rmfiles+=" $odir/$name $odir/${name}i"
        ;;
uninstall)
        if test -n "$library_names"; then
            # Do each command in the postuninstall commands.
            func_execute_cmds "$postuninstall_cmds" 'test "$rmforce" =
yes || exit_status=1'
        fi

        if test -n "$old_library"; then
            # Do each command in the old_postuninstall commands.
            func_execute_cmds "$old_postuninstall_cmds" 'test
"$rmforce" = yes || exit_status=1'
        fi
        # FIXME: should reinstall the best remaining shared library.
        ;;
    esac
fi
;;

*.lo)
# Possibly a libtool object, so verify it.
if func_lalib_p "$file"; then

    # Read the .lo file
    func_source $dir/$name

    # Add PIC object to the list of files to remove.
    if test -n "$pic_object" &&

```

```

        test "$pic_object" != none; then
    rmfiles+=" $dir/$pic_object"
fi

# Add non-PIC object to the list of files to remove.
if test -n "$non_pic_object" &&
    test "$non_pic_object" != none; then
    rmfiles+=" $dir/$non_pic_object"
fi
fi
;;

*)
if test "$opt_mode" = clean ; then
    noexename=$name
    case $file in
    *.exe)
        func_stripname ' ' '.exe' "$file"
        file=$func_stripname_result
        func_stripname ' ' '.exe' "$name"
        noexename=$func_stripname_result
        # $file with .exe has already been added to rmfiles,
        # add $file without .exe
        rmfiles+=" $file"
        ;;
    esac
    # Do a test to see if this is a libtool program.
    if func_ltwrapper_p "$file"; then
        if func_ltwrapper_executable_p "$file"; then
            func_ltwrapper_scriptname "$file"
            relink_command=
            func_source $func_ltwrapper_scriptname_result
            rmfiles+=" $func_ltwrapper_scriptname_result"
        else
            relink_command=
            func_source $dir/$noexename
        fi

        # note $name still contains .exe if it was in $file
originally
        # as does the version of $file that was added into $rmfiles
        rmfiles+=" $odir/$name $odir/${name}S.${objext}"
        if test "$fast_install" = yes && test -n "$relink_command";
then
            rmfiles+=" $odir/lt-$name"
        fi
        if test "X$noexename" != "X$name" ; then
            rmfiles+=" $odir/lt-{$noexename}.c"
        fi
    fi
fi
fi
;;

```

```

        esac
        func_show_eval "$RM $rmfiles" 'exit_status=1'
    done

    # Try to remove the ${objdir}s in the directories where we deleted
files
    for dir in $rmdirs; do
        if test -d "$dir"; then
            func_show_eval "rmdir $dir >/dev/null 2>&1"
        fi
    done

    exit $exit_status
}

{ test "$opt_mode" = uninstall || test "$opt_mode" = clean; } &&
    func_mode_uninstall ${1+"$@"}

test -z "$opt_mode" && {
    help="$generic_help"
    func_fatal_help "you must specify a MODE"
}

test -z "$exec_cmd" && \
    func_fatal_help "invalid operation mode \`$opt_mode'"

if test -n "$exec_cmd"; then
    eval exec "$exec_cmd"
    exit $EXIT_FAILURE
fi

exit $exit_status

# The TAGs below are defined such that we never get into a situation
# in which we disable both kinds of libraries.  Given conflicting
# choices, we go for a static library, that is the most portable,
# since we can't tell whether shared libraries were disabled because
# the user asked for that or because the platform doesn't support
# them.  This is particularly important on AIX, because we don't
# support having both static and shared libraries enabled at the same
# time on that platform, so we default to a shared-only configuration.
# If a disable-shared tag is given, we'll fallback to a static-only
# configuration.  But we'll never go from static-only to shared-only.

# ### BEGIN LIBTOOL TAG CONFIG: disable-shared
build_libtool_libs=no
build_old_libs=yes
# ### END LIBTOOL TAG CONFIG: disable-shared

# ### BEGIN LIBTOOL TAG CONFIG: disable-static

```

```
build_old_libs=`case $build_libtool_libs in yes) echo no;; *) echo
yes;; esac`
# ### END LIBTOOL TAG CONFIG: disable-static

# Local Variables:
# mode:shell-script
# sh-indentation:2
# End:
# vi:sw=2

# ### BEGIN LIBTOOL TAG CONFIG: CXX

# The linker used to build libraries.
LD="arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# How to create reloadable object files.
reload_flag=" -r"
reload_cmds="\$LD\$reload_flag -o \$output\$reload_objs"

# Commands used to build an old-style archive.
old_archive_cmds="\$AR \$AR_FLAGS \$oldlib\$oldobjs~\$RANLIB
\$tool_oldlib"

# A language specific compiler.
CC="arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"

# Is the compiler the GNU compiler?
with_gcc=yes

# Compiler flag to turn off builtin functions.
no_builtin_flag=" -fno-builtin"

# Additional compiler flags for building library objects.
pic_flag=" -fPIC -DPIC"

# How to pass a linker flag through the compiler.
wl="-Wl,"

# Compiler flag to prevent dynamic linking.
link_static_flag="--static"

# Does compiler simultaneously support -c and -o options?
compiler_c_o="yes"

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=no
```

```
# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=no

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec="\${wl}--export-dynamic"

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec="\${wl}--whole-archive\${convenience} \${wl}--
no-whole-archive"

# Whether the compiler copes with passing no objects directly.
compiler_needs_object="no"

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=""

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=""

# Commands used to build a shared archive.
archive_cmds="\${CC} \${pic_flag} -shared -nostdlib \${predep_objects}
\${libobjs} \${deplibs} \${postdep_objects} \${compiler_flags} \${wl}-soname
\${wl}\${soname} -o \${lib}"
archive_expsym_cmds="\${CC} \${pic_flag} -shared -nostdlib
\${predep_objects} \${libobjs} \${deplibs} \${postdep_objects}
\${compiler_flags} \${wl}-soname \${wl}\${soname} \${wl}-retain-symbols-file
\${wl}\${export_symbols} -o \${lib}"

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=""
module_expsym_cmds=""

# Whether we are building with GNU ld or not.
with_gnu_ld="yes"

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=""

# Flag that enforces no undefined symbols.
no_undefined_flag=""

# Flag to hardcode $libdir into a binary during linking.
# This must work even if $libdir does not exist
hardcode_libdir_flag_spec="\${wl}-rpath \${wl}\${libdir}"

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=""
```



```
# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=no

# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting ${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=no

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=no

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=unsupported

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=no

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=no

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=unknown

# Set to "yes" if exported symbols are required.
always_export_symbols=no

# The commands to list exported symbols.
export_symbols_cmds="\$NM \$libobjs \$convenience |
\$global_symbol_pipe | \$SED 's/.*/>' | sort | uniq >
\$export_symbols"

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms="_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*"

# Symbols that must always be exported.
include_expsyms=""

# Commands necessary for linking programs (against libraries) with
templates.
```

```

prelink_cmds=""

# Commands necessary for finishing linking programs.
postlink_cmds=""

# Specify filename containing input files.
file_list_spec=""

# How to hardcode a shared library path into an executable.
hardcode_action=immediate

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs="/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi-4.7.3 /home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/./lib/gcc /home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi /home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi-4.7.3/../../../../arm-linux-gnueabi/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib"

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects="/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-vp1/usr/lib/crti.o
/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/bin/./lib/gcc/arm-linux-gnueabi-4.7.3/crtbeginS.o"
postdep_objects="/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi-4.7.3/crtendS.o
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/crtn.o"
predeps=""
postdeps="-lstdc++ -lm -lgcc_s -lc -lgcc_s"

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path="-L/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi-4.7.3 -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/./lib/gcc -L/home/gangadhar/gcc-linaro-
arm-linux-gnueabi-4.7-2013.03-20130313_linux/bin/./lib/gcc/arm-
linux-gnueabi -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/./lib/gcc/arm-linux-
gnueabi-4.7.3/../../../../arm-linux-gnueabi/lib -
L/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib -

```

```
L/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib"

# ### END LIBTOOL TAG CONFIG: CXX

# ### BEGIN LIBTOOL TAG CONFIG: RC

# The linker used to build libraries.
LD=""

# How to create reloadable object files.
reload_flag=""
reload_cmds=""

# Commands used to build an old-style archive.
old_archive_cmds=""

# A language specific compiler.
CC=""

# Is the compiler the GNU compiler?
with_gcc=

# Compiler flag to turn off builtin functions.
no_builtin_flag=""

# Additional compiler flags for building library objects.
pic_flag=""

# How to pass a linker flag through the compiler.
wl=""

# Compiler flag to prevent dynamic linking.
link_static_flag=""

# Does compiler simultaneously support -c and -o options?
compiler_c_o="yes"

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=""

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=""

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=""
```

```
# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=""

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=""

# Commands used to build a shared archive.
archive_cmds=""
archive_expsym_cmds=""

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=""
module_expsym_cmds=""

# Whether we are building with GNU ld or not.
with_gnu_ld=""

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=""

# Flag that enforces no undefined symbols.
no_undefined_flag=""

# Flag to hardcode $libdir into a binary during linking.
# This must work even if $libdir does not exist
hardcode_libdir_flag_spec=""

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=""

# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=

# Set to "yes" if using DIR/libNAME${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting ${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=
```

```
# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=

# Set to "yes" if exported symbols are required.
always_export_symbols=

# The commands to list exported symbols.
export_symbols_cmds=""

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=""

# Symbols that must always be exported.
include_expsyms=""

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=""

# Commands necessary for finishing linking programs.
postlink_cmds=""

# Specify filename containing input files.
file_list_spec=""

# How to hardcode a shared library path into an executable.
hardcode_action=

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=""

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=""
postdep_objects=""
predeps=""
```

```
postdeps=""

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=""

# ### END LIBTOOL TAG CONFIG: RC

File = array-of-array-of-uint32.message

# Message with an array of array of uint32

VALID_HEADER method_call
REQUIRED_FIELDS
ALIGN 8
END_LENGTH Header
START_LENGTH Body

TYPE ARRAY
TYPE ARRAY
TYPE UINT32

LENGTH Array
START_LENGTH Array

## array of uint32
LENGTH SubArray1
START_LENGTH SubArray1
UINT32 1
UINT32 2
UINT32 3
END_LENGTH SubArray1

## array of uint32
LENGTH SubArray2
START_LENGTH SubArray2
UINT32 4
UINT32 5
END_LENGTH SubArray2

END_LENGTH Array

END_LENGTH Body

File = array-of-nil.message

# Message with an array of NIL (not allowed)
```

```
VALID_HEADER method_call
```

```
HEADER_FIELD INTERFACE  
TYPE STRING  
STRING 'org.freedesktop.Foo'  
HEADER_FIELD MEMBER  
TYPE STRING  
STRING 'Bar'  
HEADER_FIELD PATH  
TYPE OBJECT_PATH  
OBJECT_PATH '/foo'
```

```
ALIGN 8  
END_LENGTH Header
```

```
START_LENGTH Body  
TYPE ARRAY  
TYPE NIL  
UINT32 5
```

```
## we want it to fail because of type nil, not because the length is  
no good
```

```
## so pad out the message with 5 bytes
```

```
BYTE 1  
BYTE 2  
BYTE 3  
BYTE 4  
BYTE 5
```

```
END_LENGTH Body
```

```
File = array-with-mixed-types.message
```

```
# Message with an array of array where the child arrays are of  
# different types
```

```
VALID_HEADER method_call
```

```
HEADER_FIELD INTERFACE  
TYPE STRING  
STRING 'org.freedesktop.Foo'  
HEADER_FIELD MEMBER  
TYPE STRING  
STRING 'Bar'  
HEADER_FIELD PATH  
TYPE OBJECT_PATH  
OBJECT_PATH '/foo'
```

```
ALIGN 8  
END_LENGTH Header
```

```
START_LENGTH Body

TYPE ARRAY
TYPE ARRAY
TYPE UINT32

LENGTH Array
START_LENGTH Array

## array of uint32
LENGTH SubArray1
START_LENGTH SubArray1
UINT32 1
UINT32 2
UINT32 3
END_LENGTH SubArray1

## array of uint32
LENGTH SubArray2
START_LENGTH SubArray2
UINT32 4
UINT32 5
END_LENGTH SubArray2

## array of boolean
LENGTH SubArray3
START_LENGTH SubArray3
BOOLEAN false
BOOLEAN true
END_LENGTH SubArray3

END_LENGTH Array

END_LENGTH Body
```

File = AUTHORS

```
Ross Burton <ross@openedhand.com>
S. Nallammai <snallammai@novell.com>
Luiz Augusto von Dentz <luiz.dentz@gmail.com>
Steve FrÃ©cinaux <steve@istique_dot_net>
Marc-Andre Lureau <marcandre.lureau@gmail.com>
Rob Taylor <rob.taylor@codethink.co.uk>
Olivier Andrieu <oliv__a@users.sourceforge.net>
Philip Blundell <pb@nexus.co.uk>
Anders Carlsson <andersca@gnome.org>
Kristian Hogsberg <krh@redhat.com>
Alex Larsson <alex1@redhat.com>
Robert McQueen <robot101@debian.org>
```


Michael Meeks <michael@ximian.com>
Seth Nickell <seth@gnome.org>
John (J5) Palmieri <johnp@redhat.com>
Havoc Pennington <hp@redhat.com>
Harri Porten <porten@kde.org>
Matthew Rickard <mjricka@epoch.ncsc.mil>
Zack Rusin <zack@kde.org>
Joe Shaw <joe@assbarn.com>
Colin Walters <walters@verbum.org>
David Zeuthen <david@fubar.dk>

File = AUTHORS.~1~

Alban Crequy <alban.crequy@collabora.co.uk>
Alexander Larsson <alexl@redhat.com>
Anders Carlsson <andersca@codefactory.se>
Andre Heinecke <aheinecke@intevation.de>
Artem Bityutskiy <Artem.Bituyutskiy@nokia.com>
Aurelien Jarno <aurel32@debian.org>
Benjamin Reed <rangerrick@befunk.com>
Brian Cameron <brian.cameron@oracle.com>
Brian Cameron <brian.cameron@sun.com>
Chris Mayo <aklhfx@gmail.com>
Christian Dywan <christian.dywan@lanedo.com>
Christian Ehrlicher <Ch.Ehrlicher@gmx.de>
Christian Persch (GNOME) <chpe@gnome.org>
Colin Walters <walters@verbum.org>
Colin Watson <cjwatson@ubuntu.com>
Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Cristian Rodríguez < cristian.rodriguez@opensuse.org>
Cyril Brulebois <kibi@debian.org>
Daniel P. Berrange <dan@berrange.com>
Daniel Reed <djr@redhat.com>
David Zeuthen <davidz@redhat.com>
Dennis Kaarsemaker <dennis@kaarsemaker.net>
Diego E. 'Flameeyes' PettenÃ² <flameeyes@gmail.com>
Eamon Walsh <ewalsh@tycho.nsa.gov>
Federico Mena Quintero <federico@novell.com>
Frank Osterfeld <frank@kdab.net>
Frederic Crozat <fcrozat@mandriva.com>
Fridrich Å trba <fridrich.strba@bluewin.ch>
Grzegorz DÄ...browski <gdx@poczta.fm>
Harald Fernengel <harry@kdevelop.org>
Harri Porten <porten@kde.org>
Hasso Tepper <hasso@estpak.ee>
Havoc Pennington <hp@pobox.com>
Havoc Pennington <hp@redhat.com>
Hendrik Buschmeier <hbuschme@TechFak.Uni-Bielefeld.DE>
James Carter <jwcart2@tycho.nsa.gov>

James Westby <jw+debian@jameswestby.net>
James Willcox <jwillcox@gnome.org>
Jens Granseuer <jensgr@gmx.net>
Jeremie Dimino <jeremie@dimino.org>
Jiri Klimes <jklimes@redhat.com>
Joe Marcus Clarke <marcus@freedesktop.org>
Joe Shaw <joeshaw@novell.com>
Johan Gyllenspetz <johangy@axis.com>
Johannes Carlsson <johannes.carlsson.x@sonyericsson.com>
John (J5) Palmieri <johnp@redhat.com>
Jon Gosting <yukarionsen@gmail.com>
Jon Trowbridge <trow@ximian.com>
Kay Sievers <kay.sievers@vrfy.org>
Kimmo Hamalainen <kimmo.hamalainen@nokia.com>
Kjartan Maraas <kmaraas@gnome.org>
Kristian Hagsberg <krh@redhat.com>
Kristian Rietveld <kris@gtk.org>
Kurt Miller <kurt@intricatesoftware.com>
Lawrence R. Steeger <lsteeger@gmail.com>
Lennart Poettering <lennart@poettering.net>
Lennart Poettering <mzqohf@0pointer.de>
Lionel Landwerlin <lionel.landwerlin@openwide.fr>
Luiz Augusto Von Dentz <luiz.dentz-von@nokia.com>
Marc Brockschmidt <he@debian.org>
Marc Mutz <marc@kdab.net>
Marcus Brinkmann <marcus.brinkmann@ruhr-uni-bochum.de>
Mark Brand <mabrand@mabrand.nl>
Mark McLoughlin <mark@skynet.ie>
Matthias Clasen <mclasen@redhat.com>
Matt McCutchen <matt@mattmccutchen.net>
Michael Meeks <michael.meeks@novell.com>
Michael Meeks <michael@ximian.com>
Mikael Hallendal <micke@codefactory.se>
Mikael Hallendal <micke@imendio.com>
Mike McQuaid <mike@mikemcquaid.com>
Miloslav Trmac <mitr@volny.cz>
Murray Cumming <murrayc@murrayc.com>
Nguy n Thi Ng c Duy <pclouds@gmail.com>
Ole Andr  Vadla Ravn s <ole.andre.ravnas@tandberg.com>
Olivier Andrieu <oliv__a@users.sourceforge.net>
Oswald Buddenhagen <ossi@kde.org>
Owen Fraser-Green <owen@discobabe.net>
Patrick von Reth <patrick.vonreth@gmail.com>
Pau Garcia i Quiles <pgquiles@elpauer.org>
Paul Bolle <pebolle@tiscali.nl>
Peter Breitenlohner <peb@mppmu.mpg.de>
Peter K mmel <syntheticpp@gmx.net>
Peter McCurdy <pmccurdy@skeptopotamus>
Philip Blundell <philb@gnu.org>
Pl cido Revilla <unknown@unknown.domain>
Ralf Habacker <ralf.habacker@freenet.de>
Ray Strode <rstrode@redhat.com>

Richard A. Hankins <richard.a.hankins@gmail.com>
Richard Hughes <richard@hughsie.com>
Richard Hult <rhult@codefactory.se>
Richard Hult <richard@imendio.com>
Robert McQueen <robot101@debian.org>
Roberto Guido <bob4job@gmail.com>
Romain Pokrzywka <romain@kdab.com>
Ross Burton <ross@openedhand.com>
Ryan Lortie <desrt@desrt.ca>
Sascha Silbe <sascha-pgp@silbe.org>
Sascha Silbe <sascha-web-bugs.freedesktop.org@silbe.org>
Scott James Remnant <scott@netsplit.com>
Scott James Remnant <scott@ubuntu.com>
Sebastian Sauer <sebsauer@kdab.net>
Seth Nickell <seth@gnome.org>
Simon McVittie <simon.mcvittie@collabora.co.uk>
Sjoerd Simons <sjoerd@luon.net>
Sledz <sledz@MOBIL-400-586.intern.dresearch.de>
Steve Grubb <sgrubb@redhat.com>
Sven Herzberg <sven@imendio.com>
Thiago Macieira <thiago@kde.org>
Tim Dijkstra <tim@famdijkstra.org>
Tobias Mueller <fdo-bugs@cryptobitch.de>
Tomas Hoger <thoger@redhat.com>
Tomas Pelka <tpelka@redhat.com>
Tom Hughes <tom.hughes@palm.com>
Tor Lillqvist <tml@iki.fi>
Waldo Bastian <bastian@kde.org>
William Lachance <wrlach@gmail.com>
Will Thompson <will.thompson@collabora.co.uk>
Wulf C. Krueger <philantrop@exherbo.org>
Xan Lopez <xan@gnome.org>
Yaakov Selkowitz <yselkowitz@users.sourceforge.net>
Zack Rusin <zack@kde.org>

File = bad-boolean-array.message

a message with an invalid boolean array

VALID_HEADER includes a LENGTH Header and LENGTH Body
VALID_HEADER method_call

```
HEADER_FIELD INTERFACE
TYPE STRING
STRING 'org.freedesktop.Foo'
HEADER_FIELD MEMBER
TYPE STRING
STRING 'Bar'
HEADER_FIELD PATH
TYPE OBJECT_PATH
```

```
OBJECT_PATH '/foo'
```

```
ALIGN 8  
END_LENGTH Header
```

```
START_LENGTH Body  
TYPE ARRAY  
TYPE BOOLEAN  
ALIGN 4  
INT32 3  
BYTE 0  
BYTE 1  
BYTE 3  
END_LENGTH Body
```

```
File = bad-boolean.message
```

```
## a message with an invalid boolean value
```

```
## VALID_HEADER includes a LENGTH Header and LENGTH Body  
VALID_HEADER method_call
```

```
HEADER_FIELD INTERFACE  
TYPE STRING  
STRING 'org.freedesktop.Foo'  
HEADER_FIELD MEMBER  
TYPE STRING  
STRING 'Bar'  
HEADER_FIELD PATH  
TYPE OBJECT_PATH  
OBJECT_PATH '/foo'
```

```
ALIGN 8  
END_LENGTH Header
```

```
START_LENGTH Body  
TYPE BOOLEAN  
BYTE 3  
END_LENGTH Body
```

```
File = bad-endian.message
```

```
## message with invalid endianness tag
```

```
BYTE 'i'  
BYTE 1  
BYTE 0  
BYTE 0
```

```
LENGTH Header
LENGTH Body
```

```
HEADER_FIELD INTERFACE
TYPE STRING
STRING 'org.freedesktop.Foo'
HEADER_FIELD MEMBER
TYPE STRING
STRING 'Bar'
HEADER_FIELD PATH
TYPE OBJECT_PATH
OBJECT_PATH '/foo'
```

```
ALIGN 8
END_LENGTH Header
```

```
START_LENGTH Body
END_LENGTH Body
```

```
File = bad-header-field-alignment.message
```

```
## last field incorrectly aligned to 4 bytes
```

```
## VALID_HEADER includes a LENGTH Header and LENGTH Body
VALID_HEADER method_call
```

```
HEADER_FIELD INTERFACE
TYPE STRING
STRING 'org.freedesktop.Foo'
```

```
HEADER_FIELD MEMBER
TYPE STRING
STRING 'Bar'
```

```
HEADER_FIELD PATH
TYPE OBJECT_PATH
OBJECT_PATH '/foo'
```

```
HEADER_FIELD UNKNOWN
TYPE STRING
STRING 'a'
```

```
ALIGN 4
```

```
HEADER_FIELD UNKNOWN
TYPE ARRAY
TYPE BYTE
ALIGN 4
LENGTH ThisByteArray
```

```
START_LENGTH ThisByteArray
BYTE 1
BYTE 2
END_LENGTH ThisByteArray
```

```
ALIGN 8
END_LENGTH Header
START_LENGTH Body
END_LENGTH Body
```

File = badselinux-1.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include selinux_root_relative="jomoma">blah</include>
</busconfig>
```

File = badselinux-2.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include if_selinux_enabled="moo">blah</include>
</busconfig>
```

File = basic-1.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
```

```
<listen>unix:path=/foo/bar</listen>
<listen>tcp:port=1234</listen>
<includedir>basic.d</includedir>
<servicedir>/usr/share/foo</servicedir>
<include ignore_missing="yes">nonexistent.conf</include>
<policy context="default">
  <allow user="*" />
</policy>
```

```
<limit name="max_incoming_bytes">5000</limit>
<limit name="max_outgoing_bytes">5000</limit>
<limit name="max_message_size">300</limit>
<limit name="service_start_timeout">5000</limit>
<limit name="auth_timeout">6000</limit>
<limit name="max_completed_connections">50</limit>
<limit name="max_incomplete_connections">80</limit>
<limit name="max_connections_per_user">64</limit>
<limit name="max_pending_service_starts">64</limit>
<limit name="max_names_per_connection">256</limit>
```

```
</busconfig>
```

File = basic-2.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
<include>basic-1.conf</include>
</busconfig>
```

File = basic.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include ignore_missing="yes">nonexistent.conf</include>
  <policy context="default">
    <allow user="*" />
  </policy>
</busconfig>
```

File = basic.conf.~1~

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include ignore_missing="yes">nonexistent.conf</include>
  <policy context="default">
    <allow user="*" />
  </policy>
</busconfig>
```

File = basic.conf.~2~

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include ignore_missing="yes">nonexistent.conf</include>
  <policy context="default">
    <allow user="*" />
  </policy>
</busconfig>
```

File = basic.conf.~3~

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <user>mybususer</user>
  <listen>unix:path=/foo/bar</listen>
  <listen>tcp:port=1234</listen>
  <includedir>basic.d</includedir>
  <servicedir>/usr/share/foo</servicedir>
  <include ignore_missing="yes">nonexistent.conf</include>
  <policy context="default">
```



```
<allow user="*" />
</policy>

<limit name="max_incoming_bytes">5000</limit>
<limit name="max_outgoing_bytes">5000</limit>
<limit name="max_message_size">300</limit>
<limit name="service_start_timeout">5000</limit>
<limit name="auth_timeout">6000</limit>
<limit name="max_completed_connections">50</limit>
<limit name="max_incomplete_connections">80</limit>
<limit name="max_connections_per_user">64</limit>
<limit name="max_pending_service_starts">64</limit>
<limit name="max_names_per_connection">256</limit>

<selinux>
  <associate own="org.freedesktop.FrobnationaryMeasures"
    context="my_selinux_context_t" />
  <associate own="org.freedesktop.BlahBlahBlah"
    context="foo_t" />
</selinux>

</busconfig>
```

File = bit-hashes.sh1

```
# Configuration information for "SHA-1 Test"
# SHA tests are configured for BIT oriented implementations
H>SHS Type 1 Hashes<H
D>
DA39A3EE5E6B4B0D3255BFEF95601890AFD80709 ^
59C4526AA2CC59F9A5F56B5579BA7108E7CCB61A ^
6E42FB84067CFF056C43A49E484997AF23190879 ^
C63FBB9A87171A176E6E054890E29A8C5F125F6C ^
3109E33C1C4B9A0169D1599169D0E5A520A1E71C ^
9195E1E73CC68D7170F44BD1D83CB624BC87FA0B ^
64F7C374527278C0436DBC8DE5AABEC2BBF634BC ^
154B622EA426FB151B1FF1BE1CE871752B9EDEB4 ^
12BDD00FD4038756CBCF8ECDAD1B0CD862603CD8 ^
6700F93E1691E83735279E167F67AF61FEE9813B ^
84AF20A06799F366435BA20F99B9A10955B6D276 ^
C622FE003FD16220357F88FC6B9DE4789E77C321 ^
C8DEA5A98DADA1F9662D5C150618140A150ECA46 ^
FBB7127E77A39D856EC849D755C868CA7B9F11E9 ^
B7D8A7A39AB33ED8C3AFD8F76201BF0904149437 ^
154535460A12B9E100E251BB3A4D4DBCD80A309A ^
82FC6020C7A9B62F334F7156ADA608532DCA6A3B ^
9DD980B8073B32D21E9B2324F982024375C2951F ^
419F97A9899486DC53C58C761CEB564314FD47BD ^
A212E43EBFB31388F64C0C0B0B1CC7019704B744 ^
D8F1623C186A5615F58E846C9D164F35A8956E44 ^
```

AA32848DE1499365730A3650A3A7703FD404259A ^
79BDB01E55B61A8C1AFBF63D3CE2DCCAB8CF3DFB ^
23BF3164A448A727BA71457EBA9FCFABFBAC4F70 ^
BA26E846EBEF7434B63F5862FDA005A1CEB5656E ^
98622869A474EA85FE575CD979354248141A14A7 ^
5C9E7E6951D386205542DA0680E07E60EC61D64E ^
8EF07A02A7023AA0E6EE4BE1518627A521D1F9D5 ^
CDEA382A5DFDA1C633F8BBC77F291F40E8801CE3 ^
3C0A709FEF85DDBCC1B362AA162010461F31AEC ^
10ABD6B8D97E688AF5CDA7955CD3CC0850A4A75F ^
0EC53133F7FF955CA37D25491B592DFC0256BF23 ^
1A46A67FEED1604523C062AACC2455223B366D05 ^
FA3DD749E8F3D9029D9261F25157B3C0E2DF6834 ^
43DCAC2CF20E7CCB1DED6E379CC302F302655F3B ^
8BA25BDF331671CCB4B60961E4AF356224027862 ^
AC3A3EA0FB8727334A4478ED165128581BBC5BEA ^
CA4406A231EA2CA06CD1D8F3C77FEF9118B1501A ^
5B83465B8EED14871FB51D9E6CE499FB2E4B4209 ^
9445F521861946D4063E0CA3F41F580855A7454E ^
C2410140818962F95506B407AE8514F11F892AEC ^
F8F826FE0D93F415426ADB6956242116CBAB2426 ^
C803F7539AA036CBB31B735238120FB85180D8C8 ^
162EB5163692D9BC7B4529D214BBB932157361B6 ^
D4B0D876A1952ADB5D27A1A6BF71BCF22731A4C5 ^
CB5CD26AA39E45F248706604EDB9A2DB8EDC1BF6 ^
F1AC987F30FF0375F6115C0CCD6E22DDFD0FAFE7 ^
692E209D3A6A0228EB9823DCB738B16BDF97A4E9 ^
FB9E1E71F7645EEEEB3DDE66716DC44444CD657C ^
D51F071E360D96124DA49AD37E37DE17B564920D ^
0667AFCDBB82B4213447D0B22006DC4A1320CD21 ^
8FBFC65FEB2565F2FB8A9966B74702350C87CF40 ^
8A8BBCD6AF76E71067EB0AF54F90EE5AC2B5882F ^
8D831EF2344E336CC27281227185A34356B856A6 ^
DE7AA7424F92CBC50966FC4E6A577F98905998FC ^
1064EE6BEF704F3A618CB9C58F299B22A4D34293 ^
95F8CF7E864FEBBF77E46067063D08D5175FC508 ^
79381F230970F120D19B115BF864C794878E7797 ^
D82A30F49538561D5A7D1324ECAD321DD05E1202 ^
5926CEEAE43D29A7BE4C0EDCA626C43ABE00416A ^
563458C9F2C16C5AF96915FCD9306F1D9BA579C9 ^
C92B0BD604E45D7479AE19F641B2DB067E4CA774 ^
D69D013EE5C871D3AE3E388090BC1AAF9852A83C ^
7C341ECDEE7298605933465EA9ADBE7E044C8A79 ^
83EC687D22A13EDDCD470C52C45276A6E1554982 ^
6D27928F014269CEC4E8745EB73FA714BE2DC62E ^
75C0BF312C05231DAF7E9CD4CA33ACA78BCAA59A ^
015D799C602DD4E30D2C0104A57B1E214A423D66 ^
D2F43ECAD20F0B4C57DD07D3CECE562B6B72C7FB ^
27948CE5A56D57248D4DF683DC3FE023D2B9B43A ^
2D5A984BF32DE28242744406469E18B7A5178A97 ^
FFC5C4ECF0CC051320E071D14C7DCA4A4DDA305A ^
CED8A5487AD13AF0003129659DE2556183836C54 ^

599BB509863CC5D275416F6DB6C5D2BD827DC347 ^
3981DAC895DDB9294D933C1CE78828E3A558B152 ^
91EB715904123D326B23ACD684DFBD7203CABECB ^
5A9AC8C3705AF14DDE50E6CF2FE5B0C15C4BD282 ^
E2620DFB42B5ECA8F85F590A71DF696906AC41C4D ^
CBAADDA7B5DB389BF1775AE50040081A05A0EE48 ^
AAF80D079F3F980BF9D5366C5B1991A3E3D9FDCE ^
2A861F616604C526ACB157732DC431E81BC9A1F6 ^
FBDB22AA0B87718F464C0D1AA39695A1BDCE053F ^
4331BA331522D4F9CD6137858DB9AA16AC95316E ^
33F6BFE23A4C0E04B8BF642CD96437EF0055AAF7 ^
2046138F7899FDB2063CD84B84E821525172C03B ^
E4FDC812C5F7A37FC7A81005F1E78BFF59666887 ^
6477F931AC20B2FCE1C92171FA4DCF2D266B3DC5 ^
ED0DD516FA7FE2DB7AFCC3DBEED0879A0C10F1C2 ^
FCEDA0311C085368A5751BCAD8EAED2063C2D40A ^
23EB08B5100C8FBAF7B3049887A11CC531B6CD78 ^
A6C42B757DF078A69FFF6819A8ABC46E1B09F435 ^
0615EE4840B1F1CD628636D1B4F4495639DEDEBC ^
E2288110007AE6DBCCFC74B7735F4D94B0AA10A1 ^
159E01DC348EB067E90A63BC6FEFBF4C45A01F9A ^
A16286459CC989AE662C731F3D7D0CD487DCCB3B ^
8FDAFE2DE37215E142E27A65A881F80E0430BF49 ^
1BF978B58D98FC15731F4111E7C7A49F9CFD7C65 ^
C1F61CD4B4DB14AA60C6448C0398EEC72107826C ^
5CF0A7F6F46A730CD9103A54F1EC8A7549FA5788 ^
FDC104217940BE5A7AD1C02D128B31747C972A5C ^
DFF58BEF4FF418D26A35B55215CE419D3579A194 ^
3A2EFD444F4AA7A3DB9765E027CE5D82B51F6737 ^
EB616B6F97C5BFD02C1B9E4AFBA8A5151DD4167F ^
62F31F9A28D0587BD8253601A7EFB698B03205FA ^
E87645B2CB7DF66DA2771ABE0CEDA0F625FC8AFA ^
E801C3071CA6AEE278B222747BFA819C118A2DDD ^
EC70223BD963C122E4FCE2928934DDD2D7661CBC ^
82261D26377A5A5DC69E0F39D0BBF59A1B2B2248 ^
2E4DE781E4407AE536F70DC5A5DBEB178DBA2FC3 ^
AD3D958B884CAA28D5858DA103463C300862DEE6 ^
2A6A563C7FE33129637FABB4D40F5F0EE8B5C500 ^
7EA6003A8BEF4D5BFA6BFEBFCB6A5B3B1A3B7991 ^
6DDC620DB14882575D5BA1219284CF1C3068094F ^
91AD7E588EC829739C3632656DE1915AC23C8CF5 ^
311A491C1B5D8A26EA2A77A36B5D53FDABA1FB2A ^
9D4B752AA8FDAF92897B03C4D65B65EABBD062C ^
C569FF6F0592664F690B008EE7A12EDA5740F0C7 ^
1AE35E9CD3D0A8A165B2758E1DC4F9D1CD30FBF9 ^
819A83318700EA861CB69F6CD8ED4ECD78CC3CB7 ^
98099B550A82CE22EFAE48B5935F636D5814FDB4 ^
CFADB998167DB289A7C17BA94DF7EDA00B29085E ^
3F5A90A8F757936987938DFB95F10239F91AC264 ^
7402265908EEA5DA183490FEF2EA2E55C9FC2862 ^
EA6EF0F7050EDA6B78F35597E85B7A85AE0097E4 ^
73F2780548F7C79E15945D7EE2A3E03B30DF35CA ^

8EC3C93C7307E14E9C65F26DD847CEB195DE277A ^
CFBC571A35B82FB0B7CB59753C4E8DE36A751195 ^
3C4802A956CF93A148B9034FB8F041ECD8DD4532 ^
5DA269C1D5CCEE3C7A66E31BD9F5E0B83BA06CE1 ^
706D90C00800DB9790AF66788D1B3DC81E3FF7A1 ^
59AC856447FDD27421C52EA66E465C64C5033722 ^
EACA9E82D043F2B6E12E64CF03D82ED92FD6A64D ^
8160755BA09C74F79DE8F9496BF0C9248DB7D45E ^
8CE4390C185BDFC8BC9C4D3BEB1AB757C44F6434 ^
1A1ADB628EB7B9CAA821AE9DE417E09A42A97113 ^
97C4F92A46EF3F9169A977E1643C10745ECDAA4E ^
408E53C03D10E2B32222F2B54D118AC4D65A9386 ^
710628E811C207784E3255DA8432CE6E14B61FE7 ^
4F6DB622778F4B1B6203BD239312D86589F1B261 ^
888AD8FC1CA5B893294FDB8FC94AB7762018C1F8 ^
AE36C2F54223BC082DDA0D528B70B2AEA190779C ^
4BD2BF914223DCD0CBBCC444323FBF0E487F27B6 ^
7526D5DF7B2ECFAC7BFF62DD7A4B5F38463305F8 ^
0ED7C87B68F50813AAD5E50159CBE0BBF9BF05D2 ^
4D5CD4E40A6668893EEA15F025ECB8187475DF4F ^
D38446E842C0A94F175B772BDC707CD2EE79D098 ^
DDA868BC2903C55CF273F48E9B92C09D90CD0179 ^
0841268B95ECFAAB9FE3787BA7261227A1F23294 ^
D3DDF20B4C0DB668421DE1F242B65A8C83FA867B ^
112F3CF7DACA7527819D16704EB41302A4EE52AA ^
96072A3EA4463A156B3BD9EAEC16ADB0905160C9 ^
715F195F09E3EAD3D70AE51DEF1C9709500C0A6A ^
8A112CA82CFEB5F3001B7A00591462F4120DDC02 ^
50FC91E915270D1202F6FC8792BE16ECB73CE405 ^
907736EC45BE7BEDA1B154EA282C51B5795991F8 ^
05D49462991278720EB44BEA656AB4442C8845C3 ^
6FD56D08A2D4630F72AC97E3C94A9575F3046729 ^
DD1883376CF713C8D7AEBFEC60D51FF8E2A806E8 ^
E0457FE00F9DEEE958DD6AD948EDF9CD9412F73A ^
BA6538E91B276504C84E213B325B4A7703E924B4 ^
C20A9B848503650DE9463B9992EB977422BB53DF ^
BA4A565E5C0F9FAC89D55A2B784F291C7BAD4255 ^
6510E09AAD8288AB81CA54DD5656DF57D0EABBE6 ^
33996C9C5AD04883BCEB9771A5C4A272F88550B1 ^
7B96F3C1272AD765DBDC35A59243F6A8F75FA351 ^
5AA76B16EA4DFB267D88FF7DBE4F055184A56E56 ^
33B9CCEB1A6EDC8C91F6034245B2E3AAC898C678 ^
701A32FC0B1163BC55DB76FA46F69B574B3803A0 ^
5D6F44A4B5175BA873BFA3369DC8BFB0AD1AC26C ^
0727F12D8E554EDCB435AE3BD6425B197DF6EE2E ^
6E1F46DC4FAD671759AC969998694BD79326246E ^
85233F9CFD0780BD60240A22E037C14EB8286C33 ^
B66CC7332341FA9261D96362F0BBD7B676F95E02 ^
782A78F2FC32E0B627702C541341ED28B009BB6F ^
6B7935265C529880DA5B166EDD660BAE14A770A9 ^
49C298B1C10D380BACE26E907C213C90052D7523 ^
49CE4D4FF9FFF6724DA032FF33F8C5A7792D24E2 ^

CD2C4F19C12B086C071BB6B20E8E8210AFDD2D50 ^
39710F77BE564C417C80CB6E4B87BE72FA805A40 ^
F74A6A45F5B2979B5E773007B90804F322F13AD8 ^
6FD6A34339047CF13F777C4B7D9E287F499FB19C ^
FDB04F04D3B364A6953C341E343E310F2383A655 ^
981E8717538CE98998E950EAD28596B527736841 ^
E099987DB0CC61CF607C9852336CDAF478DEE33F ^
E9C42F25A3E3542D1510DD09BF516AD809D42702 ^
F7E580FB21C725CBEA563B193896862B9BEF7F24 ^
D778A3287F68F47B0AA421410292D47805997A94 ^
534DFE759735723649EDC89BCB063C0CCFEEC34F ^
9381FB5BD26677A7760FC0288532755062986DF1 ^
27E97EE829820399595DA0A0047B74D3C4E0DBF1 ^
D38450071D1B079A9FD518D2F317C83C33094F43 ^
0221B8E4D2FA66D2EBE1B04E679662CE41405E48 ^
B9BC4FDC4A5CE00EFB911C3062D6F33304701C28 ^
461B4206618B766C89E6D9AB2F8DB07903860234 ^
122A9306B102F2C50B7364E1D24FECB0662E2958 ^
61E91F5D167066EC904134AB1C4F9D116C9D3E80 ^
6B7558BD8F4D3C5914A9AB5D7418F7DB77583B55 ^
CE48EA94843854555DC48D24D32405DAB9DBA612 ^
5A722AAC576F4078C001D06DA91C28131A4F92C5 ^
9B8FE0F1E6573CC6832BA69DEC9F6E646228A374 ^
F98429CA2799E5CAAEDDFB168B782A54384EF0A8 ^
142877754AB95F57AA541CB978BA400E812AECC9 ^
6D84D053CCD9E68CAC5780892409F61B39DDFD48 ^
B58CC9CE0AE3DAAA02E3C040100D4B568D29E0FA ^
17FB18FA32A8237D32CF83A9664B1B2E40DC5B11 ^
C7F65E54E608CEF1C7A120BE8A29FA6C94B680D2 ^
9C46329BA34077394B439C5476B91D9DF019BA76 ^
3612B381B80A1DF6DA75AB2656BD5BA8DE6996B5 ^
EDD9BCA1C431200D3234C099791C81D8605091AB ^
C1951DDECB9796C0D39ED9AFCCBB65200AEAB0CB ^
55E4F7846D9B90D8B31EAA5B9371D395590B6528 ^
5D40B432ADBDF3A01355FFFDDBA47586B3DC7C52 ^
AF5C675D2249F4CD20E70DC9E18B88D3ACFBD6BF ^
B0EB6796A21876FAAF50FCDBA2EB121C4B742166 ^
3C577C70B76810DDF9AD579910347312B881C30F ^
0C16897D20C5F352FA6110C70F9A8C33CA35CA21 ^
DBE0B4341526060C5B37576F6B972A1EB8954059 ^
1567B10E33FBBE20033F5A52EB810348ED154F19 ^
C1E47289412D3ACC208620A1EA676CE2815DD4FB ^
ABC227EFFDCE3758061C77971860247E850382AD ^
B29EACEC903BB8AF7CC1A63516163640FFE62822 ^
5C8976A1A4FCC6B6FAB5EE631E3CDC561ED5FB10 ^
1A1EA5107CC46BE909DF8165C3EFEE8676161991 ^
268E86BA3A81FADDB20F74DD44C5BC71A98C3D08 ^
05057E5E7934AC9A9141D39B8D4F5EA6AA14F74F ^
C03EA72C4D68945726DF63803564E434426302BA ^
54AF20BFBFEAD6BB2DD68E65D39D427DF7E17FD9 ^
FDE938DB0843A844DC0F6601497A39ACF167D6DD ^
ED0292D0BD869B66E80014824A3D7886CD6D3590 ^

749B69964F5C0D9D02BF5CF903B2E56D09CF1B9C ^
6B143D30366C45F159B83A8D1AE8AAEC88E44136 ^
01169D65E71014946D0BF64BF439068978AA5B3A ^
4C36E0A8C19C0AA6C80DF0611BA3404C1025E9CC ^
A4DA9EEC2F682AF5B2EE7CC8525EB07DF66E5945 ^
F1349AD0124D11613CFA46D498671484E5F37790 ^
6A8E82D2D27D3EDD6E8B119F746761B8FE83182A ^
52E7C7C4AE02659953079DAE20F442353EE2BB29 ^
49C8D822724274B85DE62D9DF586FE880AF396E6 ^
57866C0BB976B4F8AAA7ACF3C9A0F700E73216DA ^
541A675DD37C5D6F2CF242163A5CF6B728DF9995 ^
55B26D9344D4D644CBB69C4789473543262E8B91 ^
3A8F3F0A96297020EEA583E9A280D53E0248DE26 ^
D1FF7A0CD0B69A76B5B80D2BB5736B8440DA7F7B ^
53E715FA39D3732F38FB4748D46409D998D67973 ^
377781ABB4AB4E974B9C1670867E74F87AB4950A ^
8E7DA96971265ACC0FF7C381127C966DF9F9F13A ^
28038713A33ACAAC730FB4DDC3A707886E598DD5 ^
5F6BD44D07A1C811D5B75661A61BB69A81444EE2 ^
D465E6C767BD02D457D601765DD0FE5521BA2AC7 ^
487D9E02912004E961968F387C21A65A0E653AE7 ^
1F54DF00052B3740301C43420CC2CB85B2770095 ^
7C5BB678083868EB817E616BD09421B910021649 ^
4D31526E4E2FE3BD6DD4B3F59DC2EE2A52A9FB42 ^
669F9FA9D71FB92D51ADE15C70742FB31B013901 ^
38D1E74FE26AA43D45DABC160DEFF113B7740211 ^
3724EF50CDEAEB856123BEBB190F38834461664D ^
E2634112D2977A179D4C9AF35F220D3A626AA04C ^
F14CF8133FF66A3E20D25E5DCFF70605F6B8A0BC ^
809E04A38340D010B78B5287C8FDD20C391FB61B ^
E5349D4D6EC31E2CF41001DFC440ADFF6D25B4F7 ^
139F25FDC344EC2456F6E4DD9938F65693F0CB2D ^
7C628B0954D4340B0851C89F1106589C5C6465C6 ^
E65E65D7050258B6B6B5B35866840B5ACE0DBF0D ^
77D33B87E0C13529D3AAC4D6606213EFE81FF398 ^
5BA456BA49ABE50BBCB772FAACB71D4F3AECBD45 ^
801A21F7AB73585AE3531058DB9512BD69BA5CB1 ^
192AC4D5210EC863604FC1BA388AEA8455D4175D ^
FD9F103EDFC4CC9F9C424266EE54A79C1C987F9E ^
361AC46D96A69DD6E416D4563D7C69D3B487EFC1 ^
3A7629045C0193FE08928F7D41B801827593116B ^
545B2F5FD015F81BDE1DAFAFCA184D25590560C9 ^
FEF2C50DA33621206450E21D92951B25385991A3 ^
913A417AC1ECE08952A60E0035F70993F161EF05 ^
4FFC95481F6F625B1B8E67B133F11619CE5C1460 ^
5D6D0765B156FE028780331B6A53F84DFB130F0A ^
2C5EC89F1018D34AB11E0CB6FDC812F82470B168 ^
34D62CDFF955233FE7A5308925C24CBCDC0579B8 ^
1148994D30F80C5689D7BB19CFBE33546872B4FC ^
074FCA22A8A63F446C8D196612F7C812C22CCD8B ^
228FE29FD13928D8FF0B850E50DB41DB406EFCBB ^
EFF7BBF8F2F7DF36AD51EF3D748AAB8D8DAB7B7D ^

C3AEA6AA17971C0A143ACE54A8CF515F9EEB0BD6 ^
4C3A980E40602D4621C2AF1BEB1CAE27DF19AD32 ^
E5CE2A41C72229D584E40FBA67C697EFA08F451D ^
64A5368264959C73FB0A12F85D4FA16F8E25FF08 ^
8FB770AEC165A93A6D99A0DC52F38444E0313073 ^
BCFF3C5E5C56876A3DD8C4C90D4E819B67D71394 ^
0EF20523757E25611C49C1F259D94381B08FB90A ^
5D9F05DAB91C46ADDCE4B163C6FCDFB5230BDFFA ^
11B3BDC5F3BE09148DA07CD42EC9C28C718D6A5B ^
1EAD76EF18840B3176296FA57791D832B7E4035C ^
AF8EBCBBD5B8CFC203FBD8644B154445690B02E ^
0A156DAD35A10AA7E41C5CB3B3043A051B7180B6 ^
9B3E5F797BA246E5DE660B9A36E52C204BACE822 ^
78780359CD24A2B7157FD6D55CDFA6F9DE149FF5 ^
F2DADAB4275633F7ECDF971D824A31B1C8D6643 ^
EA68BA8F16D8368E6673A5C5AEE86BA5C36F2D97 ^
788AA9451196E2BDD031E747B098D73BE113C093 ^
18FE259B3573F52F23DAD33F05D9C91942B516DD ^
0E4787F46AE8D5D7CC9589B76D00D60983079D75 ^
148E4FDF031CAFB726099DCD1EB591BC2F24D179 ^
3981DF8658AC2FA5C3151A424262324D41940DBA ^
3C50A3800636EF877D89059F66CD697A9DF44BD1 ^
D9BAB0471E2DB97FD5F3A42884DFC708768F62F9 ^
52DD1063326E2550B6EF40D5EE4E1AD67C7BB6BA ^
13CA27827127C698810AECA158DCAA7894F0C912 ^
086BDB17EC1B8CAC68030A56FBD4C741D4F95DBA ^
F7F389BC5ED225BD6399CC43EF22AFD77E895D06 ^
67F173AD3A24C26E95E2EA8B37FC0259B1DE955C ^
A7F348A3B6F67CCF7C7C69ED065B3FBFA832A8A2 ^
ECECBB25354468FB62AC87D5CEB688E4A671DC42 ^
62330BC400108DFAFF298EFB76BDC28CDC153AC1 ^
BB0ECF9CB3CE448B4EA9C57CF5F1F997F81082B4 ^
12C9741672CE8DBA14CC66AF3338B61E234F9E68 ^
DB4CB5961A2FF6964CEDECAED3DA03D140112EA4 ^
45E4D2BE3D9729BE9F69ADB245A5A9CC75064534 ^
E35253550C84BEFC26C2E97F0C236CFF1D85C36F ^
7D6D950A168107702CE3B6BCEF9298F9C8FC8F3C ^
812F459AB5887364D1F2A9A00CBEE5FC352C39A2 ^
7EE27F33C59B68C34604416D38606BD3F5CB6B06 ^
D57E2A4F15DF5C50F2485909EE3149735C6DFB84 ^
3D12D88B61EF0972808D8CAD41D8A16AA7ECCC87 ^
CF9A7CEA9B105ADAB6BBA3CFB6D090B76603F7CE ^
1D43CA164CAD0B616BE5763D7F5EA88355953F4B ^
3600F085A53FB0115C26DA90D1180FD34EEB6301 ^
E9F9C088C30A37E6CDF067F724BE6C4786F27C9E ^
2BB86B47EA34400D434F8F4F94830CBDA9D4E573 ^
3255A7D9A7A0B56D0D9C665EC13F7A8C8F05DA7B ^
453597500A7D5BEC8CFB950926A2BFC0F92E149B ^
A15EC452476870AC459B942F6A6B73822A07899C ^
5B1821BFAF917BAB3DB1284A1D1FD40EA011BCBC ^
8DDC25C7CA330A18CF39827B7CBEC00B874AF193 ^
283832F45BB5E69649FC10DD607BAF8BAD61783E ^

F39426F9C0474596B0251F0CBB5F92385D862D24 ^
714E32DD5BB8FCC75B1013E6F1F56F96D97B36F2 ^
988AC4592D50113B13F6EDB5C113A22E4D0F518E ^
93BF1463788B7F4A4A06AD4C17482DC260797A3F ^
5532535ACC8AC5668977B2A2F322B23AEB443DB2 ^
85CC7AF0929F206FA44FCAACDC24C4E78345BD80 ^
392196E3F2F4074D1D485A52119012D8C05740B6 ^
3BA88B91F1EA18E712292301FBBC375E9B48BFC6 ^
3DAAD1DAB1795F2C1C8C5CF5305B549ABFB85398 ^
6EC93AFA3AB617708276E99066FDC331C84F028D ^
CDA219303B91AF3C3FBD0B926758914984AD7C8D ^
599CE24745EB5408554077354BD10A65F6BC3037 ^
0798C1B53F949D643F7BD66C194715E7CAFC6E19 ^
A17230F164E98AB83AC1141730B23389F0EB473D ^
2C2B95173F1B06165E17EAE36A53C7AB641267C3 ^
A7F408EF449DC92961617D4C51B55AE4E4536E97 ^
2266C5133B6D6B08A09897AEFA355C768F718C6C ^
343F10EE6EB34DF005E749D1C4A98F64537272B1 ^
90EEB76B126E632308F8655FD49C52000027AA48 ^
C7EA3C7D4AA8DCBE9580C4434075017934268419 ^
455D4E7C988356A56588C40313D28E425AE219F9 ^
5C0F3090992138D34B4E33029CF83A64ED8AB03D ^
CB3C5288839D6FE79EB2E50D3AC34BBC4A942E46 ^
E605842205079D83BDD489DCF6F8A2D1EE074C60 ^
ABCBEC56EA20B6B369C85A931B2201FECB826F4A ^
480C1CB97C3419DA610D9783E6047E6C9A6101A4 ^
874818464DA1753D3AEDDB7D3B46E2D04614D7E6 ^
C95B65E3A37ED4A2FE2B27A2D52BF1BFD307ED20 ^
5E188A4F625C05482BF161C45F5094844CE70461 ^
6B2EBCFE74E0BF32480D4BCF9A5698A527AEBEAD ^
ABD3E28A2A528B7963EA8D8896651A17337304D9 ^
B873FDE2C312C2279EC8950EC2C2E9DF990F006F ^
4F1A18714689BDB766AF686623BEF08DAFD1F5 ^
BA886793A89E7EC62692A8690164BF072EAF7AA4 ^
A15C8DDAA0448DE7B361552FC51AF0D72C0737FA ^
6FB516CBB4CCE9681C44EFABABC7A26C79C79CB5 ^
C6DA89D3F7EC4D2FCDB36432042A78EAC9F04DC6 ^
2A94A9023F7D54DB26C8C2A6A4C951126712C2E5 ^
B0F115B8BF0A1464A378420A1F1F574AA1869FC8 ^
A3C1794F58FFDC361BCAD4779B5DEF48E02ABD1A ^
2DBA5D80C4B17167E715A8012876095577030D5C ^
062C3CB7A059766125B368AD6C19BE8FAA981950 ^
ECEA96EB623919DF4E22FB106616EF1FB2ACF5AF ^
C0CD5E555861FFBCB2FF4045A46CE30AE6CA1758 ^
A0150D931980E1D82A21928837F7E9E198E5D1B9 ^
BF877B202812184CA74DE61FE34DB95BA2AC5D0B ^
C863A0D2232E901920752F2AF92C201FC73609B5 ^
C54E676209D81CFED2E7ECEA6F7E35A5459A58AE ^
CBF376E8571F81792AAC49BA7E1D7F41713ACDCE ^
9CD248C07EEF6303C71901AC484F30788D9269F9 ^
AEA41C5B2260E0DC57094C5E1F80A70B85E79E77 ^
B1016F26E32624C5328566B243329A2F75A36D06 ^

3EDE118F7E2764CC49FBC3873508D38A2C6A4592 ^
03DF0C882FEED72F81B303414178976ABD089EEF ^
2A8E36B6A545D1ABE7B02D6DDD782AE032F7ED41 ^
D90588E606FBDF3CC6030756DF92FB3729441E6F ^
BCC413133B6D84621B482A7D2163EA955AC733BF ^
BE1F36D34DEC54306EF20D8326FF603A716F4331 ^
A2BAE41EA01E8AE9D59F5E8E663D22C9F414B0F6 ^
D2295FE26206DE9ED169FEA487B198E7CA0DE78B ^
06B13130DD10C94A24CB50DE6A612867F85E17FA ^
2CF4C4F2F270AF63D7035903231CF7E7180E66E5 ^
0A9EE3293A2C9B2ACC2F9B99BF17AAC618347857 ^
C338F14C156D8777ECFC36ADF1850B64EE9E6179 ^
E9BC6DA35AEB1CC399733CD0AD3B22E4E2C83E92 ^
7A42F820210158AD33935CB77D78C28F9FE7A940 ^
ECDEFA7817A436E43C3DFA35069F87948181A58A ^
761EBF336F1229BF80702C9C7798DAC37ACF3086 ^
1975B716880B49BA0863136736942E688C9EEC7C ^
A4232C3227DBB91581B86FABB1C5AB0A29801DFC ^
49947AF1D9488F146BE4C77782CB2FCF6EB13FC9 ^
A3402AE17DB9E6EC899B0E15973073CF9404A2DF ^
DCAFC2DC5BAB38629DFA0DE1454C300E5B9A2D66 ^
A7F915F745E30E4CF6AFFABF214AB6D950C66FAE ^
764515277D4AF73A14FAE2D9CFA4F9BAF306CCB5 ^
B021EB08A436B02658EAA7BA3C88D49F1219C035 ^
B97FE9E1C5918FEFBEDD986B757CCF4EC4AFA75D ^
6917353596BFAF1A5A06119ECDABB37CB017D71E ^
FCB4BCEBEBBAB66142AD2A5B35AB5DBF78C6A463 ^
BCA454B69F4486960C633D2C4453FD01D6758BCE ^
5CD25E62AD97AA5DB3A9CEE99C58439DFBF6073B ^
E8405E583D31B3B8E8B365087F2AB391753324A4 ^
39E32AE405420DEB07275224817B57187079886F ^
63AC20AFAEDAFBFC018755F4CEB13A042BDEFDFE ^
27245C74B61D19EA28FED81E59AA48EFC22946C9 ^
7254C8F5A2DF0E747760E2EB353C4AB24E253602 ^
32DD49B796ABCA76356AA9AB07D669CB7BC7F899 ^
B83F3FBE6DB38223253344E69C399C474E4C64A5 ^
B3E0EB484AE5B10B017F509730F350481A60F634 ^
A99FC247C177D66CBC6C21C10C21E10FB2F33EF9 ^
A7D033A946AA0315CD2E4B7025164B77E5979CBE ^
F2DE5CD912D5CFE727B353ADECB6B7FD11E80189 ^
111BA90582E17CB5E1BB70AB72C99FCF647876A0 ^
9AF7F1A742235C0E396F6D1319BD1104518F8475 ^
923B3E5246FB50FB5448742B8EA8CF4920E70702 ^
CBFC4563334D104E4DF24443A295F4E9A7D7D4AA ^
48F4BEDE76C6DDE9B72A9FED50D0BDBD0DF7E7AE ^
2A21E7828C064DB0D684561622802C86A504DD37 ^
2D599572D6E57A437A2A845546FBEF9F2FA11AD8 ^
CB4CFD92FFEFFDDFD87A4786D60E28C7622C71DF ^
FA6DB9D1EB92461454E7DADC36C9CE7A1F30C63E ^
98C264B7CDF774C7D82F4DA5E1160AE4370665CB ^
D503D3C5CAF7A2D124A7F9F0E7D8D0F5E29B6F19 ^
AEFF5AF0023868EF6E3AD49F2C30D65E963B6308 ^

73B1464B9D05509BE72E83C819F9D618B3121336 ^
791774666D2629D4304BC81138B4D8A7599E77D6 ^
362A2D419BAF263F1E77C5689F611AE301C044F2 ^
B4D4796D1BC1851DDB75F5D3BA3089D1D0E0A15D ^
B7C4FD404A73C0AE43FABAEB471FDB6F9B24A27C ^
B9DC118349DDAB37DD82B65E092277C1E2F20C5F ^
144782913DF2FA8F76D210A7A321F362FA5DA779 ^
29D64351A178E52EDDA9A4B2F5279B70F3FEF39C ^
31098A70FB95128BECC2C729CE5A2A4BCA121845 ^
9C05D6B171C60C37822CDBD7E8C1C57286E27461 ^
6FF7C4960E35463A8FDB77844A29DDAD1FF04F53 ^
2E08B9088646737620F1B448E81EA5E42C40C595 ^
AF2342D8F4BF12F9E524E44BCD42066F84620727 ^
543A1A99B497DBB7776403F3252A9B2E11A0610A ^
A2A46BEF09D21A9BA32AAE4FCF7A846D128F0358 ^
81960A20A755F84B3623D1E2A395EDB12DFDE73F ^
B016C90B350A5A9AD4C1F3AFE49E3EB21AAE8B7A ^
41D03A8D79E8BAAE116C2F5573D869D97D7F6E0B ^
831237F5BDAC37C58BA31F6131982AFE680BAC1F ^
2E8ACF5FDE18542BDC2D0B95D3AED7FC48B5993B ^
171763D880D45885EA61F5ED04BB51B79C6A2724 ^
405AD282A2141C38D3D731F9DF630667064B9ED0 ^
07F2ECB944D01DDF9C68AA68762BF07F385E7E6D ^
EB5FE8B813E4DD7BF852FB32466025F7D4E79DD9 ^
52DE090DB027385F17C9FD41F61272525FE108B1 ^
103E352A261134E1154ED3C312D890646FED38AE ^
65C73969E35E8F307B518ABBDE85654CAD13D0C7 ^
FB380C0099EADED9ED8DD48D70E15C7A26E2F417 ^
EF9E122E0B8F968EF262E1A98BBFED7B461846B9 ^
771A162BDE3CC4C07BE7A1DD25470286F133020E ^
5F892353DC5C65A1D8C9510ACEC43DD96F432742 ^
4E658B8743D510221B968F2C78C72A588A5DF6D6 ^
5D5295ACEC24F8A52ECC5B174E4884732BD216E2 ^
7791BD0E1C4136D97C46CCD66D36E98443C4182A ^
515419AE59BE4E824E575E497F361C3382C21783 ^
7669D0D083B3A0A7B631EE4871AD2A2AF8C512F9 ^
90A1A9BD93501D9AFE5040DDEE395F94E5594EE4 ^
B8E48F79618CDE951296F1D28B87653C3CDA6FDB ^
FE94D5373EF009ED2C06F632C739717791EB6BB6 ^
B5C3871DA2466E4635DD3BCD085A2B8E222CE8F4 ^
43032D022D24B65394F7A707C8441C2A7F3CD980 ^
87C13448017DB3B6C553250C59DF4D3D57676788 ^
F38FCA4589543051935D4F1525710AB366161F22 ^
921E3C3DE773F987452CAE28D3C10DF9804632DB ^
32308D5BCA33E694071B8C17470C6469EDC1DCCE ^
AC5258297BA22FE60B464DEE22349C70FDFAB219 ^
599322BC214D7248216F0C6ECF00CC34C792C5FB ^
99EBA91FAFC27288B3FF32DE55BF17BC6D76B16C ^
899DF7C21779CC0F54EDCD441051CACBD2B06941 ^
E5E0BA18BF3C99062BA444847876C5CA2DB8F9C4 ^
94B8B90566B2E3BDDA8FF8D37BDD9F95CF9BE0B2 ^
86422BAD65C5E0B22F02B9D1510B9D24F4227AC9 ^

427694B772A08400381D472854FA5FD4D4CC1D62 ^
3732DA8EBA84CB4591DD4A87BAE958635C85DD71 ^
D41C28C028E42AD1DF5057C97BB953B03AE8F6AE ^
4C44286F5979D44AB4CA5E3776F30CA04CE539EC ^
E695D9B049A1497A86765195BA7884E8869142DE ^
2A5DE9C5489375B8EFDB09CA29D32FDADAA07402 ^
012356EB46A587763ECD1EC9F3114B8B8B7987B7 ^
6B803DF07E5A47FBBE412149D8258B526D2BE96B ^
14A1090C3B382482AAD5EF49B30D842317792745 ^
C2A576DE385017FA84855388EE6CB1F51F43A6C4 ^
5746B8C88CE540313A0429E64841ECD0AD569299 ^
298EFB87E9C09299DEC17171902682D919899C9D ^
55AEB7930CD6202CD52E6513ED4C35824B574198 ^
ECDAA8404F88334E9C2C6FCC9209795581654C25 ^
061FFFCE8F69BE0D0C9B63E30BB065C6CC73DF33 ^
229F3A8BD2F6FFB5CE2DB330ACFBC4C4D7796D7F ^
0BABE60803248059F3B05E06E273B5F41A619C9A ^
5F496A5AAA1FC38B76069BF9F08C63777EE7530C ^
91FC8994FAA03E9A2D9D565F04478C6DF035AFE3 ^
A68552B9049E192BBF98E82E2A69E405FFC2D93F ^
7910393D1D9D8B876EC89FE9100320D371789C23 ^
9B923F97F2D2B057950DC3C0E89981F22F96EC2F ^
1EAD9C237CF3B1CB95E7ED79EF2E5842E9263447 ^
F51E5C202D698C75F8F0C2926081D4C28F641812 ^
8842D9CD97868196B31F34C02BA63CE34493DCA2 ^
26C8977BC36C3298582CE6703F62DFE4488E0DDE ^
955BEF358ACA4D44E33F31051A3B92D96BA669E3 ^
68D841B29DA285DA53A31E823462E794B3126914 ^
BB52C1806605DD37113035390D97036EFB3FE2BF ^
BC55772BC9B6066112512E1A8B633411B9508FA2 ^
BDD2CA4B43157C04CE7043F57B7E2610D3E175D9 ^
8E493F25E110F6244FBD7585CF18F285933950EC ^
650242DF99E985DE23A5CFC25C8CA72C80D7F23E ^
56A061FE49B428798C329336D4B184BFE69C4102 ^
59FF817CEE946CB948056227607A119F4C7DDA89 ^
BD4F03F1878CEA86E2AD981C6A80701E0AD30B4A ^
A635F528CC4028877184E8918CF73E1AEE6F777F ^
15DCAE87F7E40CA577C61C2AA5B6D90F0CBE6DCE ^
D67BD85A353CEBB99ADE4CD86B54673DD6D64360 ^
1619DE7229032800CDF071830CBCC3208B1EDB09 ^
09DC18BEE0C2DD9A2C7189AA6D3DEB8B7CC41CE5 ^
4165ED470D85FEDD9593577A9101306C31825E00 ^
C52942DABF4F9D72B31B2B3291705EF5201B3343 ^
88296035BE79F5281B91C0F1006E57C28671CBF3 ^
82593804880F168B9FF50D89CD149E9F263EB557 ^
52E4F42FC2EC2EB7C0A728D2251A14A6C5EA5A00 ^
C91B243CBA2031B29F365BA4451FEFD6A902298A ^
90F6F5A65E6BC9430A34C06631208AAE8B7555F5 ^
7535D77C8809B4EA21678BF9686791418FC71931 ^
027070AC6B5D77BFA0398B8AD556A23AB8E99EBF ^
B48335F6FB6EA27CE2156A1D3E00E42D3E3A8489 ^
B67303D71BF125F2441BFC599106177A32E43BAA ^

46E98D49BDC5A4604F62AAF76E964838291BFAF4 ^
F1ACC562D96999493EEBD2531A289264C3E1BEF6 ^
0B261712C3F4D0D74EBDDD052E1A058BCFF412D0 ^
50446AA7A1A81A97B1DD524AD0DB4D09971D383C ^
E3A51B6AA5AA586803EF2183DECA0CB5CA869B11 ^
C81B52FD60ABEA8CD5E3184E206F693211F008E4 ^
197270C0DF693622DFE871857A7C87B7BE2E439F ^
43453DF7620D71CAD56DEF9CEE616AD9E7C73EBA ^
3E3C6EE1141AB3D7EA5215ACEF4E876EFA89D03B ^
5773C28260C2B2D325883AE73B04D28BFD2E8884 ^
EB6F26A0E774022773CD86E1C8DFEA461CE6F8F8 ^
6C26410B11A0CFC97849F6E43E77B8E48D5C5363 ^
F0236CEF2298ED3D86A95E9F5247C17499A772DD ^
704B5496CE0AFF4289F3D915230F69ADC7A1ED7B ^
B0D4FF6B50F09FBBA19F76693AC9CA87768E2758 ^
10DE4F687577CCAD9E16AEDC9958021D90DE8726 ^
785741949059E8508DB8D132A9B053EB4277119A ^
CB041BAF13AD22778A1C73A8AAAB7458B664B1C5 ^
B4B0ADDBEFA078AA497045629840FD965BEEC33D ^
68F0E43B7DF89F484F3CBD6984C95168142E4A82 ^
D042BE14A23D0CA1A0466179C119F2A959E1E118 ^
DD23CC2824A881FDB6566AE6A57B1A2D1624E895 ^
15B75E7E13B374039B8C9FB4A1122BB8A22FAD6F ^
849F0D379A2775CCB5EBB0A88839516F30A665FB ^
8B4125FF2EBFA9459075459C1EE149D68C4F98F6 ^
1D632D3EEC63D18243FF48702AB894EA05A1AED3 ^
F2C2E1082004EDCBE3DF96C25D8860C0F5B432BD ^
4B96C3C284E9E12C3D2F18AB8C8FC6776F30C6DB ^
F40DC8B94D9FE9910117BCD66F3FFB22F6977447 ^
D65D126DF604DE306AEB41A641DE0B214449D69C ^
90FBEBFF058E825EB940122B6D8B7492FC574B32A ^
AE4414AD47216DA0AFD9AD1E22C882F665429C54 ^
5F9D97E91B7E4810AE765A09803ABFA50B6F64E4 ^
A2D4E6AF7AFD87E303E867D0E03E1945A0F0B72C ^
14ABF713982D4DC7C1CCFCE7A48680576082DCAC ^
BEB8F77A7D211D9451B698D65FBC0B5E38D26086 ^
B35F660629962D5F1F8529203999FCA0169F2675 ^
07AC5B04166E739A9FB87FCCA74112AD93A08128 ^
2D8272C9D620A730561EDFDAF2A129666CEBB5C9 ^
C25C1971EAB264F44CEE1F44614579EE6D8C7858 ^
76F8234E2F0A533E810545F79AE28A7C46056AF3 ^
35025EA09EA111E299814FBC958977DDBC3A2387 ^
0A2EF548A38D84F6A1CBB2AE01D48D296D6CD8FD ^
6F546DD1623E92DDA185BDA01618DF7E7E7B82F6 ^
A0C1C2AED8C3F436DE200E2A3F4C2A9D2DA8DD7D ^
8DC10195B967FC7FE57B338494C2C42FB43599D5 ^
0D9192B27851CAA766D90A5376CE255B2FDE906A ^
7799450D003824356EDFEE1B408AA59E830175A8 ^
2EA6F3188D3010EDF4EBB55E009CC4D99296F578 ^
2A73ACB83FB8E25D65FBC1B8085BE67342F5858E ^
67F9928A27118ECC38EE231E05977A694AD0F7AA ^
27E2078CC9FDDF5C5DD6CE629F3EF071B2107936 ^

F50540B4DEE071C64D861872F1E0BC4C7DE75B52 ^
87DC9AF771F2EE543F96DB0C924AE650A063F854 ^
76DB885DD797249A4E2F794171DD559FF056A9FB ^
9AC2E9CD3881BF14D758A7FEBDA81BC9DDD31652 ^
D3D8595D75AF4D1EEEEBEA8166CAFB156E827DE90 ^
5F7D9D9BD0DDC5A2AD4772610FE2A4923A826486 ^
F437BB6CFAA9139A1201FEB58841B18C3809F00E ^
8C3EB402D92E6543D885A9FDE2FC80B9D4340E8E ^
134307C50C24BBD7822DBA30F644B874EBB54F0A ^
24774DA4669C2037F7B2A43BCD1C8D574715F071 ^
69FEF8FBC9034D05F3C9BD9C663D2198AB48FA30 ^
0390F1C49421806B19DF1F886902F37088DF77DF ^
DD535B129F482256503AD2744B712BFEB8BF216B ^
ABEC552E41020144EFE06854149BE03F1BD7F6F6 ^
3FC8C8A14D1452713E76EDD5EF95C239F3F8A4E4 ^
8861A84429E238D94E75D3CC191FB4D128215F7D ^
DBA6377C376AED4D1E79C198EA864B6E6D90C494 ^
4E5DBF823DC5BFF56C999A1FD0ABBC51155017D3 ^
6699420C6FF512BE5CABFE1EBAEFB0AE40CDD902 ^
77CAE0D7EDFD2D13506D096D66A32B9907AB9B23 ^
F6022F2A1545842915029B344B19F4C69757925D ^
6ECD085FF34032636E7BE7269D29DA62BC4BFB5B ^
60CC512BE506BE1A580CEEAD2A009DB8D7A1BBC6 ^
0F34DFD7EDFA6A5DD132B2EE77B52EEAD98DC701 ^
1BD542585AB33A0A3694BE45D6C7EEBC266D4806 ^
1524F20389EBC2D7DE777F96156DF9B2A8652373 ^
451282E6DA707668886AF7E77DB4C58F73CA6EC2 ^
BB7D59890BC39F7F87D4DD7E2583C737B2B40B86 ^
255D67E3AF65376A1D71FFCCF505812481AF182B ^
4D559BC171516F79E22938DE999685F9FF05383D ^
2F7EB16081B2338D631153D5B70DE67F050E348F ^
5902CE6CF7700D2E23CEDEE742E273F4A6A39F6D ^
7D37A7515D1E04D44CA0F314306919C0307FF4AD ^
18C215A5B8AA9CA5C55DF8BCFC79683E9982B159 ^
63520C9DD2CF321388DC89CC69D0D9F6772A3553 ^
5FA13E6697196EBD0E40FEAED4B945CC93375DAD ^
BF8CA60D07D1B69767056B2F80A9AC4B1F9FA99D ^
6CDD13431EB70D9E346102B708E9F8AB233CF959 ^
4CB448B4131DEB8F1AC3085765303CF214436A26 ^
1FDC9CC58A748A9478A9E55C66C6446775382ECD ^
0EDC5CC7CF8C6CB318616D62CC08CD1449960315 ^
D5D0D540287D21AA519CE90B6E3D81958A2E307D ^
BA5CCD323B987F54E06AE04A09957BC417ADA611 ^
26846FD68BA956CEB2899DDC18324843DC211485 ^
12BD70565F701BB35B87364B47F73C93287E7672 ^
C57A0F96959C7DEDD353462099EFE380CB273E90 ^
9900D5268B99165BE1AA2A6B41B9FB66455B865C ^
D5AC2C2BF36B04EAEF95E8B93958BE28B5A4C836 ^
82788DAA7F95292B266CA17E86631D866D68FD47 ^
37B9E638610418C4CE4CF9AAC0D3E64F033D3D74 ^
9845A156240CE847508A1D3762362FA90216E59D ^
9E59DC76EA01567183A4B4AD65CD31F169706CA9 ^

B2B4A1500A176C883268447A09DE920F0CF77F76 ^
E14FA11B60B9389D3B9B2C1E7C45931992F666AE ^
922D169C624E79DABDB79B76E777F6994C339EF9 ^
AE99C2F46A3939CC5EE3F8CF68D6EA486392C23D ^
154F915F90340CBB03B9514810FD399EED06CE3B ^
C440E1E74C75266822919080966AB74D96FF43E2 ^
405155FA99EF5E4A0F4E93D107CAEC89A80B94A2 ^
0AA9625455A0CCDF37D486B4538E790BE407A59C ^
46D633F7F2EC56496D53138A56D11D9ABDB701E1 ^
75C1BD5738EB5380E25EC4FCFCC7BB3FD8164460 ^
B618C9E52303919BF762D452F5681D23ABFF23B6 ^
D08BE1F5485939677D294C4198FA15B118938B3F ^
F931441259B793DABAF082D1B0024ACAD3CB48A6 ^
096E206E3753BBEEC8F0F0D3F97EBF2EB3063B1D ^
13AA397D03C60AA4414271D33CAC2361C138422B ^
777E2979834E8C6E89D219E9109F0A2852961EFA ^
13960A648DB8B7CF541A56B9009A5F7FBCC1ABAB ^
B3447A3C462B53945F8295032FC9A057BC301FB7 ^
8C42386A2436DE83E1959E92B44EBF61E6C8D093 ^
D9EF91B04DD4371968E1978CBFEDB50C2BABBE67 ^
E84042479D1D669AB5871B71D1342AD6D15105C2 ^
4ACD112CD8CAC72DF2D4DA76910C2CD3BE1EC25D ^
AA20982E2B138FB2407EF99509075CE8366F10BC ^
5E3BC535BBFF471686885331CD9831C80FEB18CD ^
53C55C80D64411221C7311654921523D2BD0E227 ^
3FB8C850B9BC7A923767D036992AD79ED7F0E1AD ^
AF0BCA4D440D6599BB91396A0823DE64764C2112 ^
30B13703A702572257D54DC09463768F5AAEF759 ^
59C44B99EDC297B476402310044554B3F29F9CCF ^
FED078596C584810E5336545F6F6B93027D9CBD6 ^
6499E70E5455F9C8FFD2D5F600009A43FAE710ED ^
D26C0718A9C8222F30D021E8FA6508935E80BC85 ^
1F4152447ECB8D7CD725B57B65E30F79B466BBDE ^
6C4958665DDDFC828C1427E57D23359E9C5D1190 ^
B88EAD843C4897B2A691612B837E5A5D9F58172C ^
A51097DC1883B160AB799D2D103C443336EAC2CA ^
5DA1B85BBF4F3912C5E964242A61EBE83A3DA5B8 ^
4E9A45C0410F0211DD4D9123BFB8C29C53A5E03A ^
C2B7C89F7DEA6E5A6CF909388471DEC4CA362255 ^
4EF3B7E7846115212B3E77C29AACA6198422BEF2 ^
78C74563BB2CF819F83E73CC00072378C114639F ^
FAF96E836C381EC80FE1EA8AF76E9D8D8AD95FBD ^
3B30A2BBF84AB70E8220349CBA249DB87FFDCBFF ^
6CE0CC298D12597F801E0F1BDE66CA9DDF6AE66A ^
8274B1DB8C0C7390C383C837FDABDEBBEDD1A8C1 ^
D0F90D63ACD0126C84D714AD7C9EB673DFB459B2 ^
D1E63EB5C373C78EDF968E0651DC1171907258F7 ^
1530DC99565BE2B5AE91D1D70B027EDFD4EC6E93 ^
FEC0946FB9CC04713C67BA36FA36EA9211131281 ^
BD2BA680D2200AFC5CAF362A5517BE9DAB7E593 ^
5FB57E1AAD5BD9C1D176E1CA03C1A5A8E33DDE2D ^
F6D5E7B1B28C9070115DB6665309B8DDF5F81281 ^

BFE88F7A0FD5F6A1B26A5E15D0A109A1DA7C1FC1 ^
E5668AA40375C8CA35FB3046EDD8AD9EDBA82FE3 ^
C854D5C5A0A100411FCC9189D6307C47B57228CF ^
1EE384076308AC6B80FB66856D5FA19952B3D6D5 ^
CAA4DCFEE058D22D726292B2C722D3147F1DCD63 ^
1887C76092C9275B2DE333014C99FD13E5E533BA ^
AEAC6ACB4561985DCFF6C9696BE836FBF75E24E9 ^
CA46F23773B30BF0ABF5C38EB9D0EE021D713F8C ^
2EF007988B2161FED07E5181C342E724D3F640A3 ^
F0F1422DFD34EDE00A8B0741F4F395234480FDB8 ^
2C471BF867F5A02CC241E35D95C97B43D6536E7D ^
D4906C8804B241E28F3F2E6380019FB739EC8503 ^
632183EEF8C6EC8EAA3016A43E568DC38E29379C ^
1808120D461A867C90222469985769AF1756EE5B ^
08C2F97B6FA6E1433DAF56403CDFE8CEECF6BCFE ^
2767E111977A20D318EAE19D59E1E551A4AA7F1E ^
B1A7158CE56666BCD1FDE7DC87B40C92705BFDE5 ^
7CE90514AD734390F6FE58BAE8FFCD66391E9595 ^
C57F4C2D8B78DB8464EF1F719B8B1DA270867480 ^
C97938F34C6DF26BFD509658838F7C35FDB0C58B ^
A5CAA6946300B1C4B5F53A69AAAD67961C4C6821 ^
DDF684947B1667A77A22C7DFE8B6A0A8484E2509 ^
8FB1310F1F215E4EF62C41566C523275BC929670 ^
1573CC447B398532BFB9A2E5AB3880F23CA7C27D ^
D6E0B674B9E4D283C0AAC355989CC0E8BBF09678 ^
7D083BB9C3E005B41FE32629F84B9E292D80FC67 ^
D3E2591B7D7CD29E9ACE61DF6019D8787AC93BE6 ^
7159B10ACA9E7610C7404DE961ABFDB17B89156 ^
B4A0248088E850FFC5CCA99EA2F28ECACBFF1960 ^
CA63F54FDB6C46E877760EF4E50E56F3164A417F ^
C41DF52EB252961619BD4861C4AACCFF1AB392B69 ^
FE3EFE453C8705E1CEDD06F15DF5B35A7311DB32 ^
5E2E8BB04B327E91572992A8A61063F7A76A93FF ^
8EA0191E8491400038CC5B1416E5A56622EF1E8F ^
47727D4D7E8D9738608AEEDEB0E12C9F2427EBCB ^
7340F405BC428DA514E7B2F39728CD77FA2177F9 ^
5E451C7A51AEBABCEF9FDFBF3538AFECB968872B ^
528038274803D6DB419F6923993A45C999433336 ^
281D30FD3C725A4646A6287329140BEA4AC7A5B8 ^
922A57DBF62C01DA19B89F90F36C433817F089B0 ^
C79AB225BBD03145A981DBE7BF6FDD16A5E95D8A ^
301B4D0E270F2AEC6E1372560385D37875EB682C ^
1EF2EC6A3642C7FD61C4F1C3DFC55302535254B7 ^
1D94ECF29D2310B1A5A2042CFA616BA0D19E7C41 ^
52DDA9BEEFAAAE667A6D12BE414FAD8619750040 ^
B14486D16D1521D1283698E651A10FC14AFF8996 ^
4BBFBD0244BFEE440DBEDC8361806CADFF673253 ^
662E0E466C9D700C50C5F98CD77CFECF8278D708 ^
1AC5B8197B0689681B0F474832F316CD25147AF3 ^
8720E012F16588A3C368689113902119C9791FF2 ^
D892C1E257A997A8DF95B229C9D24C99F60EF205 ^
57F5C45286EC3D29CEC19939C04045D9C38E761E ^

07151D2EC889D24730AAA5282FA4DFC49A66E439 ^
6AC94690EA6DB4A8DC982271490EE18282A3B33D ^
32DD19F12AC0854998C7B14D1D96C15FA8995DBD ^
7817A25C6CF1EBFAA8DED7A80063D1A299AF8B18 ^
01C5EAC1830D1F5A7F186E039663CE925FCBACDF ^
B9F32F67FF4937F44094CB4A62BE12474AC3E1AA ^
05BE2CD6882447472AD2E02B08590EDED62C7638 ^
CEF149F6D6FC8987C3693D5F5CCAC0CDDD3D0787 ^
6A7CC99DAD86DA9797B1F8336356C9864F279D35 ^
3A72F8EFE64CF60D49589AB92D48301620CEC0ED ^
F1F5939419E375182A2AA928DAE3CF6C378E4589 ^
3EC1C1B74B8881F86C1D3EDE42F6689C61132710 ^
7164A5B4A69A004FD76EA4C6CDFE0273D35CFF71 ^
224325A309FDF2378C5776B1751BB0694ED30C28 ^
64E0FECBF5F94517EA9E0F4E1E16FC700EE275B8 ^
69CDF3321850BB640D5231D3B3458F83738CCE6C ^
0684349A2ABEC06CE05100EB69D01CD4FB7AB193 ^
1DFA49225C0FD725046AFCAFED17B1ABA6871FB7 ^
ECF4B1B9C78A802F748D58BF6E27FAF1B274EEC3 ^
A44B67EF91F2AAA9EB2DA8B166A1A316886E8909 ^
CF38B0F7DDD2F72A891B41AD026FCC3282218747 ^
F917033D4494A1CCC8580C1DBED6F28FEB3E2E57 ^
3C326F550039F815ECF5E236F6DDE5DB0E2E9F65 ^
7332001002A11FDAF501914A867398625398D5CD ^
DEA189DC41AF695109ADCD93906EB5372B512A57 ^
E0F14BCA9240B27E44647ECEB4A40408E2880DB9 ^
77545EF84146F952212D9E1FDBAF62243CDA630A ^
D162F654DEB5BB0D54179319D8DF9F1BB6D8A13B ^
3BF79007252537CC2EB11E148842E7AC443CA91A ^
6CB2F291C7198C9E4112B991FD4D829DBAD7EFB1 ^
C850258036781C0AE8A083D6A791D3FAEC946C99 ^
DC64C5037BDC93A3D38613D203FB3D7C82A18A24 ^
620B7714289FD82ABB9FC776B644571F72A9557F ^
4BEBE1EF44C30B0B033153F883D5C58E4AE6876D ^
E3815A4E6BF3B50EE312E5A795B3E9AA2D7832AD ^
9CB13D19C64752209D5D4D7033FAF9DE8960C147 ^
3182B9F03D0ABA3BD91A1EB8CDD0F465FD9CCB51 ^
73D8E8B9E6A04E88E29797CC5E2F489350D978A7 ^
732734CCEC18944BCA784B953DCC0E582225AB89 ^
A2152DB03048AE27B47C25DDF99D249504A8ADF1 ^
AB70B32A28D9BFAAE1CA868C4A161B6D4B3A92E5 ^
AB8B90085B47050B87F4BDAFDF971790B7950900 ^
B315A8C87011BACA969A1B79EBAC80A7343742E6 ^
B78A044908CDA9640633DC8BE828E22F90E6D7F3 ^
AC2BC5246C1BD99231CDCCF3CECDEF5688E58D71 ^
0679F888F8C353B807F544C2379BB5A4AE8EB9A4 ^
464B5754E3D42864CC0A98B4E696C827E3254317 ^
7E5385C8A1BFF0FF476905BB6B7515B6D438CC0D ^
0F5C968CA2FCB4A17E0AF0B94682B3AC347C2D12 ^
4F5E353500D63C99F58A28F5870064FA278FC7BA ^
E7A104651C4D6DFD0778EC8AC8E4F001BC7B9DD3 ^
23DCF9A854D1D5B5599B14A0E9A19AB4F2E3BB77 ^

588E429FA1C3055C4DCC99EE5407A64115B54345 ^
39215ACE01669CADFB81E9FB085710EB5B97461B ^
23566449DD29A315D413604A2DBCFAE0D40579A2 ^
1EA87C13DA38FE4C59691D96AF590EAF3908C0F1 ^
5A6170FC6BD7E8C01140E16F248C56890B51323A ^
FF328DFB506B72FDB8DF5716E291A1830BC539D4 ^
7B4DA846F9074254C676F00948BD4C9698148C4D ^
2AC310393EE1ECC5844A4BEC15A9CFC1B5022294 ^
3AE36316FA94B369D7320A8B71304CE6490FECCD ^
7078C7041275BCBE8403DE24D16A646B4462882B ^
8325A8716A2C879888BA9F674C869B1FF31BF363 ^
586200B55B5D537B59B5E0BC6FD9BACE1E72C6EE ^
0B675E7F434756F5E310B80B58F861E789495F5D ^
749C4C0856CC4629DC397678C0E8BDCBD649A6E2 ^
25327C66FF46C4466F81EE635E8EC02C5CA60F2A ^
BD9DFB59EB5E53BA6988734FBE0894194A119D33 ^
3084040E5ECA5E77406B4DF3CB7F324ACC96B506 ^
1E14A09FEDB61CEBE255B94F533C4018E5AF4BAE ^
C8840C051A204F8A13420CF17FE77EEE2FD7945D ^
4790265B5231F9A1738CB9F8423BEEF219DF110D ^
354947C0F2F25EB491D21BD0BD1D809FDD140A66 ^
23762BF3CDF3F484C70D2003B55E6F447BAF46FA ^
E5BFC99555CED5C1743B152A64AE08C9ED1365D6 ^
4CB44DC507F716BED4B4CAC742E79E1A2DFA308F ^
62790EC14E8CFD1A0D6FAF93D7984A1E9416FD5C ^
5C2D3ED988F5BD9579D2F710D5DD68B11584295C ^
CF8414ED85FA0135E44B37A8F8B40756BAED23DF ^
03432064C1C05D28393C9AC5878C9A9BAA6E5C7E ^
C37A6D64CF81CDF7DBB1B81509A67A0A1BB5F278 ^
F341D89B75E599048A911D0AE756E0065669B4FB ^
269E1529285414A74B76586EA4FCECA895C7D1EC ^
FF9BB39F9FEA1EA63D944BF12ADC98D1D4255096 ^
DD114F2DF3D92977F9DB98AEC04F367CFF9FBA2D ^
69B53BB13375F9E123FC30098CD9EDB3C1A1FDF8 ^
2A60984DC79682E96C29CC7050E1ACF988D9E4D6 ^
09AC4CA1A40A7D83A463AF0804C4EDA208E09D36 ^
A2EE2473C270A0637FAAF17DA54237D37E0C6960 ^
B3A7817ADE962DCEEA7FB6F960EBE2BC1F9BCBA7 ^
85A46CCE01A7968104496225483ECE2C198F2A27 ^
BC028F5ECD8773AE47563F1216E0BC36A9E37E4E ^
C65C40CCC3DD80045D6A4EFF3E9C7D6E4E0E5583 ^
BD9C189A639FD3D70167367DF61F14868CE7E313 ^
C6899A84B8E66548BFB50781632CC3D2D1BE8B9E ^
92F3C8E436227976F32F005469DE91B2363150DC ^
7D2A80A5C5CC88988F38334AFFE31C04AA1AFC9C ^
D60FACFA57F663C4A399B1FDDD75973910F27294 ^
CEEB9661BED2E89DDB07A21EEBF590BB1AD644A0 ^
0465C9DF2E0886AC79F46BBBEB0A243DF7D6BAAA ^
98073013597B7AE84CB884BD932E20E6CB47709C ^
3CB4F76B602BF6B58B316A2D8D169D432F02CE23 ^
5CD4EB05E8612974BD9C3B19E48D192BE8B96A09 ^
BB188D5FDE806BD30648B6058143F1802F15FA10 ^

84EF1D3DE5C1144C2C928CD4D9070A6E79543417 ^
73BE2AA66F5FD6BF63273EA136350D29491A3305 ^
53CF3BB07761910BBB486C3C0C643943D5BF17C3 ^
9A774BDBD27C3704871B52CD5F7F924C8EE4A67D ^
B106B6582F0F6C9E2EFBC4860CFC7028F2C47F74 ^
FEAFF44298CBB7517B14B81A6AAD501828C773A0 ^
B5BB48BED969F15D162301966C537A81C40CF4B3 ^
3495D7E36747B824D33465B5DB72D6C623EDA72F ^
D7F49178B43847D9984B923DEAA6DE9571690B92 ^
3AB7E26BDB0409E55063AFF456B7A29CF098AF0 ^
7984E1565BADCD2249EDCA0A3AADB4346104A556 ^
42DD99B41801B0C347580DF1FDFB2397D55D2A85 ^
8D59C00401EFB314A0DFA4D82E1D85711C77E246 ^
50CF74894F7CF5A9F37C37B47327047E910D88A1 ^
1FC40E2840C4B68F40C205B6BB06C3E106B6B8D8 ^
A49F56DFC293411736F27C19FA2A362A545C761E ^
221E48845FE8B3D2C79AFED37414C0D732A69F4F ^
AFC12E7BDF9F48FAB1B595FD656776A39F0F6CA8 ^
5EAA114491E88938276FC3ACCDE659623C024BD7 ^
5081444C3F81E18114B289020D9AEC75FB205597 ^
D4F69156DE85C61E9B4150AC9F6F6A54F8E193A6 ^
3D7C7EF4D362DE137AF0625925B36837C1FC7572 ^
575D772F9C1CBCA8FF38243348E0DFE02643C28C ^
0C821F27FD82C55CB4D76D386F41DDC92F79493E ^
C3D0101DDF552A11C8E1EE6BFADB72A983F38571 ^
F4D650440127390F3A3C2EE991C185CE7650FAEB ^
54460FDAFEF3022BC2C7D1D497CBE5CED73624C2 ^
9B3F7B9D594AAF6A26F5C21BD6E4717F46267418 ^
14E2B34F0B2C89DAE1732400C02E2CA19A767F8E ^
1072ABC4DFD2FDC6C114314D3C08457772421713 ^
C9EA4B020C11DEB794D1D05957473A46A906B472 ^
90BA4272464A020612F0085B7DB9367B7BA8E15E ^
8713A3997A3CD5DA87E97AB94851B68E748268CF ^
370358D54C819703E20EF63A2498C8586B48D44E ^
0DE40249A8B215E98E849008FB7CAE80EC23F4A6 ^
BBDB96306594D13598BB0109258E488D486D98E4 ^
DF7DB9CB2FD8A86854C5DEFCE59C4E2F0245F582 ^
84DE32ED88B3D2BD5F4E33BA3398CD2EC1E9BDF3 ^
D20967844FED11CA71D28D72B28ACFDB14A7DBB2 ^
B13A0A6A7723BCD9098E2B04FA04320C4A0E605D ^
BE66BE38F97E7DDB02EDC8E0ACE97CDF83F034FE ^
28ECC25FFB394AA8DE18F8ADE8B3577955346FBB ^
45068B6A0F61D0AEF708AF7D848683DABE817C2C ^
EAAD829C32C3AD4EAEA8B7E6090A5CF6B2FCAA82 ^
F445FEB22DF5A9E49E2CF7AAD7CC1F5F72F18B77 ^
73C424FA0B7863F1F170A850F6A0F85C87820841 ^
71F0155D40CC55F3D64BEA645608FD184E2543F2 ^
4542FFCF50267B77502D1A0ED0C2A5050CAD5200 ^
FB48F26F320D7A4B803ED459BDF92BC1F8703631 ^
C9F9A37E34BEA6220BEBC167E34B0BAC6B55A6B6 ^
B0C836CC2831FB8BFE5988360CC39FAA6A904A89 ^
B0CF783D84BF5383EF30EF2071E0FBA9ABB553EA ^

8411C8D1A4FC3F130143DEC5232B59BBDDF11B7C ^
ACB6D9EB22D474CFD5836C8CE22F387E12DC20B7 ^
D582FA3DE1DC425AE6A394FE5AC321DFE96B9305 ^
15CA32D0E246F395DE22FAF8F0FD90036C4016C8 ^
F1832FE ECBFFF1330C7A6AC47D6FC1880E79550D ^
C4846BEA159D1D43427CB5802C46E479F9157392 ^
C84FF09BA6A2EE9BD1972B8C0BF25240F5A7EE72 ^
4E6B4F363D7FDB4083AA6F5E7979966FB8FF6C5A ^
7A543B81CB4DC93081D8CB6AED436A31C4018A84 ^
85990A1C35B86BA244F60BA0EBD6F6E801552FD4 ^
D3E07F5D14AE219D3CB59B034764C40056BA95F4 ^
F4C4DCCC63E8FC5402E93CF50533E51D1BECED43 ^
3806AFF9AD3859B150F1E2A40FBC047FB624CB14 ^
B51A4E4FA213D22548CFF28323C6679D2693110A ^
A7C70C19BAC1040698C7AE6A23B0688B58405153 ^
E74FD8C457581CD09F75EE3D43485130609FA9B7 ^
86ED1E49D4CA868934D7C254D3AE0D5BF7323CD7 ^
A16507DABC3A0CF7137150C4E6F88D5AC9A0AA3C ^
DDC76D7BA54354112C2D7D31E1A80F0048551E5E ^
D943F6028146EEB0386BB3084FD354CE86ABCF00 ^
4ABCAB65C7DAA52BDBD13BCC9098643F5A165329 ^
D75FE22162C538BA1E2687D66E2923CEC7375E01 ^
1A11A2B12D93FB030B0C3441D7AAC06DAD6FB527 ^
F8F3D414C26D09C6CA1B88BE44BE69AC879D3C5C ^
D2D0BC98795740002B2EB6EA6AECA5559340B54B ^
A8A3838BA45AB6B05D1D464E8681AD61B136B6A5 ^
5B162923200439D870F57AEC584A67CA2E8939B9 ^
A8FE0B55CFB15D977E871152905D8FDB18AA6B59 ^
02D5D4928969C7A852B5A8AC772A3D1EB424C9A3 ^
4ECE08DFF1FDE1258B1BF0423E1E31D83C57557F ^
F701430EF6CB54BF0D138B54FB13A1B586A1CDCC ^
E3242A71106CB77D25A60D164BD17FE07EAC4951 ^
D8B18DE86132672EDB330190BE895E3106F92D2C ^
8CF05FF1387C61A8CC3724BB1283AAD3E2F85EEF ^
797B311DC2B5D14B5277E49A38A787D3D6A58C2D ^
6472BC5881A62BA95651DBB3DFF9B3885949C9B6 ^
86AED4B53D8617FC135D0E1FCB762271B2545A45 ^
E3C888A1C00301AA7CE203A5590D82C1CE77C6D4 ^
8A081A7A672F91493AC09D95B29FABD043DA83DD ^
D47F87022DB4FAE551FF3FA09FFF02D92D112527 ^
B2271E377E9B158ACBCA1EDAE0ED6BA4C59CB722 ^
71EDE0C9488A21E62456D06EE393F390478C3F31 ^
437A1B709176598020EBA1E92386C3979A18915D ^
E3F4AC4384848071C8A9F9868376D941DEC2A9FB ^
00444CAF6A3F4210FD97DCB8F10F43B28D890600 ^
9B6BB350E30CEE209C77A3E20BB6ACE867546313 ^
935D2688AE2B16B79B517BFB79D48CE7E4F0F4E9 ^
16B21F19ADE7B6DEE665D45A847491B7F055C46D ^
853397C3565C807AE72EDFA37F4198E801103FD4 ^
10992C2FFE5127006BAFB2172359701AB10FA4BE ^
C82D6BA4BA621E36872ACEE4AE3BDF33C2381B07 ^
D7852DBC8FF592E08A1F1EE57A70BEFF327D8EBD ^

CF67308422F072415EFB75B43D5B67C7F5362B0D ^
30DDF35E1E99243F85311A0994D91B8AA543F729 ^
DB2FE340F62A89400AE4A16B61435204405350D3 ^
744980FA26DE503743EED298B966ED5FA8471DF6 ^
D2C8B803D3DDB54CDE6464D126CE3009A9F0FB73 ^
885700ECC04EB97943A9C100EB6EC4ACFE59DF81 ^
770F9CAC5D4E9C878A6636EB0306EE8DDF0CF915 ^
03C3F0553D7CF16CFE3D3F01C0A4CE26A914512C ^
93B304BDB138DB6790DF8A84FFF3D3506770CDC ^
D5C5FA0488C5A6E14770F216B575997950A94184 ^
2F695B8EB18C28250D7C8B35BA000042EABDB245 ^
0353FFA7F7085D3B7888F96B8498486EE7DE16CD ^
239FE3AAC0D3739F5EEEC82B2BF6F5F47D058548 ^
16DE8684395CCA8A5DBAC72ECFAFA4BC3DBFC5D6 ^
AEB2BA742708DBC048B01FD77B1560CD3A23E707 ^
FAAE178B97175A7A68E51BD7EEC06234723B670A ^
B3F1311247D920110972772C3DC30F789D3FDBAB ^
466F682842C07D2C770B41B66A66BE9F6651C046 ^
39724BD29138F776ED20633B66A3A241FACD817E ^
5931F19A1985CD5F7171A5D2B81CE1CB3BA8EBBB ^
27DE5C33A18DF1A731051D6644DE8900E30E242E ^
A4E1C4F20A34C033589ED0985E80168A28C2377A ^
9BE820B134475CA31513EE1C79D06E299D70E017 ^
98F396D147EDF3D3079A6B2DCDDE92D48C9CC52B ^
D6108901909418CF7E7AECD9E4D1CA121E992E90 ^
F2E8F0FF1C652356E6056644B81FF8270A07F56A ^
EA1669ED509120F2B42CFA547A3FD6002FFB862D ^
219BAB55653B03A3CA7886212DF27B23C082B2E0 ^
BBE95F8BBEEB6F9A0062D151A7155630F2010B88 ^
FF6BFC90FE3EAB299892A5D9CA4D17DF2C6C604E ^
8D0E1DBFCC419AEB1C7827C7649333E34AC7E30C ^
EF3847F7FEA054F0E080836D0F2BFB676E3573EE ^
F60A15E6A28A5CCC9C4EFB1B3107A54C9AAEC26C ^
76328CDE5EFE6961B2CED4E3457D219229E161F7 ^
FFAF03FB632A042B837F8660412B72B1099394B1 ^
211BA6675C52CBD9E7F5F5B366A64A2466532AA7 ^
51E05B7481381358782FDF73CB958E1434CF3330 ^
BB5CEAD1F3B68227CB032B2B8E5B627461071637 ^
3EBCF1D78998D4D11C094BF28FD522EA9F5A1E37 ^
3E7186CB1D39C64CEFD28BDA2CCAA073E71A5327 ^
415B6A0083386054D3B77026A38654D45D3BA106 ^
2297372256479E31E139AA16E63C27C207519C0C ^
65A79FC4B322F655ADCF63432D295349FDCA9E7B ^
2D1130EE6E4156151C637B46527FAC54D9E79DF7 ^
80E18DE7C798F040C008B3BFE3054261740E1444 ^
0710A3A300290CC1C3BAB56CF00A682C1B711BB7 ^
C6BE5E337A1A976C253CBF356687F359548B6A3E ^
3D6E4E514277223BADE6A317C478F17EA6EBC9DC ^
25C66C904C35EBAD2613A30B73283E0FC23CCC10 ^
125709FDFD511180B08DB77366DC7E5A2B46F0CB ^
AFB5C53BB272826DB7A1C030B89C7F3AE956F65A ^
D166E71D8142E26E58E0F7A69C50CF19CFE225FC ^

426007235D8F06FDCFF20CE7FD00E994F2FC422C ^
8CA2730163EA816BF9B63CC653B443A8E8BD7656 ^
BB2DA623B5FD10032BAFA1A2F9FB1E13F74AC4EB ^
82238EC3C90BAF3227885BE78534A1B08213660C ^
64BA275EE9422356EAC19A46A4C5C9BDB4D63C05 ^
4995323B85FD3879631B439BF4F8D7D39639A7FA ^
41F95B512991A4E05B219479842285900DF8425E ^
3CEBE929AC763F2FD85F21354C9158E5169A844E ^
BAD5D53D67CBCD7767B6009546A91D59188DE99D ^
E9C2909F6F0A3079890193C8772E745A317C48B0 ^
BB97E31E9B8EAA8CE758DD775D8326F94720E198 ^
109471A1C3A427C65C4BBAF8C785DB99C748E5FA ^
751D9CC64F00176E424A1F331C63EAF2910A75CB ^
27EC395B07FC740569612741E5CF13A9A6048010 ^
F674BD87556FD0F71D9BF23A0ECCE9F39AC82099 ^
93B2F9109A25C511865D15E1683C0683555FFC01 ^
<D

H>SHS Type 2 Hashes<H

D>

EFA3EC89797EE3F1A5BE4B8D34747136CA7DFB99 ^
D146D467E321E4EA4C73E8424029BEA461F3D90B ^
25ACD32DED334AD20DDCCF01059B6B8B864E8D7A ^
25416D69665007C3EDD24161F3DF0F6ACBD0E859 ^
542A29E5C04A5B79AD0986FAD49BE741F09942B9 ^
5B197E1011286E5C130F9379843DE89776C37C59 ^
467277EB8887EDE0F2369C6E7D12281EE3B51857 ^
BC1610204E15A37D533B74412683767898B35DFF ^
3DC7C4726B3EF75AB89193701C990F86AA6DE80F ^
4AD4348E5022685C70A2B9806698664C281E7123 ^
A33E4996BD9980ED505F8AAAF0F6D64E97D7E619 ^
06280ACC2D0DA15EAD7C963253349C86A6C3C6C4 ^
33BE29779F42EE504A3CDAE0422F7B9C5CC994DE ^
412941272C5CE344AC144B69D4BA5F61F1474F4E ^
C33ECAC1FA74D54685669F8B9F85918471A88D60 ^
C48272E42F921A2749D60418965D8A7CB1C7BF0D ^
44ABEEAD2C278098062D934460E565ED3C646880 ^
08DF185007816E3EF01D6226C72B03256478F23F ^
289E89F8F15DD709701AA22B12C69E8F90C93F1D ^
5B93CAB369C99B85EEE3130F833B7BB06D4FC041 ^
7B2872C58FB340C9A1D068B618E4980B04FDF2D ^
62783421CF58EA66366ADDCBFA96968FB1369918 ^
C8BD732D0181051698160B2FBC2025476297C96B ^
038D587CC069C9B89DBC2FE9D86A98C031D340B9 ^
3059031A68585DA8F09B87B0BE3413987EB390BF ^
1B74A4A95713083609B27E67140708EDA15F80B1 ^
6EACEA69922C7ACE38E0560A91DA459C0CC89B55 ^
60E21F5299EC719D2832A82268B24AF6B2815DEB ^
3D7D51782A532F4F410AA2AB3DC9178C21A55F0A ^
B2423D96008FB800E5E35E19DA6BEA27487585E0 ^
5038FE9AEAC82966A4EE5CC69ACC8ACCF55E63DC ^
90BF0A90F88F5F1DA51E4494E2D5D4C6250D2D4B ^

F858434E0A5FD61A220CC792380801D19E0D9F77 ^
C31E0BB45FF1DC0DE44C77C53C59B15C3BDE9EB4 ^
DF5628CF6445C409B51F46A86488A95DD34A3DA5 ^
744CDCC9A51B1AD14FAFD673D4104433B7A24739 ^
CEA5E7B91773DFE13649ACA78907769B5BEBCA1F ^
DC23EC1A04F7417D9704B844E9F39F39EAABE6D8 ^
D39DDE97297B7D1C9FFD6B6733349D0B8E802688 ^
65219E79F5A3C61FC075660258A9C5F13A159271 ^
D2A86164D1366B60F146651464761B01353F9028 ^
8E3D1E4E0438EABC58AB967B1CDDDAE3CDD9B078 ^
EE9BEEC73D0E760CFD05CF62382A461C7DE3CE2A ^
5C4FDD604751596BE5419D4AC256C3D982012149 ^
43ECD0DE83F087815CF257982914611E4BFF698C ^
1A1BCA41BA127DF4264D3795E2EBCEDCD1BD8EC7 ^
FF82BC6904520E186803FFE861DCE7E566D75F03 ^
764546FFA014DF8278974BE66D01485251B8B0CF ^
A06BC97DACB07EF207808684978841C425AE1238 ^
6C7E6E0E89F2A070EDF3C4333353869942F3D883 ^
572437B2B35C5BA00EBC74521B12D20E6F14ACBC ^
08703186E74A9397CEBB1B2034A6770995AF3D0F ^
E4F9055356AABA933EEF0C0B07B6FCE90361D208 ^
2020203EFD4C5FFA02E593B7A658024B33FB368E ^
2BB2F32D64ADA56A3E53A6D750A05F8D2B5D7A7D ^
C719AEE4708E7A242440C9429B7236DFDEC02B4B ^
924A26B46649287197E142CB42506C96E309EB26 ^
0463F94B5452A2D8BA3A0DBFE9C490A371C19975 ^
9FB2FE5393238EAAF58034BC934B190784A10DD3 ^
AD4A6962A6A59CA32789AC3BBA368583B9122AF4 ^
2A89459657F9E2685D794E7AEAF30A05B06AA573 ^
9B65CB91D0A24773AC8675FE9862E28A7B84954F ^
9CB7CADEDD391A3689B04427B461C4113717B6C8 ^
F1B922D7EC240FC99692EC28C1A6CC7D907B1018 ^
DE301A1413C36A5CB891FBCD9CF9258DF4EF14A4 ^
B330738D2AD53448C287BEEFE5C5583F2B817A03 ^
B80EBE0104A25840CAEF6AD74C4676D524571A29 ^
3B5CD01661A279BA72DFC1272E683573B4F16DCD ^
1412D6A2DBBC9D00641A71164A690C640852B56E ^
0EF514429CAC95CDCBE8A2E60C2AA8B892AC5B33 ^
5F32CFF3B07D47EC13EB54E7C39E4F5216FDBDC5 ^
B931966169D171DE8F9488E8B5E0960124BC7451 ^
E5034D3A5A573442781EF0C6EEA7BD39FAD0B94A ^
F97CAC5013135944FDF8BDCF02B4CC61241BEA6A ^
FF74C93CF32506B4E17A0B596AB564989F20F1D4 ^
5CA42E33BCE4301B8FBCE56BFBDB43EF79E956D8 ^
87F5D981EDFA01A2E9885BD52CC4E649A2330621 ^
63F43844ECDDF338203DFB116FA79CBAF4E5C92B ^
92C5E2A5D380D2BB5E19E9E65DC9A77F4B8B25F4 ^
E3D78501969ACDD78A71042E7B0BD0C14882E402 ^
4E1FD6F5CFE29A0E33248F2E2EFB74E7E0A99765 ^
5FF5DADC93B2E452A60677C8CD0CBAF7BD4895E4 ^
CA490FAA2D6723F9F682D5C96586460D0C8B2010 ^
94F099C2FF7DB694FD2DCF5A9A6B06855B2A9948 ^

CF3B1E0DA7BE87180BF1847DEBEC706788C53B8B ^
F85305C4BF149B92068D363EF19B914DB90618DA ^
B4BAFCA3176B80E9800E319FB3BB74E5D65BEB64 ^
38CCF51915A38559A96E6F3258148D5E63230984 ^
6FF43B0D25C7171FF178DEAB13D7F488D21C4216 ^
3E4A034152D86E4568A2197A5328E8E97363287E ^
F231D4C3ABBE0FDF77C71EB2C28E50ACD9584537 ^
23E1A4F73F288233847E647CF8623289F1F89E55 ^
726DAEAE724E79232611CBB216B7722594E814FB ^
0E5FE512B0AF52B4183119508CB01AF5338C63E3 ^
4A5D3E12EB29B224202C0B319B086535E99F5E32 ^
363574A910830EDCFDAED1055975F0613D1B5C95 ^
4820BE6C74AD34D8A43A5DD0130CBDE3F890C53B ^
FEB6E5BF7A66ED9B838A10305F850052DDF9207E ^
D50B331AD6925545264B6614C3B4C39375AB1F65 ^
36B1BF39BF4AEF05FB054BD02FAD9062D2337239 ^
<D

H>SHS Type 3 Hashes<H

D>
9B30CAFDD7F5E525221EC036D55A9C8D78294DBD ^
9BCA33B3DAD2201D0F1981CCD0AF4874E162A6DB ^
8B06697E4A75DA5F020CEB860A29040B3DF6EF21 ^
F0D48401AE22498E281CDAAF045FFEBAA87A9776 ^
6247DF62DF5D4857A795996039B5A6271F7A2031 ^
D98433C3372F1405FCC2475959B9F35DF27D32E9 ^
C4D14B2873B0BDF01CA7C1628EE01D61682CF1C8 ^
C027A9C010F8EBE89FC88C1C9FB68DAB09A0CFCD ^
4D8D19F35FAECDEB015E116130A0A4C836A42A6A ^
928E2E95D3F09D8EE63CB8F10296EE6EF7455FFF ^
55F3E770DCA28CFB711D059174CC7A9EC2B2DD7B ^
4500987086FD8C275330E0F27E8A3ED8596D01AA ^
60BC57D16F50E2ADD1724E0C7DDD5E47DDF8E300 ^
59D174272AE602AC092EE7EF393AC19B87B8E865 ^
8C25614E6E4D825D6AEDD93A62D1A0458583EE84 ^
695390D1E57BCC4CDD442879F0978BD823322E5C ^
6F353EE6B5F4B8272E4A0D3D4BBCC83B1B942516 ^
4B3F671874220DCCF49B580189E6B13FFF0F531D ^
BC6C473B4824D8E2FA04DDA86A5865D4A36E8E91 ^
1BFEC269FD18D25DC5C56ACD83810022775F3EC8 ^
3E9921E2F685413C0AEA64801945B48CB76E195E ^
EEC7221FBAC5F27B56CF68C396B7EBBCAA3C0B05 ^
F86DDD2A8229532C3F183EA9903AD228A2521593 ^
FA17B5DF33E4326BCDCB27F1CE8B83BE8B8EF890 ^
A4AFC0B41D30CE796C2B4B06E368A5D390735889 ^
745AF12B4EF985E4D2F848D7C7ADB6B32404B6E6 ^
5CE4B657409033BF67DB7B0CA480EC2ACCA801EC ^
0B5AABC9D3A8E861103D6B5B9B994F9C743BF058 ^
86A1198983A73298BF4EB41A61EAAC3E4B06B408 ^
460A107A6AF329B330DE9425B307775553AD370 ^
62C81D06AFE8C8F3E642AD42374FFE486DE86F2F ^
FC1EECA32F7E434BBC74F4360808EDB141087BDC ^

BF494B4E8D3311F167B1EAAE58A3343683F337EE ^
961247D01E1C4CC0D71307637CD559D530521FD5 ^
B34F846411BE61E15361922E0E2AF83F3F03B291 ^
CD845B240FC2551DCB2CE487646648CF249AE8F0 ^
5C124B615FAC548DAA35AD0287E032EC77D220BD ^
3C74771D838CCF7FAE2DAB100CEA7B46BE2FFF3F ^
62860DB8C0C897C7369A1AB61747B04C0AE9F8A6 ^
C98400CDD7FE71C7B35488C16D3395C04300EDFA ^
95167E2BBFC0349C0DB7CB7BD2030BC30F9F286A ^
9F835DBB27076B5F0D8880603F417CE83A0D2242 ^
08442B715C84B7A8B2C7FFB46010AE56A0C8C05D ^
B7132DBA522C54F42903CAEB705167485AAD30B0 ^
9E99F03ED08374EFA4F9A26551A909F40BEE503C ^
999977E6AFFAE0451E93E7F77C7D177DE121402C ^
30E0BB7F21433A348E3B446F08D80CEAAE06DB71 ^
B3A41EEDF017DA7B20FBD9111BA6E2E59842CED2 ^
5B9CA331BD03D97E8D057DFD35C9E600F90C57DC ^
5338A25B31CC9C64552B6F6742F9FEBDC13D9733 ^
1B3A55BF6FD748B3B650EA5BE839B61530C871C0 ^
1503F21523CE8BC5A19E6A7F5CB0EC34A7E9F75C ^
7F6A16B30377D3D0014B0B0B4C8639160C721DDB ^
8F621F8CFB6AF216746D8B99C913267DB14ADB08 ^
42E9177C487B4B711A4451C360BD7E2C6DA73446 ^
5B2497CA6EC2426B12653D711E6BFDDBEEFD4A389 ^
30F4690FD4F09AC8E29B926F64C673539D7FB65D ^
CE12D1AA4961245E606097B672EC3E16103D0D16 ^
9BE5D693BC0CC56B34B8DDB92FACEC2F8F4B638A ^
D2040005AE230EFDAE95942372607577EFA5DBC2 ^
8A920F9407C74FA37D9C93C3606B996DB945841D ^
F26915C08DC1D3D31D9CA73F18F04DBB57E8A672 ^
E449381C8148837B9E44251B17825530D8754C94 ^
BD8402B21520E40E31D1EAFD9C99A9E409C3A77D ^
624CCE538A78CF0FF788B9BE433A008FA7779280 ^
B796B521B90D55396E8EC02AD9B69B7E91093E4C ^
5C4D584F97C1D8FAF810CEF3A4CA0915FB0B630A ^
A179FEAF341BA97B1D0278822ACCAE5FCF8F4ADE ^
693DCF85919E06A5423F92780E79C2C0C4528D67 ^
DB94BEADFEA3682051D622519AF9F87BAB28C5F9 ^
2E20D1925FBD460E7A5E6B48A92964A7DDCB874F ^
86651FD71862CF72F01F099F68CC164DEF0E9625 ^
D5948BDCA91D6707015C79BBC2BC74C1B60573F5 ^
8870B8A6884FF54E0E400441FB2EC74F2452F39C ^
7DF1C3C052E8396D251E9D522C4227344DF92C11 ^
59A290A0BD41B973DEE35947CB240006DBCEC582 ^
DD0EDB3AA60CDEBB85DC01258B9F49F3A5E83589 ^
FC6794E4B4F2735773A195F9BB2022C70AA4C754 ^
8BBC40A02D87A8CF14D6CBA0A2C9857D93EFDEFA ^
D28FF6E2366097135EE565F10263E8798140C5DC ^
2E4289EBE02E75E53B52D6724CDC16B89ECF3DC6 ^
F2E716831CC0CB5CB2A623197D14C2C9FC25A914 ^
47B926A13D65377A8E2AF27A84E175BE93739E38 ^
D368B8ACC50353EC8070936C73242A2BF6CEC3AA ^


```
8556025BED695B975CB924DDFB372F6899A7B09D ^
9B4ED89E63D17B7210A3FC1BBF0FDA607694F5D9 ^
0D2221F5CDA184EC7F6CCA29A75F502FCA25673A ^
3C152CCB811DD9BC16529B23DFA1757D744FE676 ^
48D1808CD62F868FB6F45560137CFB1DFD856266 ^
8A3FA42A984C95CA75D6AE51391AAAC0688C2ABD ^
AB570D91E9B6F3F1188DC8639A3B144837EF1ED8 ^
673F62A7D205D5C90B7454E81A5EDA9A60650D91 ^
D00575E45FA324FA05142359BC3A07C803836EE4 ^
B908BE44A2A66F7370DCC022378A9C2F9BA70B38 ^
1EF79029F88E7A091D6F2FA2F44A96829C120C62 ^
491CB2EE0994EE6A4FAC59C4B9FBC9CF8183DE8C ^
CEAAC4E211E7DDDCE3931C47ACA3D626B62E5054 ^
BE2041CB55A4882A8335A46B744ECB8611336F5D ^
E4C0BC6857AFFE0C2B77B38FC01829B44247B6CC ^
122A6AF11DEB6C3B97E6ADF5E077B5D6363887E3 ^
<D
```

File = bit-messages.sha1

```
# Configuration information for "SHA-1 Test"
# SHA tests are configured for BIT oriented implementations
H>SHS Type 1 Strings<H
D>
0 1 ^
1 1 1 ^
2 1 1 1 ^
3 0 1 1 1 ^
2 0 2 2 ^
4 1 1 1 2 1 ^
3 0 2 2 2 ^
4 1 1 2 2 2 ^
5 1 2 2 1 1 2 ^
5 0 2 2 1 1 3 ^
4 0 3 3 3 1 ^
5 0 2 2 3 2 2 ^
5 1 2 3 3 1 3 ^
5 0 3 3 3 2 2 ^
6 0 2 3 1 2 3 3 ^
7 1 3 3 3 1 1 1 3 ^
7 0 2 3 1 2 3 2 3 ^
6 1 2 4 3 3 3 2 ^
7 0 2 4 3 3 1 1 4 ^
8 0 2 3 1 1 2 2 4 4 ^
7 0 2 4 2 3 1 4 4 ^
9 0 4 1 2 1 1 2 4 4 2 ^
8 1 4 4 1 1 4 2 2 4 ^
7 0 3 3 4 3 4 4 2 ^
11 0 1 3 1 1 2 1 2 4 4 4 1 ^
```

6 0 4 3 4 5 5 4 ^
7 1 2 5 2 4 5 4 4 ^
12 1 1 2 3 3 1 1 1 4 2 1 5 3 ^
11 0 2 3 3 3 1 2 2 4 1 5 2 ^
8 1 3 3 5 5 2 5 2 4 ^
9 1 5 4 5 1 2 1 5 3 4 ^
9 1 5 1 5 2 3 4 4 4 3 ^
12 1 1 2 5 1 1 5 5 1 4 1 5 1 ^
12 1 5 3 2 1 3 2 2 2 1 4 5 3 ^
10 0 5 1 4 5 5 4 1 1 3 5 ^
12 1 1 5 2 3 1 5 1 3 3 5 1 5 ^
12 0 1 1 2 3 2 5 1 3 3 6 4 5 ^
11 1 3 1 5 1 3 4 3 1 5 5 6 ^
8 1 4 6 4 5 4 4 5 6 ^
10 1 6 5 4 3 3 3 2 6 2 5 ^
10 1 2 3 4 4 5 5 2 5 6 4 ^
13 1 3 1 3 1 4 1 6 6 5 1 2 6 2 ^
10 1 4 3 3 6 6 6 3 2 6 3 ^
13 0 3 5 3 1 4 1 4 3 5 3 4 1 6 ^
10 0 2 5 5 2 6 6 5 2 6 5 ^
12 1 2 2 5 3 6 6 4 3 2 4 3 5 ^
13 1 2 6 3 6 5 2 3 2 1 2 3 5 6 ^
13 1 3 5 6 2 1 1 6 2 4 5 4 5 3 ^
16 1 2 1 4 3 2 3 2 5 3 4 1 4 1 5 3 5 ^
11 1 3 1 3 6 7 7 4 1 6 5 6 ^
12 0 2 5 2 7 5 5 4 2 2 6 7 3 ^
13 1 6 4 4 1 4 2 5 2 4 2 7 4 6 ^
12 1 3 4 7 6 7 1 4 4 3 2 5 6 ^
16 0 3 2 4 5 6 1 1 1 5 3 4 1 4 4 5 4 ^
14 1 5 1 3 7 6 3 1 3 7 3 1 5 7 2 ^
14 0 1 5 3 2 7 1 5 5 4 6 5 3 6 2 ^
14 0 4 7 6 6 2 3 5 1 1 3 6 2 3 7 ^
13 1 6 4 6 3 7 5 1 4 5 1 7 4 4 ^
13 0 7 5 3 1 4 7 4 2 2 7 5 4 7 ^
15 1 7 3 6 1 1 2 7 5 1 5 4 6 1 6 4 ^
16 0 2 5 3 2 7 2 3 5 2 1 3 3 3 7 5 7 ^
18 1 3 7 2 4 1 7 3 4 1 5 1 5 1 1 6 1 2 7 ^
17 0 2 1 7 3 7 1 2 1 4 3 6 6 6 2 2 5 4 ^
22 1 2 2 1 1 2 6 2 4 1 1 3 5 2 3 4 5 6 1 1 2 6 3 ^
12 1 7 8 7 2 1 3 5 7 8 7 4 5 ^
15 1 3 5 5 2 2 2 7 4 2 8 7 4 5 5 4 ^
16 0 1 4 4 6 1 2 7 1 6 4 1 8 4 7 4 6 ^
16 0 8 7 7 2 5 2 1 5 6 1 1 5 1 5 6 5 ^
14 0 7 3 2 3 8 3 3 8 2 8 3 7 7 4 ^
14 0 4 2 6 5 4 5 3 8 8 4 8 2 6 4 ^
14 0 8 7 1 3 5 8 2 8 1 4 5 2 8 8 ^
16 0 5 5 2 2 4 2 4 7 6 1 5 8 3 8 7 2 ^
16 0 1 6 6 6 6 4 2 2 6 4 2 4 4 6 7 6 ^
16 1 2 3 7 4 1 7 7 5 7 2 4 7 2 6 6 3 ^
13 0 7 4 8 6 5 7 6 3 5 7 5 8 3 ^
16 1 4 1 1 5 4 3 6 5 5 8 5 4 8 5 3 8 ^
13 0 7 4 4 5 8 3 8 3 8 6 6 6 8 ^

14 1 5 4 7 5 8 4 6 1 6 2 7 8 6 8 ^
18 1 3 4 4 5 3 4 1 6 3 7 6 7 3 3 2 8 4 5 ^
19 0 5 8 1 2 1 3 5 3 1 6 6 3 7 4 5 7 1 4 7 ^
17 0 5 7 7 2 2 6 4 7 1 7 8 1 5 3 2 7 6 ^
14 0 9 3 6 9 3 4 7 6 9 1 7 2 9 6 ^
21 1 8 6 5 2 2 4 3 1 1 6 2 8 4 2 5 4 5 1 2 5 6 ^
16 1 2 6 5 4 9 8 1 8 1 7 6 7 7 1 2 9 ^
15 0 6 9 5 4 7 1 7 3 9 4 7 7 5 2 8 ^
16 0 1 7 8 2 7 2 9 8 8 3 4 5 2 4 8 7 ^
16 1 1 6 8 2 8 1 6 7 7 5 9 2 9 2 8 5 ^
16 0 4 2 9 6 2 6 6 7 9 9 1 9 4 1 5 7 ^
18 1 6 9 2 7 2 8 5 8 4 3 4 4 6 6 3 1 6 4 ^
19 0 1 7 4 4 4 6 1 6 3 9 5 6 4 7 2 2 5 5 8 ^
17 1 8 5 3 7 3 5 6 4 1 9 2 6 5 7 9 1 9 ^
18 0 8 8 6 3 5 3 3 5 5 3 8 3 2 5 5 4 7 8 ^
16 1 4 7 7 2 7 3 8 5 7 5 6 8 5 2 8 8 ^
19 0 3 7 2 3 9 5 9 7 7 6 5 3 1 1 5 2 3 9 6 ^
18 0 2 4 6 7 8 1 2 9 7 2 3 5 8 9 1 7 9 4 ^
21 1 4 1 8 3 3 1 7 3 5 7 4 1 8 3 6 3 5 7 6 8 2 ^
17 0 4 8 3 7 9 8 6 8 4 2 8 7 2 4 4 5 7 ^
18 1 3 9 7 1 4 5 6 9 2 9 1 3 8 4 7 8 7 4 ^
25 1 2 4 8 1 7 2 7 5 5 3 2 3 2 1 2 1 1 2 4 2 8 7 8 7 4 ^
25 0 9 4 6 5 8 1 1 1 3 2 1 3 1 1 1 5 3 3 6 5 7 4 5 6 8 ^
19 1 7 5 6 6 8 2 4 1 8 7 1 6 3 4 8 5 8 5 6 ^
19 0 6 1 2 1 7 1 10 7 4 5 1 6 4 6 8 7 10 9 6 ^
16 0 10 10 5 5 9 7 2 5 4 3 6 3 10 10 5 8 ^
18 1 2 3 5 9 8 5 10 6 8 4 2 2 4 7 9 6 9 4 ^
18 0 5 10 3 1 4 5 4 6 8 3 7 7 7 5 9 2 9 9 ^
18 1 4 9 7 5 4 4 10 6 9 3 6 6 8 6 2 4 6 6 ^
17 0 5 7 4 8 5 9 10 4 9 1 4 8 6 6 5 9 6 ^
17 1 3 4 8 9 1 5 8 10 5 10 3 8 1 6 8 9 9 ^
18 0 8 3 6 1 3 10 3 7 6 5 9 3 10 9 4 6 10 5 ^
16 1 9 1 9 6 5 7 7 10 9 6 1 5 7 10 8 9 ^
20 0 6 4 10 10 4 3 9 8 10 3 3 5 2 1 8 1 3 2 9 9 ^
17 1 10 6 10 6 8 3 6 7 3 10 8 6 6 4 4 9 5 ^
19 1 9 4 3 4 7 3 8 10 2 4 10 7 6 10 1 3 5 7 9 ^
21 1 2 2 2 3 9 2 8 3 9 3 7 7 4 9 9 5 4 2 6 10 7 ^
17 1 5 9 8 8 9 10 9 1 8 3 10 2 7 2 10 4 9 ^
23 0 5 4 10 4 10 7 8 3 1 1 3 4 2 1 1 10 2 1 4 10 10 10 4 ^
19 0 9 2 5 7 9 6 7 1 8 9 1 2 1 10 8 10 4 7 10 ^
22 1 2 6 8 8 9 7 2 1 3 3 10 6 2 6 2 6 5 2 5 10 5 9 ^
22 1 8 7 8 10 1 6 3 7 4 5 5 5 1 9 2 2 5 5 2 10 9 4 ^
21 0 1 10 2 6 10 9 4 3 3 7 10 2 9 6 10 2 1 5 6 3 10 ^
21 1 8 5 7 10 5 6 7 3 3 1 2 9 4 8 6 3 9 1 7 7 9 ^
22 0 5 5 6 5 2 3 5 3 11 4 5 1 10 6 4 11 8 3 6 5 6 7 ^
20 0 2 1 9 9 3 6 1 8 11 6 7 11 6 1 10 6 7 5 5 8 ^
19 0 4 1 7 10 4 8 10 8 10 4 11 9 6 1 4 5 6 11 4 ^
19 1 9 2 4 11 11 11 7 1 1 6 8 1 9 8 5 11 5 4 10 ^
20 0 8 3 10 1 7 6 10 8 2 9 3 8 6 8 9 3 7 5 1 11 ^
22 0 4 5 8 4 10 6 5 4 5 3 3 4 1 5 9 4 10 9 6 7 3 11 ^
21 0 6 5 8 7 8 11 2 1 11 10 3 3 2 9 1 3 6 9 10 10 2 ^
24 1 3 1 1 9 4 2 5 7 3 10 11 5 6 7 1 7 7 8 7 2 2 7 4 9 ^

23 1 9 4 6 3 7 5 10 9 8 9 3 2 1 7 2 2 7 6 2 8 3 6 10 ^
22 1 7 10 2 10 5 1 8 7 7 3 8 6 6 8 3 2 11 6 3 4 7 6 ^
20 1 11 3 7 2 10 10 7 5 2 4 3 5 11 3 11 8 9 5 5 10 ^
19 0 6 2 6 8 5 9 4 9 10 9 8 3 2 10 11 2 6 11 11 ^
20 1 8 2 4 2 7 3 9 9 11 8 2 11 4 8 4 8 10 6 9 8 ^
21 0 6 10 10 7 9 2 5 11 9 10 7 3 1 4 5 4 10 1 2 8 10 ^
18 1 8 11 6 10 8 9 11 6 7 7 2 6 2 6 10 6 9 11 ^
21 0 5 11 1 6 8 11 8 10 5 10 3 7 8 9 9 1 1 1 10 2 10 ^
25 1 5 10 2 9 9 9 7 2 2 4 6 1 11 7 3 3 5 5 2 11 2 5 5 10 2 ^
21 1 1 8 1 11 6 9 3 6 11 7 11 8 6 9 4 4 10 7 1 8 7 ^
23 1 7 7 5 11 1 4 7 11 11 11 9 2 1 3 8 2 8 7 4 5 2 5 8 ^
24 1 4 11 5 10 7 7 9 5 1 5 6 2 9 1 5 7 9 9 11 1 1 3 1 11 ^
27 1 10 4 6 8 5 7 11 4 10 8 6 4 3 4 4 2 1 1 3 3 1 11 2 3 4 10 6 ^
21 1 7 11 2 7 10 10 9 10 1 3 10 4 3 11 10 2 5 3 9 11 4 ^
22 0 7 9 10 9 5 2 6 6 9 2 7 4 9 8 4 9 2 5 9 8 10 3 ^
21 0 12 2 6 12 7 9 3 5 11 6 11 3 5 4 4 11 10 2 1 10 10 ^
27 1 1 4 12 2 10 1 6 5 5 4 1 10 3 1 12 3 2 11 5 1 4 6 4 6 11 11 4 ^
22 1 8 9 4 8 12 5 5 9 3 5 2 3 4 12 12 8 5 3 1 9 10 9 ^
24 0 5 12 8 6 2 4 11 11 2 7 11 9 4 5 2 1 8 3 10 2 5 2 8 9 ^
21 0 3 7 7 7 5 9 11 5 9 11 2 9 7 5 6 11 2 12 4 9 7 ^
26 0 4 4 6 4 7 2 4 2 5 7 2 6 11 5 9 8 2 5 10 2 10 8 8 5 2 11 ^
23 1 3 3 2 6 9 10 5 6 8 5 8 11 3 6 10 4 6 9 2 11 10 3 10 ^
25 1 1 3 12 3 7 4 5 5 3 8 9 2 4 8 7 7 9 1 9 11 4 5 7 7 10 ^
20 1 11 9 8 10 11 1 7 12 6 7 4 2 12 12 2 3 9 10 8 8 ^
20 0 12 11 1 4 10 6 10 3 9 7 12 4 11 8 4 6 12 4 7 12 ^
24 0 8 6 4 8 2 2 9 2 9 6 3 6 3 3 10 12 2 7 8 7 8 12 10 7 ^
26 0 5 11 11 7 3 3 9 10 3 9 3 3 7 2 9 9 1 2 4 12 2 5 5 6 10 4 ^
22 1 12 9 10 10 12 12 7 8 2 7 1 6 8 2 8 7 6 3 1 1 12 12 ^
22 1 11 10 2 11 10 8 1 2 8 12 8 3 9 5 7 9 8 7 11 1 10 4 ^
22 0 10 1 8 6 12 2 11 7 10 4 10 6 5 7 10 11 3 3 8 8 4 12 ^
26 0 8 5 5 12 1 8 6 7 6 11 8 10 4 3 7 6 1 5 1 1 9 1 5 8 9 12 ^
21 0 1 12 4 11 12 9 2 12 12 3 12 5 4 12 4 1 5 8 12 9 10 ^
26 1 5 3 8 1 11 4 7 3 3 10 4 9 6 6 4 3 9 11 1 3 7 10 7 9 6 11 ^
23 0 7 12 3 5 8 9 7 10 8 11 10 11 6 1 8 1 1 6 3 9 8 12 6 ^
25 0 1 5 5 6 5 12 1 1 1 11 9 9 3 11 9 6 1 11 3 1 12 8 11 11 10 ^
22 1 10 9 7 3 10 2 6 4 11 3 11 6 4 5 5 11 12 9 7 8 12 9 ^
22 0 8 5 10 7 12 8 3 4 4 5 8 9 12 6 8 10 6 7 11 4 8 10 ^
26 0 3 5 3 11 11 2 1 9 1 10 4 4 6 9 4 7 6 5 10 10 1 8 9 9 6 12 ^
24 0 12 11 4 10 1 3 7 5 11 9 11 7 2 11 5 4 7 8 1 6 12 7 4 9 ^
28 0 1 12 10 2 2 2 4 9 11 4 11 7 2 4 3 11 7 10 5 10 5 5 3 5 2 7 5 9 ^
26 1 4 1 4 2 13 6 11 5 6 8 5 1 5 6 3 8 1 5 8 9 9 10 12 13 12 2 ^
26 0 12 3 1 11 2 12 6 8 6 8 11 4 11 10 9 6 3 3 2 2 2 13 2 1 13 9 ^
22 0 3 3 12 13 13 7 12 2 8 11 9 10 10 1 1 13 11 13 4 1 11 3 ^
26 1 2 6 2 8 12 3 7 12 8 9 8 3 12 7 4 9 2 6 4 9 9 4 8 2 4 12 ^
24 1 8 11 11 3 13 2 10 3 10 8 6 6 4 4 13 4 2 4 8 1 11 12 9 10 ^
29 1 1 2 5 1 1 10 3 2 4 9 11 8 2 5 2 8 10 2 10 11 7 8 12 6 1 9 8 4 12
^
23 0 6 12 7 10 7 7 13 2 2 7 10 2 8 13 10 7 11 2 5 9 11 13 1 ^
25 1 3 11 2 10 5 11 5 7 10 8 10 1 10 3 11 7 9 5 3 3 4 5 13 12 8 ^
26 1 5 13 9 4 3 3 4 3 13 1 6 11 10 8 13 3 4 11 9 1 9 1 7 8 7 11 ^
22 1 1 3 13 8 11 3 4 10 10 12 7 2 11 4 7 7 6 13 13 9 13 11 ^
26 1 7 9 7 8 9 9 5 1 13 3 3 4 9 2 1 8 9 3 9 12 8 10 10 6 11 3 ^

23 1 3 8 12 8 10 4 3 8 5 5 7 11 13 11 12 11 4 12 3 6 5 11 8 ^
27 0 3 10 9 6 9 7 2 10 4 4 5 5 2 12 13 5 3 1 10 1 4 7 8 13 13 12 3 ^
30 0 2 6 5 4 7 3 10 6 13 6 3 9 6 2 10 5 3 8 4 1 11 3 5 3 7 11 1 12 9 7
^
29 0 5 2 13 6 5 2 6 3 2 9 5 3 6 6 4 10 2 4 7 7 10 13 9 6 10 6 5 9 8 ^
31 1 9 8 2 7 3 5 11 3 7 7 5 6 2 1 1 6 2 11 5 4 5 7 9 1 11 10 2 3 9 12
10 ^
28 0 11 1 4 1 12 10 5 5 1 7 2 3 10 7 12 9 2 8 2 6 10 5 12 12 4 9 2 13
^
26 1 11 9 8 12 4 12 7 3 4 4 6 1 2 7 12 4 6 8 10 11 11 1 10 2 9 12 ^
28 0 10 9 7 10 1 10 9 12 7 10 7 6 6 5 3 2 3 1 3 12 1 4 13 3 9 7 9 8 ^
27 1 4 11 1 5 6 9 12 12 13 13 13 1 2 13 1 2 1 12 12 3 2 2 8 3 13 7 7 ^
29 0 6 5 5 2 4 2 11 4 4 13 9 13 7 3 12 8 3 7 6 4 12 3 1 8 11 1 6 13 6
^
23 1 11 6 6 11 7 13 11 6 7 4 4 12 2 13 5 12 5 7 9 6 11 11 11 ^
23 0 11 13 11 7 4 3 6 12 13 10 11 5 12 13 11 6 1 1 4 7 13 10 7 ^
27 1 7 5 13 5 4 8 7 3 9 6 7 11 7 1 9 3 8 5 8 8 2 13 13 8 5 13 4 ^
30 1 2 5 8 4 13 10 6 4 2 6 1 1 12 4 13 5 10 4 10 8 5 7 1 5 5 12 4 8 6
12 ^
28 1 12 1 12 11 10 2 3 10 1 2 6 11 12 9 11 12 2 1 9 4 1 5 7 2 12 11 4
11 ^
29 0 6 8 7 8 8 12 2 7 4 12 11 5 5 3 5 13 4 11 8 8 9 2 2 1 6 2 8 6 12 ^
28 1 7 13 5 7 11 3 1 9 2 5 12 8 6 2 5 9 10 10 11 4 5 12 5 4 6 5 6 13 ^
25 1 14 4 8 2 9 11 7 4 3 2 8 14 6 14 7 13 5 5 14 6 5 11 6 6 13 ^
26 1 3 5 12 1 11 13 1 3 3 10 11 5 6 12 11 12 7 14 4 3 6 6 10 11 14 4 ^
27 0 4 2 3 10 7 12 13 8 8 12 10 13 1 7 10 2 7 2 13 10 2 3 3 6 4 13 14
^
24 1 11 1 9 14 4 14 14 8 10 13 4 3 2 7 6 6 8 14 5 11 7 14 14 1 ^
22 1 13 13 10 9 13 8 14 9 3 14 3 11 9 9 12 13 2 10 1 5 8 12 ^
25 0 9 10 9 8 4 14 6 9 3 13 10 11 3 1 4 6 10 3 4 3 12 12 14 11 13 ^
26 1 11 8 11 7 8 10 9 12 11 6 10 6 3 9 9 1 7 5 6 7 7 12 4 5 8 11 ^
26 1 12 10 8 14 7 4 13 13 8 4 5 3 2 4 7 14 3 10 6 9 13 9 2 1 9 14 ^
28 0 6 12 2 4 7 8 11 10 12 2 14 3 1 1 1 12 5 13 9 12 5 9 10 5 1 12 10
8 ^
29 1 2 6 5 14 14 3 6 13 1 2 6 2 12 1 2 10 1 4 5 14 10 12 11 2 3 14 14
12 5 ^
30 1 5 2 3 5 11 12 3 7 12 7 9 8 3 3 14 10 5 2 9 3 8 6 11 8 7 8 1 6 12
7 ^
28 1 14 5 2 14 7 8 1 5 4 12 9 6 12 1 1 7 9 5 14 11 9 4 1 10 1 14 10 12
^
27 0 7 2 8 5 3 14 8 9 4 6 11 3 11 6 12 1 7 14 9 14 6 10 6 1 8 14 10 ^
26 1 6 14 2 2 12 10 5 10 10 8 3 7 13 10 1 14 1 14 7 11 12 9 1 8 8 12 ^
30 0 6 13 5 13 3 5 8 8 8 2 5 5 4 2 13 14 7 2 12 12 3 3 5 7 5 11 3 7 13
7 ^
26 0 13 7 2 8 4 10 14 2 7 1 10 8 11 11 2 13 9 11 2 9 11 7 13 11 2 14 ^
31 1 7 7 8 5 13 4 1 12 14 1 9 11 3 5 3 13 11 7 3 2 9 1 3 13 6 3 10 12
1 13 3 ^
23 0 9 8 8 7 14 13 6 13 14 12 13 12 7 4 14 10 1 3 9 7 11 14 5 ^
27 1 1 10 4 12 4 11 5 13 13 4 12 13 7 12 12 4 13 6 2 7 5 2 8 7 13 12 3
^
26 1 7 6 11 12 14 12 3 12 8 7 4 9 14 1 8 11 6 5 10 10 6 8 5 2 11 14 ^

33 1 1 11 1 5 3 8 1 12 14 5 1 3 7 2 3 9 3 4 14 4 4 10 8 5 14 1 11 12
12 10 4 13
2 ^
32 0 11 4 4 9 5 5 8 9 5 10 4 2 4 7 9 9 6 3 5 1 8 3 2 13 3 14 9 8 9 10
14 5 ^
28 0 12 9 5 8 7 2 14 12 3 8 14 6 6 4 7 5 7 10 7 11 10 1 9 6 7 12 8 9 ^
25 0 9 9 2 11 13 12 11 11 6 14 13 10 5 6 8 10 4 3 11 11 14 5 5 10 7 ^
30 0 13 9 14 11 4 8 12 9 5 9 1 2 9 13 12 6 1 14 11 10 7 8 6 1 4 3 2 2
2 13 ^
33 1 13 5 9 6 13 9 1 13 2 6 8 9 1 3 12 13 3 4 4 12 5 2 1 12 1 6 2 1 8
2 14 13 9
^
32 1 8 6 7 1 4 4 6 1 13 3 3 1 13 7 6 11 6 4 9 10 4 8 13 1 9 10 12 7 10
11 10 5 ^
30 0 12 6 6 3 4 13 1 4 11 10 2 6 14 6 9 12 6 2 2 11 9 13 4 7 9 14 2 11
8 7 ^
26 1 2 11 13 2 9 11 8 11 11 15 13 13 3 15 2 12 1 15 3 8 4 9 13 1 13 7
^
32 1 9 6 1 2 2 1 15 3 15 1 10 1 4 11 14 14 6 10 2 5 9 10 10 9 1 2 14
10 12 1 1
15 ^
29 1 13 2 3 8 12 5 14 9 8 9 2 14 1 4 6 15 10 9 2 7 9 4 8 7 9 10 11 11
5 ^
34 1 12 2 7 2 12 8 2 8 6 3 12 4 4 10 1 14 12 11 5 15 3 3 3 14 13 2 2 3
8 2 1 8
8 8 ^
31 0 6 2 7 10 1 13 10 6 6 6 2 8 7 6 7 14 3 12 6 8 9 3 11 1 3 10 12 11
11 9 9 ^
24 1 13 15 4 7 11 13 10 14 14 13 4 11 5 6 11 8 12 13 9 1 10 9 10 7 ^
33 0 3 1 3 3 4 13 1 6 14 8 3 4 3 2 1 14 14 14 5 13 13 11 10 13 4 3 10
11 3 6 1
10 7 ^
28 0 7 9 13 9 15 2 4 12 10 10 4 12 3 8 5 6 13 1 6 15 1 8 9 6 12 11 7
14 ^
32 1 3 1 13 6 4 1 9 4 5 7 1 8 8 11 10 5 15 14 2 8 14 9 7 12 2 4 13 7 4
3 9 14 ^
31 1 14 12 13 8 4 12 2 6 5 12 9 3 11 8 5 13 5 10 5 2 3 7 12 14 4 1 7 3
6 9 9 ^
32 1 10 15 7 2 13 5 7 2 5 2 10 7 11 5 5 9 8 2 9 3 3 6 3 13 10 3 5 7 15
11 10 12
^
38 0 2 3 7 1 6 3 3 12 5 2 9 4 5 11 2 1 7 6 14 7 14 10 4 11 6 3 11 2 10
6 4 8 1
4 9 4 15 4 ^
27 0 7 7 4 14 2 3 14 15 10 7 12 7 5 8 7 15 15 14 10 2 3 1 10 12 15 9 9
^
31 1 15 3 9 9 5 4 2 6 14 4 9 5 12 8 1 6 1 4 8 7 10 14 3 10 14 8 11 9 8
10 9 ^
28 0 14 11 8 3 5 5 7 2 11 4 4 4 6 14 13 10 14 10 3 9 6 9 7 5 10 15 15
15 ^
30 1 8 13 4 2 13 6 4 7 4 4 15 2 2 12 13 3 14 12 7 11 6 2 14 2 6 6 9 14
13 12 ^

27 1 13 14 15 12 14 5 14 15 11 4 1 1 9 5 13 10 6 5 5 6 14 13 1 5 3 14
13 ^
29 0 4 2 13 12 15 7 3 15 5 1 6 7 6 13 1 14 11 9 5 15 6 13 3 6 11 7 15
8 9 ^
31 0 5 4 4 14 3 13 8 13 3 7 13 7 9 12 3 11 15 11 5 14 1 3 4 10 12 7 12
1 2 13 4
^
28 1 12 7 13 14 7 7 1 9 13 8 5 14 2 5 9 9 15 9 13 7 8 9 8 1 12 3 9 15
^
34 1 6 12 4 14 12 5 3 15 4 4 6 8 2 11 8 4 2 5 1 15 15 1 15 2 11 4 10 9
4 7 3 2
15 6 ^
26 0 14 5 6 5 11 12 14 12 12 11 15 2 14 9 15 1 7 12 11 12 13 14 1 1 12
5 ^
32 0 2 2 4 5 2 11 12 5 4 12 9 14 5 5 8 6 2 15 15 6 13 11 3 5 15 5 11 3
12 7 10
8 ^
26 1 2 15 14 9 14 7 6 15 9 3 4 8 8 12 14 2 9 11 14 6 11 15 10 8 14 8 ^
34 1 15 7 4 5 10 1 13 7 1 1 8 3 13 11 2 8 15 11 12 6 6 1 3 2 2 11 3 13
9 10 3 6
12 15 ^
25 0 15 7 10 2 10 15 5 10 13 12 8 15 3 1 11 14 15 8 15 12 9 6 8 13 13
^
27 0 10 8 13 10 15 13 5 8 4 10 10 6 7 13 6 12 12 9 12 2 5 10 12 10 8
15 6 ^
28 0 5 3 8 12 15 8 14 15 9 12 12 3 10 13 6 11 10 4 13 14 8 9 1 8 1 5
11 12 ^
29 0 9 12 4 13 15 5 9 11 7 14 11 1 11 7 8 8 11 1 13 15 12 1 5 8 3 8 10
7 14 ^
30 0 1 5 13 12 14 5 3 6 4 12 15 6 6 10 11 13 9 1 11 6 10 3 7 14 4 14
11 7 8 13 ^
34 1 1 12 5 1 7 4 6 7 5 11 1 4 7 12 10 10 8 15 13 4 11 15 11 6 9 1 2
15 5 1 5
12 6 13 ^
35 1 12 15 4 11 13 3 6 5 10 8 1 3 9 3 11 1 2 16 12 10 6 1 9 1 16 5 6
15 1 12 4
4 2 15 4 ^
27 0 15 11 15 5 11 9 7 15 16 6 16 12 3 2 10 16 5 5 7 1 7 11 13 7 9 11
12 ^
31 1 5 5 5 4 13 13 14 4 7 12 6 4 8 2 9 9 13 13 3 3 6 7 16 7 6 15 5 8
15 6 15 ^
30 1 6 16 15 11 14 14 4 7 10 3 4 10 3 6 7 14 4 6 6 5 2 7 8 16 2 12 16
10 14 7 ^
36 1 2 3 7 14 6 9 4 4 5 9 7 12 3 9 2 5 1 3 10 11 11 6 12 7 1 7 12 2 1
2 8 12 15
16 9 13 ^
34 1 6 3 12 11 16 15 8 8 8 14 13 7 7 8 2 7 6 2 2 14 7 1 4 2 4 11 11 5
7 2 12 1
14 11 ^
29 0 12 11 2 9 8 5 6 14 10 12 5 11 5 3 1 13 15 9 12 11 13 4 14 5 16 3
5 13 15 ^

30 1 5 16 11 2 6 13 16 3 6 4 9 10 9 8 6 14 8 9 5 13 6 5 14 2 8 15 8 7
14 11 ^
28 1 14 10 13 13 11 10 13 9 8 4 11 14 11 6 10 10 1 9 13 16 5 3 4 7 9
12 10 8 ^
36 1 10 15 12 8 8 10 13 5 6 11 4 6 2 3 5 6 11 1 1 8 6 13 15 6 3 15 5
10 1 6 2 5
2 11 10 10 ^
31 1 4 14 7 2 14 3 1 12 7 15 13 1 5 9 8 3 3 14 12 13 2 15 12 7 16 12 3
8 11 5
15 ^
34 0 10 2 10 16 4 16 7 8 1 16 6 14 9 5 12 16 8 9 4 4 4 1 7 13 10 1 7
10 5 7 4 1
5 15 ^
30 0 16 3 8 7 3 8 8 11 8 13 4 16 8 16 1 4 16 11 4 12 1 14 5 1 16 11 8
14 12 9 ^
33 0 7 9 9 16 11 7 6 1 11 14 12 14 10 9 11 8 11 7 5 10 15 2 6 9 3 9 4
2 1 1 9
14 6 ^
32 0 5 8 14 15 4 1 6 1 10 8 1 15 15 3 10 7 8 10 12 4 3 11 15 8 5 3 3
16 15 7 14
13 ^
36 0 3 10 12 2 11 9 2 1 4 3 8 10 14 2 11 3 2 4 15 14 5 2 8 16 16 1 5 7
5 9 5 15
9 11 3 14 ^
31 1 5 4 14 4 8 3 3 1 10 4 15 15 13 2 13 9 7 13 11 9 1 15 15 15 16 7 7
6 5 13 9
^
30 0 14 16 12 1 11 11 7 3 7 3 3 13 4 13 14 5 15 12 5 14 16 14 4 5 9 6
6 3 16 11
^
28 1 10 16 10 14 13 6 16 14 3 14 15 7 13 12 4 1 8 4 10 6 1 13 6 14 11
3 16 14 ^
33 0 15 11 13 8 11 1 15 2 10 9 9 14 8 13 2 13 8 1 2 1 3 12 8 5 4 11 10
14 8 9 4
5 16 ^
34 1 12 12 6 7 13 13 12 1 13 14 1 1 6 3 13 2 16 3 1 16 9 9 15 3 13 1 8
3 2 6 12
8 9 13 ^
27 0 11 12 15 14 12 10 8 15 9 15 2 7 5 11 8 9 7 16 6 15 4 15 7 11 15 4
14 ^
29 1 9 1 14 7 4 14 16 11 10 11 12 14 6 12 11 15 7 14 3 15 6 1 12 5 12
13 3 6 14
^
28 0 6 10 7 7 9 10 7 11 12 16 9 6 9 14 10 10 2 7 4 15 16 13 11 8 10 15
10 15 ^
32 1 8 12 1 10 3 7 15 14 9 16 14 1 15 3 14 9 2 2 12 14 7 3 5 14 5 15
10 2 14 1
15 8 ^
33 0 9 16 10 16 6 13 1 6 2 8 13 5 7 2 15 11 12 6 8 7 1 5 8 13 9 4 8 12
9 15 1
10 13 ^

30 0 10 4 16 4 13 12 2 4 14 6 11 3 15 8 4 12 3 11 16 16 16 2 13 7 10 2
13 10 15
10 ^
33 1 16 2 15 10 4 1 13 12 12 9 12 7 16 4 12 4 2 16 4 9 16 8 2 6 15 2 3
4 9 12 6
8 12 ^
34 1 13 11 11 4 9 13 14 2 9 6 8 6 10 11 9 3 12 9 13 4 10 3 5 1 14 11
11 16 1 2
2 11 6 14 ^
33 1 15 12 6 5 16 2 14 2 10 12 2 5 5 6 10 13 12 3 10 8 16 9 5 12 15 4
11 13 3 6
5 10 8 ^
35 1 1 3 9 3 11 1 2 16 12 10 6 1 9 1 16 5 6 15 1 12 4 4 2 15 13 15 11
15 5 11 9
7 15 16 4 ^
36 0 16 12 3 2 10 16 5 5 7 1 7 11 13 7 9 11 2 5 5 5 4 13 13 14 4 7 12
6 4 8 2 9
9 13 13 4 ^
35 0 3 6 7 16 7 6 15 5 8 15 6 4 6 16 15 11 14 14 4 7 10 3 4 10 3 6 7
14 4 6 6 5
2 7 16 ^
31 0 7 4 8 17 16 16 11 14 2 10 14 15 10 9 1 14 10 14 6 6 16 3 2 3 8 3
12 8 11
17 2 ^
30 1 9 7 16 14 4 11 15 5 13 9 5 12 17 17 13 8 2 6 8 16 1 12 5 17 2 9 8
10 13 6 ^
27 0 11 17 12 5 14 9 11 9 11 4 11 13 16 6 10 5 8 3 17 16 14 1 15 15 15
6 17 ^
29 1 2 11 6 13 11 13 4 6 7 11 11 12 16 13 1 16 16 14 16 2 4 16 11 6 15
7 4 17
11 ^
32 0 5 7 6 3 14 16 5 17 11 13 1 1 14 13 3 6 14 5 17 5 4 8 14 3 16 7 13
4 14 8 9
17 ^
31 1 8 12 8 15 9 6 6 1 5 13 16 5 8 4 6 10 6 11 16 17 15 15 12 4 13 5 3
4 10 15
16 ^
33 0 7 15 16 12 12 11 8 13 14 5 8 1 11 10 9 3 14 3 1 9 6 15 12 16 4 5
9 7 1 9 4
8 17 ^
36 1 7 9 16 11 11 6 13 7 1 7 8 1 14 12 2 7 4 5 15 16 12 3 1 6 11 4 15
8 9 9 2 1
13 2 14 14 ^
29 1 6 8 8 15 13 13 10 2 16 12 17 6 7 14 13 12 13 3 14 12 12 2 15 2 5
15 10 7
15 ^
38 1 5 15 7 8 10 10 1 11 2 4 3 11 11 3 5 2 3 11 5 12 12 8 6 2 17 15 6
16 15 4 3
14 6 3 2 10 5 15 ^
30 0 12 11 15 14 2 16 14 7 6 15 9 3 11 8 8 5 16 2 16 2 16 6 9 6 12 10
16 7 15
10 ^

36 0 11 5 3 16 1 15 7 10 17 10 10 13 2 2 17 6 2 5 6 6 10 12 2 9 11 12
6 1 1 8
12 2 6 16 12 16 ^
30 1 10 17 5 1 6 14 17 8 12 10 11 16 15 1 8 3 9 13 7 7 16 8 13 11 10 8
15 3 15
12 ^
31 1 14 8 4 12 1 6 5 13 15 3 5 11 14 11 14 17 5 10 17 7 8 13 5 12 15 3
6 17 7
17 7 ^
31 0 12 3 10 10 4 13 13 8 11 13 7 17 11 1 10 1 5 11 11 9 14 13 15 8 14
2 7 16
11 8 15 ^
32 1 7 16 17 17 4 1 13 15 5 1 5 6 14 15 1 5 1 9 9 17 12 13 12 7 12 3 7
15 4 16
12 13 ^
36 0 8 17 1 11 4 10 2 6 10 3 16 16 2 7 13 9 15 7 17 14 3 10 16 13 6 7
5 2 1 4
16 3 10 1 15 5 ^
32 0 4 6 4 15 11 4 2 17 9 15 5 10 7 12 15 16 15 17 12 14 9 2 13 4 9 2
4 15 2 11
9 16 ^
29 0 4 4 3 17 2 14 17 17 1 17 13 11 15 17 4 9 15 8 11 16 9 10 8 9 12
15 1 17 11
^
35 0 16 4 2 14 14 15 1 2 12 9 1 10 16 10 11 7 1 12 17 8 2 15 11 1 7 17
12 7 3 7
6 9 9 14 6 ^
36 0 2 15 17 10 11 9 5 9 4 17 16 3 11 2 10 12 11 5 3 8 17 15 3 3 4 7 2
1 10 7
14 4 16 6 12 8 ^
36 1 12 14 6 9 3 17 11 10 6 8 14 6 11 7 10 8 6 17 4 4 9 8 2 8 14 8 7
14 1 3 6 8
11 10 17 1 ^
39 1 12 14 9 4 11 7 6 6 13 14 2 13 1 14 9 7 7 15 2 2 5 15 13 17 3 15
16 9 1 1 2
3 2 12 1 3 3 13 9 ^
34 0 13 10 15 5 8 10 15 10 8 7 5 5 14 4 13 11 12 4 4 12 10 13 12 4 10
3 3 7 9
13 16 9 17 1 ^
36 0 4 12 1 9 12 13 5 10 1 3 15 4 1 13 17 1 8 9 6 12 4 7 7 10 17 13 17
6 17 9 6
9 3 5 17 10 ^
34 0 1 4 10 12 17 14 11 8 5 16 5 5 2 13 13 16 9 13 10 9 9 5 3 13 6 4 3
9 15 14
14 7 3 16 ^
38 1 9 17 3 6 5 16 10 14 9 1 16 7 3 5 4 17 14 7 12 6 2 8 8 17 7 2 4 5
7 12 2 1
7 11 3 14 9 15 ^
41 0 2 12 3 4 1 4 1 16 10 5 16 8 7 2 10 3 5 9 16 8 16 13 3 12 5 5 11 2
2 7 4 2
8 16 4 14 5 14 10 4 17 ^

36 0 6 1 2 7 1 15 13 1 17 4 7 4 8 17 16 16 11 14 2 10 14 15 10 9 1 14
10 14 6 6
16 3 2 3 8 14 ^
32 1 12 8 11 17 3 9 7 16 14 4 11 15 5 13 9 5 12 17 17 13 8 2 6 8 16 1
12 5 17 2
9 14 ^
29 0 10 13 14 11 17 12 5 14 9 11 9 11 4 11 13 16 6 10 5 8 3 17 16 14 1
15 15 15
14 ^
32 0 13 2 11 6 13 11 13 4 6 7 11 11 12 16 13 1 16 16 14 16 2 4 16 11 6
15 7 4
17 6 5 15 ^
35 1 6 3 14 16 5 17 11 13 1 1 14 13 3 6 14 5 17 5 4 8 14 3 16 7 13 4
14 8 9 3 8
12 8 15 11 ^
33 1 6 6 1 5 13 16 5 8 4 6 10 6 11 16 17 15 15 12 4 13 5 3 4 10 15 12
7 15 16
12 12 11 11 ^
40 0 13 14 5 8 1 11 10 9 3 14 3 1 9 6 15 12 16 4 5 9 7 1 9 4 8 5 7 9
16 11 11 6
13 7 1 7 8 1 14 10 ^
34 0 2 10 15 10 14 8 18 9 9 12 12 3 13 12 6 4 9 17 13 13 5 7 3 2 1 17
14 4 16 6
13 1 13 13 ^
35 1 6 10 1 3 18 3 11 7 9 5 7 11 17 1 9 16 5 15 10 17 3 8 15 17 8 15
11 3 15 17
11 1 1 4 15 ^
36 1 14 18 4 2 18 8 15 6 4 6 3 15 11 16 10 17 17 9 6 3 2 6 16 4 9 12 6
8 1 11 3
2 12 2 14 16 ^
37 0 4 1 5 11 16 18 16 8 11 15 2 18 7 5 14 5 15 13 3 3 5 4 3 11 4 4 5
10 3 12
17 1 4 17 10 10 17 ^
35 0 1 3 4 2 1 12 5 14 11 10 4 6 14 9 1 5 8 2 9 15 12 13 7 16 2 13 17
15 9 18
15 9 13 16 17 ^
34 1 18 10 4 11 13 9 6 1 17 6 8 6 5 11 5 14 16 9 13 11 10 9 15 4 14 7
14 3 1 15
5 5 18 16 ^
34 0 13 8 13 16 5 17 5 10 1 8 8 12 6 13 18 9 6 17 3 16 2 8 9 8 7 2 14
14 8 10
13 9 11 11 ^
40 0 7 11 4 6 9 10 11 12 2 14 1 13 1 7 1 15 7 9 8 2 1 4 9 14 17 8 17
15 3 4 15
15 2 2 3 16 13 2 9 12 ^
33 0 14 6 6 5 2 13 4 16 11 3 10 18 2 15 5 9 4 8 4 18 18 17 18 3 14 12
9 11 15
16 1 9 16 ^
32 1 18 5 10 6 7 17 7 12 3 13 8 12 3 8 9 3 17 12 8 18 6 14 3 6 14 14
10 14 13
17 18 8 ^

36 1 18 12 17 9 17 2 12 5 9 12 1 18 10 5 5 2 18 3 6 16 18 3 1 2 10 15
2 5 8 16
10 14 3 9 8 13 ^
33 1 16 14 1 16 15 16 17 4 12 12 16 5 3 9 14 5 7 17 12 11 12 18 14 2 7
7 15 8 3
5 1 8 13 ^
34 1 7 6 7 14 9 15 15 18 16 3 10 5 16 6 8 11 4 1 10 1 16 4 12 17 15 14
8 5 4 6
4 17 16 16 ^
38 0 4 5 18 18 12 9 3 9 6 9 7 6 10 15 4 9 17 5 17 5 6 1 6 2 8 17 7 7 4
4 13 16
3 4 10 16 12 13 ^
33 0 16 7 18 3 3 11 9 13 14 15 18 2 12 10 3 15 3 1 10 6 15 13 17 13 5
9 7 11 9
5 18 15 12 ^
33 1 16 11 1 6 18 13 8 1 18 8 1 5 13 1 18 5 15 15 16 12 13 2 17 12 14
17 8 1 10
12 10 4 18 ^
34 0 5 6 18 18 7 10 13 14 10 12 7 4 10 8 16 16 15 5 13 14 4 14 2 4 12
6 11 5 17
8 5 1 14 16 ^
31 1 17 9 11 2 11 11 3 13 13 11 12 13 5 12 12 11 6 13 4 18 15 12 18 6
6 7 15 13
12 14 16 ^
38 1 4 11 2 5 6 12 10 2 7 15 11 8 14 7 16 16 1 4 3 9 14 16 3 18 1 7 17
1 7 3 16
17 6 16 12 5 3 17 ^
40 1 10 15 18 10 8 10 2 5 11 2 7 3 5 6 7 1 12 2 9 2 4 15 1 11 11 17 3
3 16 8 3
11 18 16 11 6 15 8 8 13 ^
40 1 10 3 17 7 10 18 13 14 8 7 8 14 2 2 8 6 3 7 5 6 9 5 12 10 7 6 5 16
4 14 2
15 3 15 9 15 1 9 7 12 ^
35 0 14 15 12 7 2 16 18 8 18 1 13 12 1 1 14 14 13 3 14 7 2 10 10 11 14
3 12 5
13 6 17 14 1 11 13 ^
34 0 7 12 8 12 17 17 18 17 13 2 4 7 5 1 15 7 14 17 11 6 10 9 18 9 14
13 3 8 4
13 8 7 13 7 ^
35 0 4 16 8 17 10 2 13 10 2 2 6 2 3 18 17 2 17 4 18 7 16 18 14 4 11 7
4 15 17 6
11 11 14 17 4 ^
37 0 10 17 16 13 15 14 6 1 14 11 8 16 6 2 16 13 7 8 6 2 15 1 9 12 4 4
11 13 7 2
11 9 18 4 5 4 18 ^
39 0 2 14 9 9 1 8 13 11 15 8 5 9 10 16 9 2 7 1 1 17 13 6 11 10 8 5 12
15 6 15
10 12 4 18 1 2 8 11 15 ^
32 1 16 10 12 18 11 16 12 11 17 17 4 7 13 7 10 7 10 6 1 12 7 18 11 18
2 10 15
10 14 8 18 2 ^

40 1 9 12 12 3 13 12 6 4 9 17 13 13 5 7 3 2 1 17 14 4 16 6 13 1 13 6 6
10 1 3
18 3 11 7 9 5 7 11 17 12 ^
36 1 9 16 5 15 10 17 3 8 15 17 8 15 11 3 15 17 11 1 1 4 3 14 18 4 2 18
8 15 6 4
6 3 15 11 16 8 ^
39 1 17 17 9 6 3 2 6 16 4 9 12 6 8 1 11 3 2 12 2 14 12 4 1 5 11 16 18
16 8 11
15 2 18 7 5 14 5 15 10 ^
47 1 3 3 5 4 3 11 4 4 5 10 3 12 17 1 4 17 10 10 9 1 3 4 2 1 12 5 14 11
10 4 6
14 9 1 5 8 2 9 15 12 13 7 16 2 13 17 3 ^
34 1 9 18 15 9 13 16 18 10 4 11 13 9 6 1 17 6 8 6 5 11 5 14 16 9 13 11
10 9 15
4 14 7 14 9 ^
37 1 1 15 5 5 18 13 13 8 13 16 5 17 5 10 1 8 8 12 6 13 18 9 6 17 3 16
2 8 9 8 7
2 14 14 8 10 13 ^
42 0 9 11 10 7 11 4 6 9 10 11 12 2 14 1 13 1 7 1 15 7 9 8 2 1 4 9 14
17 8 17 15
3 4 15 15 2 2 3 16 13 2 17 ^
36 0 16 14 6 6 5 2 13 4 16 11 3 10 18 2 15 5 9 4 8 4 18 18 17 18 3 14
12 9 11
15 16 1 9 2 18 6 ^
33 1 10 6 7 17 7 12 3 13 8 12 3 8 9 3 17 12 8 18 6 14 3 6 14 14 10 14
13 17 18
14 18 12 13 ^
39 1 9 17 2 12 5 9 12 1 18 10 5 5 2 18 3 6 16 18 3 1 2 10 15 2 5 8 16
10 14 3 9
8 10 16 14 1 16 15 14 ^
30 1 17 19 5 7 14 4 11 3 6 18 13 7 18 18 19 19 13 9 10 19 14 12 4 15
15 18 2 10
7 15 ^
34 1 2 13 8 19 15 1 11 14 7 7 17 6 16 4 16 17 2 18 12 5 14 19 17 1 8
12 6 6 6
19 13 10 16 5 ^
36 1 14 13 6 16 14 15 4 13 15 4 4 11 4 10 3 17 4 19 6 17 14 3 2 16 1
14 16 5 17
8 1 1 19 13 12 12 ^
34 0 17 11 13 5 19 11 18 19 1 3 12 11 8 4 11 13 17 3 8 9 4 5 12 15 12
6 7 16 16
12 1 13 17 15 ^
40 0 18 9 3 16 2 16 4 19 6 6 18 3 13 1 5 3 2 6 15 11 6 4 14 15 15 9 13
8 1 9 15
2 7 5 11 5 12 7 16 15 ^
37 0 18 4 3 1 4 11 15 7 6 9 1 14 7 1 17 17 16 14 13 14 12 14 9 12 11
18 7 14 3
1 6 1 15 8 15 19 9 ^
32 0 3 9 9 7 7 10 12 13 7 11 12 16 9 9 9 17 13 13 2 17 10 4 18 17 19
16 11 11
13 18 13 12 ^

35 0 8 15 18 10 6 10 18 17 9 2 14 1 17 18 6 14 12 5 5 15 17 10 3 8 8 1
10 5 1
18 18 9 16 10 14 ^
38 1 6 16 1 6 3 5 8 16 8 7 5 18 11 15 17 6 11 10 4 8 14 16 6 7 8 12 12
18 1 10
4 7 1 19 7 19 15 12 ^
34 0 1 14 6 11 18 5 4 12 6 14 2 19 2 5 13 10 10 2 16 10 18 16 19 2 15
10 7 18
16 17 15 15 9 13 ^
38 1 10 16 4 15 7 5 19 1 15 16 11 19 9 18 2 4 6 12 9 8 7 13 14 11 4 12
13 5 9 6
5 6 16 11 9 3 15 6 ^
40 0 16 1 13 6 8 18 11 11 2 1 2 2 14 9 13 1 15 6 5 16 19 14 19 7 18 5
2 2 3 13
13 12 13 11 2 12 2 12 18 5 ^
35 0 7 8 13 6 9 8 17 10 11 1 9 14 4 19 19 3 15 13 9 18 12 17 1 16 8 9
15 1 12 4
4 19 18 13 11 ^
36 1 14 18 8 14 9 3 4 18 16 9 19 12 3 17 5 10 16 5 8 4 4 10 14 13 4 9
11 19 8 8
5 1 13 16 14 13 ^
41 1 10 15 3 4 8 2 12 9 16 3 16 3 6 17 3 10 19 7 6 1 5 11 18 9 4 6 2
15 11 17
17 17 4 10 13 3 7 10 3 9 14 ^
42 0 7 9 3 3 8 19 10 8 2 19 12 19 13 6 19 3 4 17 6 9 1 7 2 9 7 18 6 15
2 2 4 3
10 11 11 6 12 10 18 4 12 10 ^
35 1 4 5 11 12 15 19 12 10 6 15 14 19 18 11 8 11 17 16 7 7 11 2 10 6
19 5 14 7
1 7 3 19 4 17 15 ^
36 1 5 10 2 18 1 17 8 12 17 2 9 8 8 3 14 10 15 2 14 8 3 1 16 18 12 18
11 16 7
17 5 19 2 19 12 19 ^
37 0 16 14 2 9 16 2 6 6 7 9 10 9 11 9 14 11 15 5 16 9 2 17 2 8 15 8 4
3 14 14
16 16 13 11 10 16 14 ^
40 0 11 7 14 14 14 6 10 10 1 6 13 19 5 6 4 7 12 12 10 5 10 15 15 8 5
13 17 13 5
6 14 1 9 2 6 5 17 9 11 13 ^
38 0 18 8 6 13 15 3 3 15 5 13 18 3 2 5 5 14 7 13 4 17 7 2 17 3 18 15 7
15 16 18
5 12 8 6 3 17 12 13 ^
38 1 19 15 9 7 17 16 15 3 11 11 5 2 13 19 16 2 4 16 7 8 1 2 9 17 12 3
5 18 19
11 17 9 1 4 1 18 10 10 ^
41 1 13 1 7 13 5 7 4 1 8 13 16 3 5 7 6 8 8 14 8 13 7 19 8 16 18 4 2 8
7 12 4 14
5 1 3 19 17 11 14 15 19 ^
39 0 10 9 12 19 11 10 3 18 14 17 18 17 7 12 14 8 11 7 2 10 17 15 5 3
12 3 12 1
2 1 1 12 14 10 2 8 17 18 2 ^

45 1 4 9 4 16 5 18 1 1 10 10 11 3 10 12 1 3 11 1 11 5 3 3 16 15 10 17
14 15 6
13 12 2 3 14 3 9 19 4 4 8 10 14 5 11 9 ^
35 0 2 7 15 14 8 2 11 19 19 4 5 10 8 9 5 15 9 17 19 5 7 14 4 11 3 6 18
13 7 18
18 19 19 13 13 ^
35 1 10 19 14 12 4 15 15 18 2 10 7 9 2 13 8 19 15 1 11 14 7 7 17 6 16
4 16 17 2
18 12 5 14 19 9 ^
38 1 1 8 12 6 6 6 19 13 10 16 13 14 13 6 16 14 15 4 13 15 4 4 11 4 10
3 17 4 19
6 17 14 3 2 16 1 14 19 ^
38 0 5 17 8 1 1 19 13 12 12 17 11 13 5 19 11 18 19 1 3 12 11 8 4 11 13
17 3 8 9
4 5 12 15 12 6 7 16 11 ^
43 0 12 1 13 17 4 18 9 3 16 2 16 4 19 6 6 18 3 13 1 5 3 2 6 15 11 6 4
14 15 15
9 13 8 1 9 15 2 7 5 11 5 12 16 ^
39 1 16 6 18 4 3 1 4 11 15 7 6 9 1 14 7 1 17 17 16 14 13 14 12 14 9 12
11 18 7
14 3 1 6 1 15 8 15 19 12 ^
35 0 3 9 9 7 7 10 12 13 7 11 12 16 9 9 9 17 13 13 2 17 10 4 18 17 19
16 11 11
13 18 13 8 8 15 6 ^
40 0 10 6 10 18 17 9 2 14 1 17 18 6 14 12 5 5 15 17 10 3 8 8 1 10 5 1
18 18 9
16 10 19 6 16 1 6 3 5 8 16 ^
40 0 8 7 5 18 11 15 17 6 11 10 4 8 14 16 6 7 8 12 12 18 1 10 4 7 1 19
7 19 15 2
1 14 6 11 18 5 4 12 6 19 ^
35 0 2 19 2 5 13 10 10 2 16 10 18 16 19 2 15 10 7 18 16 17 15 15 9 15
10 16 4
15 7 5 19 1 15 16 6 ^
43 1 19 9 18 2 4 6 12 9 8 7 13 14 11 4 12 13 5 9 6 5 6 16 11 9 3 15 12
16 1 13
6 8 18 11 11 2 1 2 2 14 9 13 11 ^
39 1 15 6 5 16 19 14 19 7 18 5 2 2 3 13 13 12 13 11 2 12 2 12 18 7 8
13 6 9 8
17 10 11 1 9 14 4 19 19 3 ^
34 1 15 13 9 18 12 17 1 16 8 9 15 1 12 4 4 19 18 13 15 14 18 8 14 9 3
4 18 16 9
19 12 3 17 15 ^
45 1 10 16 5 8 4 4 10 14 13 4 9 11 19 8 8 5 1 13 16 14 1 10 15 3 4 8 2
12 9 16
3 16 3 6 17 3 10 19 7 6 1 5 11 18 2 ^
43 1 15 4 13 14 3 4 1 3 1 5 11 4 9 7 20 8 15 4 16 7 19 11 1 11 2 13 13
11 12 13
15 12 1 5 13 3 8 15 12 8 16 16 6 ^
44 0 5 13 12 14 5 3 1 4 16 12 20 1 6 15 11 18 4 17 16 6 10 3 2 9 9 14
6 2 8 6 7
10 17 12 20 6 7 5 16 1 4 2 17 9 ^

38 0 15 8 20 18 20 11 15 11 11 9 6 2 15 5 17 12 1 9 1 3 5 11 7 2 2 6
18 2 9 11
8 16 11 16 15 18 18 18 ^
35 1 20 2 17 17 10 8 13 20 4 18 6 18 9 4 1 1 18 5 2 17 15 16 19 15 1
10 17 16 4
16 14 14 11 15 10 ^
38 0 5 9 15 10 9 17 18 14 15 12 6 1 16 8 18 8 13 12 10 14 14 13 9 2 14
17 9 1
20 10 4 2 12 9 4 13 16 5 ^
36 0 14 10 11 16 17 12 18 12 7 17 8 7 3 1 3 17 4 9 5 17 14 17 11 5 20
19 8 19
14 13 2 18 3 3 18 13 ^
41 1 13 8 14 6 18 7 10 1 13 10 11 11 16 1 2 8 7 1 7 14 8 17 6 8 4 3 11
17 3 5 7
16 11 11 14 7 13 20 20 7 20 ^
40 1 13 15 14 16 14 8 9 16 5 1 6 3 17 18 16 9 1 15 9 10 9 19 1 3 3 20
11 13 17
1 19 8 3 4 3 7 1 14 19 17 ^
37 1 18 13 11 5 18 4 19 10 6 19 11 17 10 10 7 9 13 16 9 10 18 4 12 5
16 5 20 12
3 8 10 1 18 1 6 20 14 ^
36 0 8 9 6 12 11 7 7 3 17 13 6 20 17 9 20 16 10 12 17 8 11 8 11 10 5
10 14 18 8
19 9 12 12 2 20 12 ^
38 0 12 16 20 3 9 9 19 17 13 13 4 17 2 11 7 14 3 6 16 13 10 13 5 16 10
2 8 2 17
19 4 17 7 19 6 9 15 7 ^
41 1 20 7 2 18 5 7 18 5 2 15 7 11 10 9 3 2 14 19 3 11 8 18 15 5 3 5 12
15 16 10
17 7 19 16 2 1 16 6 3 19 10 ^
46 0 5 18 9 9 11 2 1 12 11 14 12 14 10 4 11 6 8 16 7 5 11 20 8 17 4 14
4 15 3 2
2 4 3 2 3 14 15 10 2 12 7 3 7 20 20 5 ^
41 1 10 2 3 1 10 12 15 14 11 20 3 16 14 9 4 18 1 19 20 9 5 12 13 6 6 1
9 13 7
15 9 8 5 19 3 16 9 8 10 20 8 ^
39 1 11 13 19 5 7 16 18 16 4 4 4 6 9 13 10 19 15 3 14 6 4 12 10 15 15
4 8 13 13
9 18 13 6 4 7 20 4 20 7 ^
31 1 18 18 12 16 13 19 19 5 7 2 6 11 18 9 18 6 11 14 14 18 10 13 19 20
17 11 14
5 19 20 13 ^
46 1 20 1 1 4 13 10 6 5 6 19 18 1 3 9 14 20 7 18 7 15 2 3 15 5 1 1 2 6
13 1 19
6 14 16 5 15 11 13 3 6 11 2 20 8 1 20 ^
37 0 4 20 19 3 18 3 8 19 7 8 12 14 12 19 11 15 6 19 1 19 4 10 16 17 7
17 1 2 13
18 12 2 18 19 2 7 15 ^
42 1 9 18 8 5 14 2 5 4 4 20 14 13 7 8 14 8 1 7 19 9 7 11 12 4 19 12 10
19 20 4
4 1 8 2 16 16 8 20 2 1 20 13 ^

37 1 1 15 7 6 20 10 14 9 12 19 18 20 6 14 6 5 6 17 14 12 12 16 4 2 19
14 15 17
7 12 11 12 8 14 1 17 7 ^
39 0 15 18 2 4 10 18 11 17 5 9 17 4 19 5 5 8 6 2 20 15 6 5 18 11 3 10
20 10 16
3 12 7 5 14 18 15 14 9 14 ^
44 0 14 7 6 15 9 3 4 8 8 12 19 2 9 16 19 6 16 20 5 3 19 7 4 7 4 5 10 5
1 18 7
17 6 3 3 13 16 2 3 20 16 12 6 16 ^
38 0 17 19 2 2 11 19 13 4 15 8 1 12 9 20 2 5 2 10 20 5 15 13 7 13 15
19 17 11 9
15 8 15 17 9 6 16 8 13 ^
38 1 18 10 8 8 10 15 13 8 4 5 15 6 12 13 1 17 12 4 7 18 10 12 10 3 15
20 5 19 8
17 20 13 14 10 9 12 17 5 ^
46 1 10 18 6 6 15 4 13 14 3 4 1 3 1 5 11 4 9 7 20 8 15 4 16 7 19 11 1
11 2 13
13 11 12 13 15 12 1 5 13 3 8 15 12 8 16 6 ^
45 1 17 5 13 12 14 5 3 1 4 16 12 20 1 6 15 11 18 4 17 16 6 10 3 2 9 9
14 6 2 8
6 7 10 17 12 20 6 7 5 16 1 4 2 17 16 ^
40 0 15 8 20 18 20 11 15 11 11 9 6 2 15 5 17 12 1 9 1 3 5 11 7 2 2 6
18 2 9 11
8 16 11 16 15 18 18 3 20 19 ^
38 0 17 17 10 8 13 20 4 18 6 18 9 4 1 1 18 5 2 17 15 16 19 15 1 10 17
16 4 16
14 14 11 15 2 5 9 15 10 15 ^
38 0 17 18 14 15 12 6 1 16 8 18 8 13 12 10 14 14 13 9 2 14 17 9 1 20
10 4 2 12
9 4 13 16 7 14 10 11 16 19 ^
39 0 12 18 12 7 17 8 7 3 1 3 17 4 9 5 17 14 17 11 5 20 19 8 19 14 13 2
18 3 3
18 17 13 8 14 6 18 7 10 12 ^
41 0 13 10 11 11 16 1 2 8 7 1 7 14 8 17 6 8 4 3 11 17 3 5 7 16 11 11
14 7 13 20
20 7 16 13 15 14 16 14 8 9 16 ^
43 1 5 1 6 3 17 18 16 9 1 15 9 10 9 19 1 3 3 20 11 13 17 1 19 8 3 4 3
7 1 14 19
19 18 13 11 5 18 4 19 10 6 19 4 ^
41 1 17 10 10 7 9 13 16 9 10 18 4 12 5 16 5 20 12 3 8 10 1 18 1 6 20
12 8 9 6
12 11 7 7 3 17 13 6 20 17 9 15 ^
37 0 16 10 12 17 8 11 8 11 10 5 10 14 18 8 19 9 12 12 2 20 13 12 16 20
3 9 9 19
17 13 13 4 17 2 11 7 16 ^
43 0 3 6 16 13 10 13 5 16 10 2 8 2 17 19 4 17 7 19 6 9 15 15 20 7 2 18
5 7 18 5
2 15 7 11 10 9 3 2 14 19 3 11 14 ^
43 1 18 15 5 3 5 12 15 16 10 17 7 19 16 2 1 16 6 3 19 12 5 18 9 9 11 2
1 12 11
14 12 14 10 4 11 6 8 16 7 5 11 20 2 ^

45 0 17 4 14 4 15 3 2 2 4 3 2 3 14 15 10 2 12 7 3 7 20 20 19 10 2 3 1
10 12 15
14 11 20 3 16 14 9 4 18 1 19 20 9 5 18 ^
42 0 13 6 6 1 9 13 7 15 9 8 5 19 3 16 9 8 10 20 14 11 13 19 5 7 16 18
16 4 4 4
6 9 13 10 19 15 3 14 6 4 12 18 ^
36 1 15 15 4 8 13 13 9 18 13 6 4 7 20 4 20 18 18 12 16 13 19 19 5 7 2
6 11 18 9
18 6 11 14 14 18 15 ^
43 1 13 19 20 17 11 14 5 19 20 11 20 1 1 4 13 10 6 5 6 19 18 1 3 9 14
20 7 18 7
15 2 3 15 5 1 1 2 6 13 1 19 6 19 ^
40 1 16 5 15 11 13 3 6 11 2 20 8 1 5 4 20 19 3 18 3 8 19 7 8 12 14 12
19 11 15
6 19 1 19 4 10 16 17 7 17 16 ^
39 1 12 19 20 1 13 12 10 8 21 15 7 19 13 6 8 19 20 18 2 12 14 3 10 6 2
6 2 17
16 2 5 21 5 17 19 7 8 11 15 ^
33 0 12 17 14 18 15 16 16 16 12 20 13 2 5 5 2 10 20 16 12 10 2 17 21
20 15 14
14 18 21 13 9 17 10 ^
40 1 2 12 16 17 3 9 19 5 7 10 5 11 17 18 14 14 12 2 20 15 8 17 3 20 11
12 18 7
12 7 9 8 3 18 14 3 5 17 9 14 ^
43 0 8 21 11 8 8 16 21 12 5 8 7 20 10 14 7 1 9 5 4 5 2 21 12 1 1 9 20
14 11 9
19 16 3 1 14 10 2 8 20 18 7 8 18 ^
40 1 2 19 6 11 18 11 15 6 5 16 14 9 7 21 3 15 21 16 8 7 3 6 14 17 17 5
10 20 10
15 10 1 15 18 6 10 9 7 16 6 ^
39 0 11 15 12 2 9 1 1 6 6 13 8 20 6 18 8 5 8 1 1 17 20 20 19 17 13 14
17 12 5 3
18 20 20 11 18 15 13 16 7 ^
48 1 7 10 1 19 10 7 17 7 16 3 8 4 4 15 17 13 9 11 2 2 11 7 13 4 17 7 1
5 13 19
16 5 14 16 2 11 18 5 11 6 4 7 3 17 2 1 18 12 ^
41 1 21 18 3 5 16 6 11 2 8 8 7 14 13 21 6 14 12 6 12 19 14 3 16 3 2 7
11 14 8
16 10 19 12 19 4 5 6 13 19 12 13 ^
44 1 7 5 12 4 13 21 2 5 17 15 15 1 13 5 21 6 4 12 14 15 5 18 12 1 19 2
14 1 8 8
11 6 20 10 3 21 8 20 2 4 1 16 11 21 ^
42 1 5 3 8 16 15 8 12 14 20 16 18 7 2 18 2 18 19 7 4 19 3 1 20 7 15 1
4 5 12 10
19 6 11 4 4 9 20 20 8 2 20 18 ^
42 1 19 2 8 4 15 7 2 2 6 18 20 16 1 18 17 13 3 14 2 1 2 10 14 19 12 9
15 20 1 7
15 17 7 12 18 1 15 14 8 21 21 5 ^
43 1 7 20 10 11 10 1 9 21 7 5 1 12 15 2 9 17 4 13 5 4 11 21 14 13 3 20
21 1 3
12 18 11 4 14 20 5 3 14 13 15 9 21 3 ^

40 1 19 8 1 5 9 5 9 16 17 9 6 12 6 16 7 4 3 1 21 16 19 15 18 17 17 17
21 13 20
9 6 6 9 1 13 17 21 1 9 14 ^
36 0 11 12 13 18 19 19 15 12 20 2 16 12 9 21 17 16 8 2 14 6 4 1 6 16
15 19 19
21 9 13 18 3 16 8 13 11 ^
47 0 21 15 4 21 4 2 3 8 15 19 8 6 16 2 15 3 12 3 11 3 17 12 21 6 3 4
13 1 12 4
10 2 6 7 6 9 12 21 10 10 11 2 13 19 2 14 17 ^
44 1 11 8 10 19 1 11 11 9 3 13 15 4 3 18 9 14 5 15 18 5 6 17 11 19 2 4
12 8 18
12 17 3 4 8 5 19 16 16 6 1 13 1 18 18 ^
39 1 10 18 15 11 5 1 17 4 17 19 11 18 21 9 2 10 10 5 5 20 18 13 5 15 3
6 3 10
10 16 13 13 14 16 20 19 11 16 8 ^
42 1 6 8 15 13 11 13 15 18 11 20 17 20 4 16 10 14 1 4 16 4 17 8 3 7 7
18 16 20
2 9 9 4 20 7 16 4 11 4 10 6 20 4 ^
41 0 17 6 19 17 2 15 6 5 7 4 8 11 16 9 10 15 20 12 15 8 6 17 5 9 3 3 4
19 20 12
5 19 21 5 21 11 14 19 1 17 6 ^
39 0 9 4 19 3 17 1 14 21 14 7 6 5 20 14 21 20 4 6 21 7 11 20 12 9 11 6
16 18 18
10 11 20 6 12 11 5 7 21 3 ^
43 1 18 6 15 21 10 4 14 9 19 10 3 3 5 13 1 8 12 3 13 9 7 10 17 10 6 8
3 17 18 3
21 19 6 17 15 4 9 15 9 15 14 4 18 ^
45 1 14 8 10 13 4 11 10 7 6 21 1 14 5 11 7 7 2 13 13 3 9 13 8 14 9 3 7
18 4 9 9
5 15 13 17 10 15 16 20 8 19 9 10 9 13 ^
40 1 19 14 21 2 18 13 10 4 18 16 4 21 15 10 18 19 3 12 12 14 4 13 11 1
11 1 10
2 12 4 21 10 21 18 9 2 16 7 20 7 ^
41 0 7 12 19 20 1 13 12 10 8 21 15 7 19 13 6 8 19 20 18 2 12 14 3 10 6
2 6 2 17
16 2 5 21 5 17 19 7 8 11 10 21 ^
35 0 17 14 18 15 16 16 16 12 20 13 2 5 5 2 10 20 16 12 10 2 17 21 20
15 14 14
18 21 13 9 17 21 2 12 10 ^
40 0 17 3 9 19 5 7 10 5 11 17 18 14 14 12 2 20 15 8 17 3 20 11 12 18 7
12 7 9 8
3 18 14 3 5 17 9 18 8 21 20 ^
46 1 8 8 16 21 12 5 8 7 20 10 14 7 1 9 5 4 5 2 21 12 1 1 9 20 14 11 9
19 16 3 1
14 10 2 8 20 18 7 8 15 2 19 6 11 18 10 ^
44 1 15 6 5 16 14 9 7 21 3 15 21 16 8 7 3 6 14 17 17 5 10 20 10 15 10
1 15 18 6
10 9 7 16 14 11 15 12 2 9 1 1 6 6 19 ^
38 1 8 20 6 18 8 5 8 1 1 17 20 20 19 17 13 14 17 12 5 3 18 20 20 11 18
15 13 16
13 7 10 1 19 10 7 17 7 15 ^

50 0 3 8 4 4 15 17 13 9 11 2 2 11 7 13 4 17 7 1 5 13 19 16 5 14 16 2
11 18 5 11
6 4 7 3 17 2 1 18 13 21 18 3 5 16 6 11 2 8 8 18 ^
42 1 14 13 21 6 14 12 6 12 19 14 3 16 3 2 7 11 14 8 16 10 19 12 19 4 5
6 13 19
12 13 7 5 12 4 13 21 2 5 17 15 15 12 ^
45 1 13 5 21 6 4 12 14 15 5 18 12 1 19 2 14 1 8 8 11 6 20 10 3 21 8 20
2 4 1 16
11 1 5 3 8 16 15 8 12 14 20 16 18 7 18 ^
46 0 18 2 18 19 7 4 19 3 1 20 7 15 1 4 5 12 10 19 6 11 4 4 9 20 20 8 2
20 3 19
2 8 4 15 7 2 2 6 18 20 16 1 18 17 13 14 ^
43 1 14 2 1 2 10 14 19 12 9 15 20 1 7 15 17 7 12 18 1 15 14 8 21 21 19
7 20 10
11 10 1 9 21 7 5 1 12 15 2 9 17 4 19 ^
45 1 5 4 11 21 14 13 3 20 21 1 3 12 18 11 4 14 20 5 3 14 13 15 9 21 11
19 8 1 5
9 5 9 16 17 9 6 12 6 16 7 4 3 1 21 15 ^
35 0 19 15 18 17 17 17 21 13 20 9 6 6 9 1 13 17 21 1 9 16 11 12 13 18
19 19 15
12 20 2 16 12 9 21 12 ^
44 1 16 8 2 14 6 4 1 6 16 15 19 19 21 9 13 18 3 16 8 13 21 15 4 21 4 2
3 8 15
19 8 6 16 2 15 3 12 3 11 3 17 12 21 9 ^
48 0 3 4 13 1 12 4 10 2 6 7 6 9 12 21 10 10 11 2 13 19 2 14 17 11 8 10
19 1 11
11 9 3 13 15 4 3 18 9 14 5 15 18 5 6 17 11 19 15 ^
44 0 4 12 8 18 12 17 3 4 8 5 19 16 16 6 1 13 1 18 1 10 18 15 11 5 1 17
4 17 19
11 18 21 9 2 10 10 5 5 20 18 13 5 15 18 ^
39 1 6 3 10 10 16 13 13 14 16 20 19 11 16 21 6 8 15 13 11 13 15 18 11
20 17 20
4 16 10 14 1 4 16 4 17 8 3 7 21 ^
46 1 18 16 20 2 9 9 4 20 7 16 4 11 4 10 6 20 14 17 6 19 17 2 15 6 5 7
4 8 11 16
9 10 15 20 12 15 8 6 17 5 9 3 3 4 19 3 ^
40 0 12 5 19 21 5 21 11 14 19 1 17 8 9 4 19 3 17 1 14 21 14 7 6 5 20
14 21 20 4
6 21 7 11 20 12 9 11 6 16 11 ^
44 0 18 10 11 20 6 12 11 5 7 21 19 18 6 15 21 10 4 14 9 19 10 3 3 5 13
1 8 12 3
13 9 7 10 17 10 6 8 3 17 18 3 21 19 8 ^
47 0 4 12 22 5 11 20 16 15 4 17 22 1 8 16 11 6 13 18 3 14 8 13 2 19 16
1 19 1 5
3 16 9 8 20 7 5 1 11 15 2 3 17 3 12 7 4 19 ^
42 0 18 16 13 7 1 19 5 12 18 12 6 15 19 4 4 9 13 13 8 17 13 18 12 2 8
4 7 17 17
14 11 13 1 16 11 2 21 3 22 17 16 11 ^
39 0 18 22 12 20 17 13 1 7 10 7 20 11 1 7 8 14 14 12 15 13 15 18 15 11
18 1 17
14 15 20 16 20 8 2 17 10 4 21 2 ^

44 1 19 19 14 22 21 18 13 14 1 3 12 11 11 4 22 13 5 18 7 21 9 22 19 12
8 16 5
17 5 9 1 2 9 6 12 6 1 7 4 3 15 1 14 16 ^
42 1 12 3 10 2 10 14 21 13 17 6 6 17 1 21 2 14 16 17 9 11 20 21 11 12
12 8 20
13 2 9 20 9 14 10 1 16 2 22 6 4 16 18 ^
51 0 15 1 12 4 14 9 21 3 3 9 8 21 15 14 8 4 14 4 2 3 8 12 8 6 1 2 18
20 15 3 19
3 10 20 14 6 3 4 21 1 12 4 18 2 6 7 6 9 20 14 13 ^
40 1 10 19 17 21 12 15 17 7 10 11 8 10 12 1 19 19 9 18 21 4 18 11 9 22
5 15 8
15 3 5 6 2 19 12 17 4 20 8 11 20 ^
45 0 2 18 4 16 20 12 9 9 6 16 21 16 3 16 18 3 19 5 16 2 4 2 12 11 15
11 14 17 2
10 18 5 5 13 3 21 5 3 6 18 18 10 1 21 15 ^
45 1 7 1 13 12 19 1 14 6 8 21 19 21 11 19 13 2 13 4 1 10 22 16 4 9 4
10 16 3 7
15 11 9 13 17 15 9 9 15 4 13 15 9 4 19 9 ^
46 0 18 6 13 22 10 6 12 2 17 15 6 5 7 4 8 11 1 9 10 8 5 20 16 6 10 5
17 3 3 19
12 13 20 5 12 14 5 14 11 22 12 16 10 8 9 16 ^
39 0 15 12 3 2 16 22 14 22 7 6 5 13 22 14 13 4 6 14 7 11 13 20 9 19 6
9 11 11
18 19 13 6 20 19 20 7 22 12 12 ^
41 0 6 14 18 19 7 17 12 10 3 18 15 5 21 16 8 20 3 21 9 15 18 10 15 10
6 8 18 2
11 3 14 4 6 10 19 9 8 17 8 22 20 ^
36 1 7 22 16 18 21 4 11 10 15 6 22 16 22 5 19 15 7 7 2 21 21 3 17 21
16 22 9 3
7 11 4 17 17 20 21 21 ^
37 1 18 9 5 8 12 17 18 17 16 12 22 14 17 11 21 10 4 3 1 4 14 10 11 4 3
20 20 22
19 21 19 16 19 16 10 17 17 ^
40 1 19 14 18 14 3 9 17 1 7 13 6 7 15 20 12 13 16 21 20 10 8 22 7 12
21 6 16 4
13 11 2 12 15 22 3 18 6 17 6 22 ^
42 0 10 9 2 5 22 5 10 4 7 8 19 10 20 10 22 11 8 1 9 1 20 13 21 17 5 5
17 10 13
9 20 18 17 2 14 13 22 22 3 14 21 10 ^
46 0 2 14 2 20 1 10 18 9 12 5 7 18 20 19 10 11 22 7 12 2 13 8 8 2 3 13
19 20 3
7 20 7 9 8 3 11 22 3 5 10 9 11 16 14 19 16 ^
46 1 15 8 1 14 20 5 8 7 13 10 22 7 16 9 5 4 5 2 14 20 1 1 15 17 13 22
19 17 4 1
3 1 22 10 15 15 2 16 13 11 7 16 17 12 6 20 ^
40 1 3 19 8 6 5 9 15 22 17 7 14 18 14 9 16 7 18 14 22 2 2 5 18 13 18 8
18 16 8
11 15 6 18 9 7 9 22 19 20 15 ^
44 0 9 16 16 6 6 21 8 13 6 11 8 5 16 16 1 10 13 13 4 10 21 22 15 10 20
5 3 3 13
15 13 11 11 15 21 1 21 7 10 1 12 18 7 20 ^

51 0 7 1 18 8 4 19 2 21 17 19 2 2 19 7 21 19 2 7 15 15 16 5 21 12 1 5
22 1 17
19 11 5 11 6 4 7 3 10 2 1 3 6 14 11 18 5 1 6 11 17 8 ^
39 1 8 15 22 21 14 6 22 20 6 20 12 22 18 1 3 2 7 19 22 8 9 18 4 20 12
4 5 6 21
12 20 21 7 5 20 4 21 14 14 ^
45 0 15 5 2 16 15 21 5 14 15 6 4 12 22 5 11 20 16 15 4 17 22 1 8 16 11
6 13 18
3 14 8 13 2 19 16 1 19 1 5 3 16 9 8 20 14 ^
48 1 5 1 11 15 2 3 17 3 12 7 4 12 18 16 13 7 1 19 5 12 18 12 6 15 19 4
4 9 13
13 8 17 13 18 12 2 8 4 7 17 17 14 11 13 1 16 11 22 ^
38 0 21 3 22 17 16 17 18 22 12 20 17 13 1 7 10 7 20 11 1 7 8 14 14 12
15 13 15
18 15 11 18 1 17 14 15 20 16 10 ^
43 0 8 2 17 10 4 21 5 19 19 14 22 21 18 13 14 1 3 12 11 11 4 22 13 5
18 7 21 9
22 19 12 8 16 5 17 5 9 1 2 9 6 12 22 ^
47 1 1 7 4 3 15 1 14 1 12 3 10 2 10 14 21 13 17 6 6 17 1 21 2 14 16 17
9 11 20
21 11 12 12 8 20 13 2 9 20 9 14 10 1 16 2 22 20 ^
53 0 4 16 14 15 1 12 4 14 9 21 3 3 9 8 21 15 14 8 4 14 4 2 3 8 12 8 6
1 2 18 20
15 3 19 3 10 20 14 6 3 4 21 1 12 4 18 2 6 7 6 9 20 15 ^
42 1 10 10 19 17 21 12 15 17 7 10 11 8 10 12 1 19 19 9 18 21 4 18 11 9
22 5 15
8 15 3 5 6 2 19 12 17 4 20 8 11 20 12 ^
45 0 18 4 16 20 12 9 9 6 16 21 16 3 16 18 3 19 5 16 2 4 2 12 11 15 11
14 17 2
10 18 5 5 13 3 21 5 3 6 18 18 10 1 21 21 18 ^
45 0 1 13 12 19 1 14 6 8 21 19 21 11 19 13 2 13 4 1 10 22 16 4 9 4 10
16 3 7 15
11 9 13 17 15 9 9 15 4 13 15 9 4 19 19 19 ^
49 0 6 13 22 10 6 12 2 17 15 6 5 7 4 8 11 1 9 10 8 5 20 16 6 10 5 17 3
3 19 12
13 20 5 12 14 5 14 11 22 12 16 10 8 9 4 15 12 3 22 ^
39 0 16 22 14 22 7 6 5 13 22 14 13 4 6 14 7 11 13 20 9 19 6 9 11 11 18
19 13 6
20 19 20 7 22 12 11 6 14 18 17 ^
43 1 7 17 12 10 3 18 15 5 21 16 8 20 3 21 9 15 18 10 15 10 6 8 18 2 11
3 14 4 6
10 19 9 8 17 8 22 4 7 22 16 18 21 11 ^
39 0 11 10 15 6 22 16 22 5 19 15 7 7 2 21 21 3 17 21 16 22 9 3 7 11 4
17 17 20
21 10 18 9 5 8 12 17 18 17 17 ^
40 1 12 22 14 17 11 21 10 4 3 1 4 14 10 11 4 3 20 20 22 19 21 19 16 19
16 10 17
20 19 14 18 14 3 9 17 1 7 13 6 18 ^
44 1 15 20 12 13 16 21 20 10 8 22 7 12 21 6 16 4 13 11 2 12 15 22 3 18
6 17 6 2
10 9 2 5 22 5 10 4 7 8 19 10 20 10 22 7 ^

43 0 8 1 9 1 20 13 21 17 5 5 17 10 13 9 20 18 17 2 14 13 22 22 3 14 21
9 2 14 2
20 1 10 18 9 12 5 7 18 20 19 10 11 19 ^
53 1 7 12 2 13 8 8 2 3 13 19 20 3 7 20 7 9 8 3 11 22 3 5 10 9 11 16 14
19 8 15
8 1 14 20 5 8 7 13 10 22 7 16 9 5 4 5 2 14 20 1 1 15 8 ^
42 0 13 22 19 17 4 1 3 1 22 10 15 15 2 16 13 11 7 16 17 12 6 19 3 19 8
6 5 9 15
22 17 7 14 18 14 9 16 7 18 14 22 19 ^
43 0 2 5 18 13 18 8 18 16 8 11 15 6 18 9 7 9 22 19 20 17 9 16 16 6 6
21 8 13 6
11 8 5 16 16 1 10 13 13 4 10 21 22 14 ^
48 0 10 20 5 3 3 13 15 13 11 11 15 21 1 21 7 10 1 12 18 7 2 7 1 18 8 4
19 2 21
17 19 2 2 19 7 21 19 2 7 15 15 16 5 21 12 1 5 21 ^
49 1 1 17 19 11 5 11 6 4 7 3 10 2 1 3 6 14 11 18 5 1 6 11 17 8 8 15 22
21 14 6
22 20 6 20 12 22 18 1 3 2 7 19 22 8 9 18 4 20 10 ^
45 1 4 5 6 21 12 20 21 7 5 20 4 21 14 2 15 5 2 16 15 21 5 14 15 6 4 12
22 5 11
20 16 15 4 17 22 1 8 16 11 6 13 18 3 14 13 ^
52 0 13 2 19 16 1 19 1 5 3 16 9 8 20 7 5 1 11 15 2 3 17 3 12 7 4 12 18
16 13 7
1 19 5 12 18 12 6 15 19 4 4 9 13 13 8 17 13 18 12 2 8 15 ^
46 0 20 1 1 11 21 2 3 16 8 12 21 22 1 16 1 10 22 17 4 14 7 2 16 9 18
23 9 4 8
16 7 9 18 11 11 17 20 2 7 10 20 8 10 16 14 14 ^
41 1 20 4 16 4 18 1 14 12 19 13 15 19 19 11 22 13 10 2 11 16 10 17 8 8
6 22 2
15 10 9 13 18 14 22 8 6 18 16 15 14 20 ^
43 1 14 3 12 14 17 11 3 9 19 10 20 16 11 3 17 7 21 23 12 23 11 13 2 14
14 16 13
12 11 5 14 3 8 4 13 8 17 17 18 17 2 12 22 ^
43 1 17 3 11 12 16 5 1 22 6 16 11 20 3 18 17 6 22 14 13 21 21 3 18 13
20 11 18
19 11 6 1 10 18 7 17 5 16 1 7 10 4 20 22 ^
46 0 8 10 12 4 11 21 6 13 16 17 6 23 1 9 14 4 11 23 23 19 1 13 6 20 1
9 12 8 5
23 17 16 8 8 3 10 13 7 6 10 8 14 9 15 20 20 ^
46 0 20 21 15 12 19 17 1 6 4 5 8 4 12 10 6 5 4 17 3 3 16 13 5 21 16 10
21 18 19
15 16 12 6 12 17 8 7 8 11 14 1 23 10 15 15 13 ^
43 0 21 13 15 18 21 10 10 23 3 13 13 9 16 2 17 19 3 22 11 5 18 19 13 7
7 8 19 2
12 2 19 16 23 9 16 6 3 19 12 5 10 9 17 ^
47 0 21 3 2 1 7 14 14 7 6 2 20 3 6 19 19 10 2 22 12 17 12 1 20 7 7 15
20 6 18 8
3 14 23 18 15 4 7 5 23 15 7 14 10 10 19 17 11 ^
44 0 4 15 17 11 15 11 8 9 17 5 12 18 14 6 20 17 21 12 16 9 22 9 20 15
2 22 11 2
6 11 9 8 2 4 14 19 3 21 21 23 8 2 11 15 ^

37 0 8 4 20 22 6 21 18 22 10 19 9 14 17 23 21 10 7 15 13 16 5 4 10 13
14 20 23
12 20 23 18 10 12 8 21 11 19 ^
39 0 12 7 19 14 18 14 18 22 6 9 22 5 23 13 6 8 23 20 22 5 22 15 19 20
9 9 1 13
13 10 14 13 5 22 14 21 9 21 3 ^
45 1 14 14 4 18 13 12 23 7 3 15 5 17 14 23 14 5 17 22 11 1 8 13 23 6
21 3 6 11
7 23 8 6 21 4 4 22 19 13 8 5 19 7 5 23 3 ^
50 1 4 19 11 23 11 21 14 1 3 20 2 20 7 4 17 2 5 13 4 23 5 22 7 7 9 17
13 5 6 2
21 1 17 7 9 21 23 1 1 12 3 1 15 22 4 12 6 20 5 23 ^
46 1 23 4 12 22 8 18 16 3 16 4 2 13 1 15 15 1 23 2 3 4 10 1 12 7 11 2
7 22 22
21 22 13 3 12 22 12 4 3 23 10 14 17 15 9 23 20 ^
44 0 19 12 21 22 9 17 12 2 18 18 12 21 2 19 4 21 20 4 9 14 5 21 8 22
10 15 23
14 8 5 11 19 5 7 5 3 11 4 15 18 9 2 23 4 ^
46 0 9 16 14 15 6 15 1 11 4 16 16 20 14 2 9 19 14 6 3 7 10 16 22 23 7
7 12 5 2
8 9 5 7 1 17 19 21 19 18 15 3 20 22 14 9 16 ^
44 1 10 18 11 3 5 9 10 11 22 12 12 15 23 2 10 18 23 5 18 21 20 23 3 9
3 22 7 8
7 4 1 14 16 16 13 18 2 21 18 15 5 5 16 21 ^
46 0 2 2 6 23 13 22 20 12 17 4 10 21 2 20 2 8 8 7 20 13 3 20 23 16 6
23 9 12 20
3 10 5 6 19 18 12 13 14 8 1 1 8 9 13 19 23 ^
47 0 9 20 20 5 15 13 17 3 15 22 3 1 19 8 15 8 19 9 21 7 12 1 16 21 11
8 10 15
16 8 1 5 5 20 22 13 11 22 4 4 7 6 22 10 3 21 4 ^
45 1 9 19 22 18 3 23 6 4 6 19 15 13 17 4 13 9 15 4 6 7 13 11 1 20 15
12 18 18
16 7 13 4 11 7 11 6 17 9 18 15 21 17 5 20 11 ^
49 0 1 22 16 18 5 8 11 2 23 10 11 5 2 1 19 16 3 19 16 15 21 5 3 7 18 4
9 15 3 8
20 1 21 1 21 17 9 6 17 10 16 2 9 7 16 19 15 17 9 ^
51 0 17 20 19 6 6 14 2 2 5 1 2 10 17 1 18 6 18 20 1 1 11 21 2 3 16 8
12 21 22 1
16 1 10 22 17 4 14 7 2 16 9 18 23 9 4 8 16 7 9 18 17 ^
45 1 11 17 20 2 7 10 20 8 10 16 14 11 20 4 16 4 18 1 14 12 19 13 15 19
19 11 22
13 10 2 11 16 10 17 8 8 6 22 2 15 10 9 13 18 8 ^
44 0 22 8 6 18 16 15 14 15 14 3 12 14 17 11 3 9 19 10 20 16 11 3 17 7
21 23 12
23 11 13 2 14 14 16 13 12 11 5 14 3 8 4 13 20 ^
43 0 17 17 18 17 2 12 3 17 3 11 12 16 5 1 22 6 16 11 20 3 18 17 6 22
14 13 21
21 3 18 13 20 11 18 19 11 6 1 10 18 7 17 20 ^
50 1 16 1 7 10 4 20 10 8 10 12 4 11 21 6 13 16 17 6 23 1 9 14 4 11 23
23 19 1
13 6 20 1 9 12 8 5 23 17 16 8 8 3 10 13 7 6 10 8 14 17 ^

46 1 15 20 18 20 21 15 12 19 17 1 6 4 5 8 4 12 10 6 5 4 17 3 3 16 13 5
21 16 10
21 18 19 15 16 12 6 12 17 8 7 8 11 14 1 23 21 ^
45 0 15 15 2 21 13 15 18 21 10 10 23 3 13 13 9 16 2 17 19 3 22 11 5 18
19 13 7
7 8 19 2 12 2 19 16 23 9 16 6 3 19 12 5 10 15 ^
48 1 20 21 3 2 1 7 14 14 7 6 2 20 3 6 19 19 10 2 22 12 17 12 1 20 7 7
15 20 6
18 8 3 14 23 18 15 4 7 5 23 15 7 14 10 10 19 17 12 ^
47 0 4 15 17 11 15 11 8 9 17 5 12 18 14 6 20 17 21 12 16 9 22 9 20 15
2 22 11 2
6 11 9 8 2 4 14 19 3 21 21 23 8 2 11 4 8 4 20 ^
39 0 22 6 21 18 22 10 19 9 14 17 23 21 10 7 15 13 16 5 4 10 13 14 20
23 12 20
23 18 10 12 8 21 11 6 12 7 19 14 14 ^
40 0 14 18 22 6 9 22 5 23 13 6 8 23 20 22 5 22 15 19 20 9 9 1 13 13 10
14 13 5
22 14 21 9 21 19 14 14 4 18 13 12 ^
46 0 23 7 3 15 5 17 14 23 14 5 17 22 11 1 8 13 23 6 21 3 6 11 7 23 8 6
21 4 4
22 19 13 8 5 19 7 5 23 1 4 19 11 23 11 21 9 ^
52 0 1 3 20 2 20 7 4 17 2 5 13 4 23 5 22 7 7 9 17 13 5 6 2 21 1 17 7 9
21 23 1
1 12 3 1 15 22 4 12 6 20 5 19 23 4 12 22 8 18 16 3 22 ^
48 0 4 2 13 1 15 15 1 23 2 3 4 10 1 12 7 11 2 7 22 22 21 22 13 3 12 22
12 4 3
23 10 14 17 15 9 23 4 19 12 21 22 9 17 12 2 18 18 9 ^
46 0 21 2 19 4 21 20 4 9 14 5 21 8 22 10 15 23 14 8 5 11 19 5 7 5 3 11
4 15 18
9 2 23 22 9 16 14 15 6 15 1 11 4 16 16 20 22 ^
47 0 2 9 19 14 6 3 7 10 16 22 23 7 7 12 5 2 8 9 5 7 1 17 19 21 19 18
15 3 20 22
14 9 11 10 18 11 3 5 9 10 11 22 12 12 15 23 22 ^
45 0 10 18 23 5 18 21 20 23 3 9 3 22 7 8 7 4 1 14 16 16 13 18 2 21 18
15 5 5 16
12 2 2 6 23 13 22 20 12 17 4 10 21 2 20 19 ^
47 0 8 8 7 20 13 3 20 23 16 6 23 9 12 20 3 10 5 6 19 18 12 13 14 8 1 1
8 9 13
19 12 9 20 20 5 15 13 17 3 15 22 3 1 19 8 15 23 ^
48 0 19 9 21 7 12 1 16 21 11 8 10 15 16 8 1 5 5 20 22 13 11 22 4 4 7 6
22 10 3
21 1 9 19 22 18 3 23 6 4 6 19 15 13 17 4 13 9 17 ^
49 1 4 6 7 13 11 1 20 15 12 18 18 16 7 13 4 11 7 11 6 17 9 18 15 21 17
5 20 6 1
22 16 18 5 8 11 2 23 10 11 5 2 1 19 16 3 19 16 15 18 ^
55 1 5 3 7 18 4 9 15 3 8 20 1 21 1 21 17 9 6 17 10 16 2 9 7 16 19 15
17 10 17
20 19 6 6 14 2 2 5 1 2 10 17 1 18 6 18 20 1 1 11 21 2 3 16 8 17 ^
47 0 21 22 1 16 1 10 22 17 4 14 7 2 16 9 18 23 9 4 8 16 7 9 18 11 11
17 20 2 7
10 20 8 10 16 14 11 20 4 16 4 18 1 14 12 19 13 19 ^

45 1 19 19 11 22 13 10 2 11 16 10 17 8 8 6 22 2 15 10 9 13 18 14 22 8
6 18 16
15 14 15 14 3 12 14 17 11 3 9 19 10 20 16 11 3 21 ^
47 1 7 21 23 12 23 11 13 2 14 14 16 13 12 11 5 14 3 8 4 13 8 17 17 18
17 2 12 3
17 3 11 12 16 5 1 22 6 16 11 20 3 18 17 6 22 14 20 ^
47 1 21 21 3 18 13 20 11 18 19 11 6 1 10 18 7 17 5 16 1 7 10 4 20 10 8
10 12 4
11 21 6 13 16 17 6 23 1 9 14 4 11 23 23 19 1 13 22 ^
51 0 20 1 9 12 8 5 23 17 16 8 8 3 10 13 7 6 10 8 14 9 15 20 18 20 21
15 12 19
17 1 6 4 5 8 4 12 10 6 5 4 17 3 3 16 13 5 21 16 10 21 21 ^
52 0 15 13 4 8 12 22 9 6 3 1 13 16 12 23 10 14 2 1 6 11 8 19 24 16 2 5
21 2 7
22 4 5 23 3 12 20 8 4 5 17 9 18 18 18 4 10 13 2 4 7 24 21 ^
51 1 10 21 4 8 24 2 6 19 10 6 22 22 14 22 13 21 5 22 2 4 18 5 11 21 1
7 8 4 17
5 14 1 2 10 9 12 5 11 10 20 4 14 3 1 3 21 12 11 17 22 21 ^
43 1 7 5 21 14 15 19 20 12 6 12 18 19 4 7 12 10 5 22 3 15 21 7 23 7 2
22 4 18
18 3 24 10 17 24 23 18 6 11 18 22 8 6 20 ^
45 0 2 12 10 14 12 9 2 23 16 20 14 17 9 16 7 21 6 22 24 19 11 9 19 21
15 11 19
2 2 7 8 10 8 9 8 15 9 14 6 16 8 21 21 22 13 ^
50 1 17 9 4 24 21 15 15 16 16 13 9 10 16 14 12 7 12 15 15 5 10 10 23 5
13 22 3
5 1 4 11 14 7 6 10 17 14 6 3 13 18 16 3 5 15 23 8 5 3 22 ^
47 0 8 12 20 20 9 2 13 24 17 2 2 17 3 13 20 2 6 3 15 7 13 1 18 7 22 18
24 20 14
7 17 16 20 3 11 6 5 2 18 14 16 22 14 11 7 18 22 ^
48 1 17 24 12 12 3 13 19 16 22 4 16 4 6 23 8 18 11 2 3 20 22 9 21 8 23
1 23 20
7 16 13 23 4 13 3 7 4 23 6 13 19 2 3 7 2 9 9 19 ^
45 1 6 13 4 22 6 19 20 1 9 7 14 1 15 3 23 24 22 18 12 12 17 19 10 8 11
22 12 10
2 20 15 18 17 18 7 11 12 21 6 12 4 7 18 17 18 ^
44 0 2 14 24 14 1 23 1 11 15 10 6 18 20 7 1 8 1 16 6 20 23 23 21 10 10
12 24 10
11 23 2 12 23 9 3 24 24 19 14 10 18 15 14 12 ^
50 1 16 11 22 2 15 24 8 22 1 4 24 9 10 15 3 9 5 4 17 15 9 12 19 19 1 3
10 6 8 3
17 8 18 24 19 3 4 15 4 9 2 24 5 20 13 7 20 17 19 11 ^
43 0 13 8 10 19 15 11 1 14 17 20 22 10 7 11 16 9 21 22 17 23 12 15 4
24 7 21 18
2 21 16 1 19 18 20 11 3 15 19 18 1 6 14 15 ^
45 0 5 5 19 13 10 24 19 16 24 15 13 2 19 15 24 21 17 4 13 17 1 1 9 1
10 2 18 1
21 19 5 18 12 2 22 16 23 15 19 6 18 9 1 23 20 ^
49 0 21 13 14 11 18 12 13 3 19 9 20 22 20 2 11 12 6 1 12 16 18 2 9 8 4
3 11 17
11 5 4 19 16 11 23 13 18 1 20 8 2 16 16 21 4 19 5 5 24 ^

46 0 24 13 23 6 24 5 17 12 5 1 15 17 17 21 11 13 9 23 11 14 21 7 6 9 6
14 7 16
2 17 1 24 23 4 12 14 4 2 11 23 12 4 23 21 21 4 ^
49 1 15 13 15 14 15 8 14 9 17 7 15 2 23 2 1 17 6 14 19 24 2 8 8 7 7 10
14 13 7
9 11 16 11 8 8 18 13 13 18 10 1 20 18 22 19 12 9 16 12 ^
43 1 13 6 9 17 20 10 5 7 20 18 8 15 23 21 20 5 15 11 3 3 14 21 10 24 6
21 21 6
23 10 3 21 23 17 20 8 16 10 22 5 16 1 24 ^
50 0 24 3 9 16 6 7 3 20 12 17 18 21 5 9 7 1 9 12 19 5 7 6 24 11 11 20
23 10 4 7
1 19 7 19 17 23 15 5 12 21 20 3 1 14 2 15 22 3 16 11 ^
45 1 10 19 10 20 16 19 17 10 4 21 20 16 21 14 14 8 17 10 19 1 17 4 3
22 23 14
16 9 22 8 20 24 1 5 11 11 6 4 13 15 12 1 11 13 22 ^
45 0 3 8 5 6 5 16 12 11 2 17 14 19 23 16 5 6 20 21 9 24 9 22 23 22 12
8 13 23
17 5 3 16 22 15 22 7 17 3 2 13 13 11 24 13 17 ^
42 1 21 11 14 20 24 24 4 9 13 5 8 9 18 7 12 1 21 8 24 15 1 22 19 24 14
13 8 20
14 18 23 19 9 8 14 23 11 4 4 22 20 17 ^
49 1 17 15 20 6 15 8 24 4 20 19 8 19 14 21 24 18 3 10 16 3 6 4 1 10 15
13 4 8
12 22 9 6 3 1 13 16 12 23 10 14 2 1 6 11 8 19 24 16 23 ^
50 0 5 21 2 7 22 4 5 23 3 12 20 8 4 5 17 9 18 18 18 4 10 13 2 4 7 24 8
10 21 4
8 24 2 6 19 10 6 22 22 14 22 13 21 5 22 2 4 18 5 24 ^
53 1 21 1 7 8 4 17 5 14 1 2 10 9 12 5 11 10 20 4 14 3 1 3 21 12 11 17
22 11 7 5
21 14 15 19 20 12 6 12 18 19 4 7 12 10 5 22 3 15 21 7 23 7 18 ^
43 1 22 4 18 18 3 24 10 17 24 23 18 6 11 18 22 8 6 6 2 12 10 14 12 9 2
23 16 20
14 17 9 16 7 21 6 22 24 19 11 9 19 21 6 ^
49 1 11 19 2 2 7 8 10 8 9 8 15 9 14 6 16 8 21 21 22 6 17 9 4 24 21 15
15 16 16
13 9 10 16 14 12 7 12 15 15 5 10 10 23 5 13 22 3 5 22 ^
52 0 4 11 14 7 6 10 17 14 6 3 13 18 16 3 5 15 23 8 5 3 18 8 12 20 20 9
2 13 24
17 2 2 17 3 13 20 2 6 3 15 7 13 1 18 7 22 18 24 20 14 7 23 ^
45 1 16 20 3 11 6 5 2 18 14 16 22 14 11 7 18 16 17 24 12 12 3 13 19 16
22 4 16
4 6 23 8 18 11 2 3 20 22 9 21 8 23 1 23 20 23 ^
51 1 16 13 23 4 13 3 7 4 23 6 13 19 2 3 7 2 9 9 15 6 13 4 22 6 19 20 1
9 7 14 1
15 3 23 24 22 18 12 12 17 19 10 8 11 22 12 10 2 20 15 15 ^
48 1 17 18 7 11 12 21 6 12 4 7 18 17 3 2 14 24 14 1 23 1 11 15 10 6 18
20 7 1 8
1 16 6 20 23 23 21 10 10 12 24 10 11 23 2 12 23 9 20 ^
49 1 24 24 19 14 10 18 15 14 5 16 11 22 2 15 24 8 22 1 4 24 9 10 15 3
9 5 4 17
15 9 12 19 19 1 3 10 6 8 3 17 8 18 24 19 3 4 15 4 24 ^

43 0 2 24 5 20 13 7 20 17 19 22 13 8 10 19 15 11 1 14 17 20 22 10 7 11
16 9 21
22 17 23 12 15 4 24 7 21 18 2 21 16 1 19 11 ^
48 1 20 11 3 15 19 18 1 6 14 5 5 5 19 13 10 24 19 16 24 15 13 2 19 15
24 21 17
4 13 17 1 1 9 1 10 2 18 1 21 19 5 18 12 2 22 16 23 19 ^
50 1 19 6 18 9 1 23 22 21 13 14 11 18 12 13 3 19 9 20 22 20 2 11 12 6
1 12 16
18 2 9 8 4 3 11 17 11 5 4 19 16 11 23 13 18 1 20 8 2 16 16 ^
48 1 16 21 4 19 5 5 20 24 13 23 6 24 5 17 12 5 1 15 17 17 21 11 13 9
23 11 14
21 7 6 9 6 14 7 16 2 17 1 24 23 4 12 14 4 2 11 23 15 ^
50 1 4 23 21 21 16 15 13 15 14 15 8 14 9 17 7 15 2 23 2 1 17 6 14 19
24 2 8 8 7
7 10 14 13 7 9 11 16 11 8 8 18 13 13 18 10 1 20 18 22 3 ^
44 1 12 9 16 17 13 6 9 17 20 10 5 7 20 18 8 15 23 21 20 5 15 11 3 3 14
21 10 24
6 21 21 6 23 10 3 21 23 17 20 8 16 10 22 12 ^
53 0 16 1 5 24 3 9 16 6 7 3 20 12 17 18 21 5 9 7 1 9 12 19 5 7 6 24 11
11 20 23
10 4 7 1 19 7 19 17 23 15 5 12 21 20 3 1 14 2 15 22 3 16 9 ^
46 1 10 19 10 20 16 19 17 10 4 21 20 16 21 14 14 8 17 10 19 1 17 4 3
22 23 14
16 9 22 8 20 24 1 5 11 11 6 4 13 15 12 1 11 13 21 21 ^
46 0 8 5 6 5 16 12 11 2 17 14 19 23 16 5 6 20 21 9 24 9 22 23 22 12 8
13 23 17
5 3 16 22 15 22 7 17 3 2 13 13 11 24 13 12 21 7 ^
43 1 14 20 24 24 4 9 13 5 8 9 18 7 12 1 21 8 24 15 1 22 19 24 14 13 8
20 14 18
23 19 9 8 14 23 11 4 4 22 20 13 17 15 24 ^
55 0 6 15 8 24 4 20 19 8 19 14 21 24 18 3 10 16 3 6 4 1 10 15 13 4 8
12 22 9 6
3 1 13 16 12 23 10 14 2 1 6 11 8 19 24 16 2 5 21 2 7 22 4 5 23 4 ^
55 1 12 20 8 4 5 17 9 18 18 18 4 10 13 2 4 7 24 8 10 21 4 8 24 2 6 19
10 6 22
22 14 22 13 21 5 22 2 4 18 5 11 21 1 7 8 4 17 5 14 1 2 10 9 12 14 ^
49 0 11 10 20 4 14 3 1 3 21 12 11 17 22 11 7 5 21 14 15 19 20 12 6 12
18 19 4 7
12 10 5 22 3 15 21 7 23 7 2 22 4 18 18 3 24 10 17 24 12 ^
50 0 18 6 11 18 22 8 6 6 2 12 10 14 12 9 2 23 16 20 14 17 9 16 7 21 6
22 24 19
11 9 19 21 15 11 19 2 2 7 8 10 8 9 8 15 9 14 6 16 8 22 ^
51 1 21 22 6 17 9 4 24 21 15 15 16 16 13 9 10 16 14 12 7 12 15 15 5 10
10 23 5
13 22 3 5 1 4 11 14 7 6 10 17 14 6 3 13 18 16 3 5 15 23 8 21 ^
51 1 5 3 18 8 12 20 20 9 2 13 24 17 2 2 17 3 13 20 2 6 3 15 7 13 1 18
7 22 18
24 20 14 7 17 16 20 3 11 6 5 2 18 14 16 22 14 11 7 18 16 20 ^
52 0 24 12 12 3 13 19 16 22 4 16 4 6 23 8 18 11 2 3 20 22 9 21 8 23 1
23 20 7
16 13 23 4 13 3 7 4 23 6 13 19 2 3 7 2 9 9 15 6 13 4 22 16 ^

49 1 19 20 1 9 7 14 1 15 3 23 24 22 18 12 12 17 19 10 8 11 22 12 10 2
20 15 18
17 18 7 11 12 21 6 12 4 7 18 17 3 2 14 24 14 1 23 1 11 16 ^
47 0 10 6 18 20 7 1 8 1 16 6 20 23 23 21 10 10 12 24 10 11 23 2 12 23
9 3 24 24
19 14 10 18 15 14 5 16 11 22 2 15 24 8 22 1 4 24 3 ^
45 0 17 6 24 11 7 21 9 17 8 25 5 5 22 16 13 5 2 16 4 25 25 8 18 13 13
3 1 22 23
24 18 12 21 19 2 4 24 13 11 2 7 24 22 17 21 ^
46 1 15 1 8 3 10 15 18 15 8 10 16 20 20 7 18 22 11 18 9 11 15 22 20 18
6 24 15
25 4 22 12 17 4 6 17 20 2 4 2 5 20 15 19 24 18 15 ^
52 1 8 2 5 23 2 10 21 8 5 2 9 16 16 15 6 22 23 5 6 21 5 21 24 3 6 10
17 19 9 15
11 23 16 17 1 3 12 3 20 4 5 5 3 21 22 15 4 21 20 7 22 18 ^
43 0 9 24 4 18 7 7 25 24 22 21 23 24 14 20 12 6 7 4 2 12 25 15 3 9 23
8 16 25
11 14 4 10 19 23 16 13 9 20 25 17 11 8 19 ^
55 0 19 25 17 7 16 21 6 4 8 2 15 9 2 9 19 3 6 3 3 10 25 13 15 7 8 20
21 12 10
12 5 24 11 20 3 13 13 16 9 13 10 3 9 16 3 7 15 9 9 14 2 13 17 25 3 ^
46 0 18 5 19 23 9 25 9 10 23 12 12 7 13 8 15 7 1 6 21 2 8 7 6 16 14 14
12 15 13
24 10 15 11 10 8 14 15 21 25 21 16 9 18 20 22 21 ^
50 1 20 11 14 23 22 10 13 14 8 19 12 2 11 20 23 13 4 10 6 5 7 23 11 3
16 8 21 4
8 18 5 12 14 8 6 20 19 24 8 23 17 3 7 19 1 18 1 14 22 11 ^
44 1 14 5 8 22 18 14 25 17 11 12 22 2 12 12 16 12 13 18 17 12 17 14 18
8 25 9
23 5 3 8 14 24 17 7 3 3 23 17 22 19 19 17 16 19 ^
51 0 24 14 16 20 23 20 9 19 16 7 12 16 5 8 9 7 10 21 24 10 11 19 1 21
14 14 19
3 22 8 12 20 1 18 5 6 5 12 14 1 1 11 9 22 3 24 9 6 1 11 20 ^
47 0 1 12 8 11 11 17 10 22 7 3 10 2 6 4 24 16 24 19 4 5 18 11 12 9 20
21 25 2
21 18 10 20 25 21 3 17 17 5 8 22 25 19 8 10 19 7 25 ^
55 1 4 8 18 3 18 3 11 24 2 14 9 1 10 4 4 13 19 22 4 7 19 5 10 10 1 8
22 22 6 8
13 12 11 17 23 14 16 7 5 11 22 24 7 2 11 13 18 4 20 8 1 6 25 20 16 ^
51 1 20 1 25 10 15 15 2 17 13 9 24 17 24 20 13 15 17 3 11 20 10 5 13
23 18 5 3
18 15 7 5 7 4 15 20 16 2 17 1 7 4 12 7 25 14 1 4 7 19 19 22 ^
52 0 3 13 12 4 3 14 4 14 11 6 3 25 18 16 20 24 12 19 18 25 6 23 6 15 5
4 4 6 14
13 20 5 14 15 2 13 21 11 5 17 2 2 10 11 13 14 12 11 21 15 23 20 ^
50 1 2 5 7 23 10 20 6 12 1 19 10 24 21 22 25 16 12 19 22 15 10 18 24 1
5 16 10
18 2 19 14 3 5 1 7 20 7 16 6 20 6 6 25 10 22 20 2 1 15 18 ^
48 0 23 25 23 4 11 15 15 6 9 22 22 19 22 25 6 19 6 2 20 21 7 22 18 10
7 17 4 16
12 10 25 12 23 23 9 2 22 1 20 2 8 3 14 7 1 2 3 24 ^

38 1 21 21 23 20 22 11 25 20 24 25 1 9 21 20 14 10 14 19 9 17 15 13 5
15 24 21
12 13 25 22 21 1 16 23 7 24 15 22 ^
54 1 22 5 25 3 4 2 9 12 24 6 19 1 14 9 5 9 1 17 9 4 23 21 1 3 15 16 11
1 12 18
6 2 7 8 17 8 21 13 22 9 21 4 9 1 13 17 21 10 9 18 24 23 13 24 ^
43 1 23 23 19 23 22 17 18 23 18 21 17 1 10 12 4 6 1 21 11 18 2 6 6 12
18 13 16
16 18 19 13 11 21 19 15 12 15 16 19 2 23 19 23 ^
48 0 1 2 25 14 11 25 24 16 8 14 21 21 16 6 13 1 23 15 8 17 4 3 4 9 23
12 8 8 24
13 6 11 3 8 24 19 8 23 1 24 24 18 25 13 2 6 25 14 ^
51 1 9 3 5 11 19 11 16 5 4 17 24 23 15 14 10 7 14 17 25 15 10 14 6 18
18 4 1 13
10 16 1 25 16 2 24 5 1 17 6 17 6 24 11 7 21 9 17 8 25 5 23 ^
48 1 22 16 13 5 2 16 4 25 25 8 18 13 13 3 1 22 23 24 18 12 21 19 2 4
24 13 11 2
7 24 22 17 5 15 1 8 3 10 15 18 15 8 10 16 20 20 7 25 ^
49 0 22 11 18 9 11 15 22 20 18 6 24 15 25 4 22 12 17 4 6 17 20 2 4 2 5
20 15 19
24 18 9 8 2 5 23 2 10 21 8 5 2 9 16 16 15 6 22 23 17 ^
47 1 6 21 5 21 24 3 6 10 17 19 9 15 11 23 16 17 1 3 12 3 20 4 5 5 3 21
22 15 4
21 20 7 22 14 9 24 4 18 7 7 25 24 22 21 23 24 14 ^
53 0 20 12 6 7 4 2 12 25 15 3 9 23 8 16 25 11 14 4 10 19 23 16 13 9 20
25 17 11
8 4 19 25 17 7 16 21 6 4 8 2 15 9 2 9 19 3 6 3 3 10 25 13 15 ^
54 1 7 8 20 21 12 10 12 5 24 11 20 3 13 13 16 9 13 10 3 9 16 3 7 15 9
9 14 2 13
17 25 2 18 5 19 23 9 25 9 10 23 12 12 7 13 8 15 7 1 6 21 2 8 25 ^
44 1 6 16 14 14 12 15 13 24 10 15 11 10 8 14 15 21 25 21 16 9 18 20 22
21 20 11
14 23 22 10 13 14 8 19 12 2 11 20 23 13 4 10 6 25 ^
49 1 7 23 11 3 16 8 21 4 8 18 5 12 14 8 6 20 19 24 8 23 17 3 7 19 1 18
1 14 22
13 14 5 8 22 18 14 25 17 11 12 22 2 12 12 16 12 13 18 25 ^
45 1 12 17 14 18 8 25 9 23 5 3 8 14 24 17 7 3 3 23 17 22 19 19 17 16
22 24 14
16 20 23 20 9 19 16 7 12 16 5 8 9 7 10 21 24 7 ^
58 0 11 19 1 21 14 14 19 3 22 8 12 20 1 18 5 6 5 12 14 1 1 11 9 22 3
24 9 6 1
11 16 1 12 8 11 11 17 10 22 7 3 10 2 6 4 24 16 24 19 4 5 18 11 12 9 20
21 7 ^
54 1 2 21 18 10 20 25 21 3 17 17 5 8 22 25 19 8 10 19 7 11 4 8 18 3 18
3 11 24
2 14 9 1 10 4 4 13 19 22 4 7 19 5 10 10 1 8 22 22 6 8 13 12 11 21 ^
47 1 23 14 16 7 5 11 22 24 7 2 11 13 18 4 20 8 1 6 25 20 17 20 1 25 10
15 15 2
17 13 9 24 17 24 20 13 15 17 3 11 20 10 5 13 23 18 21 ^
57 1 3 18 15 7 5 7 4 15 20 16 2 17 1 7 4 12 7 25 14 1 4 7 19 19 20 3
13 12 4 3

14 4 14 11 6 3 25 18 16 20 24 12 19 18 25 6 23 6 15 5 4 4 6 14 13 20 7
^
51 1 14 15 2 13 21 11 5 17 2 2 10 11 13 14 12 11 21 15 23 3 2 5 7 23
10 20 6 12
1 19 10 24 21 22 25 16 12 19 22 15 10 18 24 1 5 16 10 18 2 19 8 ^
49 0 3 5 1 7 20 7 16 6 20 6 6 25 10 22 20 2 1 15 14 23 25 23 4 11 15
15 6 9 22
22 19 22 25 6 19 6 2 20 21 7 22 18 10 7 17 4 16 12 24 ^
46 0 10 25 12 23 23 9 2 22 1 20 2 8 3 14 7 1 2 3 19 21 21 23 20 22 11
25 20 24
25 1 9 21 20 14 10 14 19 9 17 15 13 5 15 24 21 14 ^
55 0 13 25 22 21 1 16 23 7 24 15 3 22 5 25 3 4 2 9 12 24 6 19 1 14 9 5
9 1 17 9
4 23 21 1 3 15 16 11 1 12 18 6 2 7 8 17 8 21 13 22 9 21 4 9 22 ^
45 1 13 17 21 10 9 18 24 23 13 7 23 23 19 23 22 17 18 23 18 21 17 1 10
12 4 6 1
21 11 18 2 6 6 12 18 13 16 16 18 19 13 11 21 19 8 ^
50 1 12 15 16 19 2 23 19 4 1 2 25 14 11 25 24 16 8 14 21 21 16 6 13 1
23 15 8
17 4 3 4 9 23 12 8 8 24 13 6 11 3 8 24 19 8 23 1 24 24 12 ^
53 0 25 13 2 6 25 7 9 3 5 11 19 11 16 5 4 17 24 23 15 14 10 7 14 17 25
15 10 14
6 18 18 4 1 13 10 16 1 25 16 2 24 5 1 17 6 17 6 24 11 7 21 9 19 ^
51 1 8 25 5 5 22 16 13 5 2 16 4 25 25 8 18 13 13 3 1 22 23 24 18 12 21
19 2 4
24 13 11 2 7 24 22 17 5 15 1 8 3 10 15 18 15 8 10 16 20 20 8 ^
50 1 18 22 11 18 9 11 15 22 20 18 6 24 15 25 4 22 12 17 4 6 17 20 2 4
2 5 20 15
19 24 18 9 8 2 5 23 2 10 21 8 5 2 9 16 16 15 6 22 23 18 ^
48 1 6 21 5 21 24 3 6 10 17 19 9 15 11 23 16 17 1 3 12 3 20 4 5 5 3 21
22 15 4
21 20 7 22 14 9 24 4 18 7 7 25 24 22 21 23 24 14 19 ^
55 0 12 6 7 4 2 12 25 15 3 9 23 8 16 25 11 14 4 10 19 23 16 13 9 20 25
17 11 8
4 19 25 17 7 16 21 6 4 8 2 15 9 2 9 19 3 6 3 3 10 25 13 15 7 8 24 ^
56 0 21 12 10 12 5 24 11 20 3 13 13 16 9 13 10 3 9 16 3 7 15 9 9 14 2
13 17 25
2 18 5 19 23 9 25 9 10 23 12 12 7 13 8 15 7 1 6 21 2 8 7 6 16 14 14 22
^
47 0 15 13 24 10 15 11 10 8 14 15 21 25 21 16 9 18 20 22 21 20 11 14
23 22 10
13 14 8 19 12 2 11 20 23 13 4 10 6 5 7 23 11 3 16 8 21 12 ^
49 0 4 8 18 5 12 14 8 6 20 19 24 8 23 17 3 7 19 1 18 1 14 22 13 14 5 8
22 18 14
25 17 11 12 22 2 12 12 16 12 13 18 17 12 17 14 18 8 25 22 ^
46 1 23 5 3 8 14 24 17 7 3 3 23 17 22 19 19 17 16 22 24 14 16 20 23 20
9 19 16
7 12 16 5 8 9 7 10 21 24 10 11 19 1 21 14 14 19 20 ^
57 1 22 8 12 20 1 18 5 6 5 12 14 1 1 11 9 22 3 24 9 6 1 11 16 1 12 8
11 11 17
10 22 7 3 10 2 6 4 24 16 24 19 4 5 18 11 12 9 20 21 25 2 21 18 10 20
25 7 ^

56 1 3 17 17 5 8 22 25 19 8 10 19 7 11 4 8 18 3 18 3 11 24 2 14 9 1 10
4 4 13
19 22 4 7 19 5 10 10 1 8 22 22 6 8 13 12 11 17 23 14 16 7 5 11 22 24
18 ^
54 1 2 11 13 18 4 20 8 1 6 25 20 17 20 1 25 10 15 15 2 17 13 9 24 17
24 20 13
15 17 3 11 20 10 5 13 23 18 5 3 18 15 7 5 7 4 15 20 16 2 17 1 7 4 23 ^
56 0 7 25 14 1 4 7 19 19 20 3 13 12 4 3 14 4 14 11 6 3 25 18 16 20 24
12 19 18
25 6 23 6 15 5 4 4 6 14 13 20 5 14 15 2 13 21 11 5 17 2 2 10 11 13 14
24 ^
49 0 12 11 20 3 14 2 3 2 23 12 12 17 24 11 8 6 24 16 15 22 21 14 6 12
20 19 5 5
12 11 6 23 2 16 23 7 24 6 21 2 17 17 5 25 11 25 20 25 19 ^
54 0 18 6 12 19 25 7 6 5 2 14 12 16 1 15 7 1 26 19 19 13 20 11 17 6 20
5 24 24
1 21 11 9 20 21 15 10 19 26 3 2 6 7 12 9 10 8 14 10 15 5 17 8 21 7 ^
56 0 20 25 6 19 8 3 22 16 16 16 17 13 11 10 17 15 12 6 13 14 17 4 12
10 24 5 13
24 3 5 2 5 11 14 8 5 10 17 16 8 4 14 21 15 3 6 17 25 8 2 3 3 19 10 13
23 ^
51 1 22 8 2 13 25 17 2 1 19 1 14 20 2 5 4 15 8 14 20 7 25 20 26 20 16
7 17 17
22 1 13 6 5 1 18 14 15 23 15 10 5 19 18 18 26 12 13 3 25 12 18 ^
51 1 16 24 4 16 3 6 26 26 10 20 13 1 20 24 10 21 7 25 2 25 22 9 15 16
26 5 12 1
7 4 24 7 14 20 1 4 7 3 10 10 15 6 13 6 22 6 20 22 2 26 26 ^
47 1 6 14 1 16 3 24 25 24 17 12 14 18 18 9 9 11 22 13 9 2 22 15 20 20
18 8 13
13 21 8 11 5 26 8 19 17 4 2 15 26 13 24 26 11 16 9 24 ^
52 0 8 18 21 6 1 8 1 16 4 20 23 23 24 10 11 12 10 10 25 1 11 25 10 3
26 26 20
15 11 19 16 17 3 16 12 25 15 9 23 1 2 24 8 10 18 2 9 5 4 17 16 12 ^
50 0 12 20 20 1 3 11 5 9 3 17 9 17 24 21 3 4 16 2 10 1 25 5 22 16 6 21
19 21 23
15 9 9 21 17 11 1 15 17 23 24 11 7 10 26 15 8 22 22 18 16 ^
48 0 11 17 5 26 7 23 19 3 24 17 2 19 18 26 21 12 5 18 22 20 7 15 6 4 5
20 13 10
23 18 16 25 18 15 2 20 14 24 22 17 3 15 16 1 10 2 12 16 ^
47 0 21 1 21 21 4 18 13 1 24 18 23 17 19 6 19 9 26 25 23 23 14 13 13
20 12 13 4
21 10 22 21 22 1 12 12 7 13 16 18 8 9 4 3 10 18 12 25 ^
47 1 5 19 19 12 24 13 18 1 22 8 1 17 18 24 3 22 7 7 19 26 13 26 6 26 6
18 12 5
26 17 18 19 22 10 12 8 23 11 17 23 25 5 5 11 4 13 20 ^
52 0 18 4 19 23 3 13 15 4 3 9 24 14 5 25 21 22 18 17 12 15 14 16 10 14
9 19 7
16 3 23 4 26 18 5 16 21 24 2 8 8 8 7 11 13 14 7 9 12 18 11 10 20 ^
47 1 20 12 14 19 11 21 21 22 19 13 10 16 19 14 7 8 16 20 10 3 9 23 18
7 26 16
22 21 21 5 14 11 2 2 16 22 11 26 7 22 23 5 23 10 4 22 5 ^

50 0 19 20 9 16 10 24 4 19 26 4 25 4 8 17 7 7 2 23 11 18 19 23 6 10 9
1 9 14 19
5 6 6 25 12 11 21 26 12 3 8 20 8 21 18 25 25 14 4 11 25 ^
45 1 23 4 14 3 17 26 22 26 2 15 9 8 25 19 12 23 17 20 17 10 4 23 20 17
21 14 14
9 18 11 18 2 18 4 2 22 25 15 18 10 23 8 21 26 15 ^
54 0 7 14 10 6 6 15 15 13 2 11 15 22 2 9 4 4 6 16 11 11 3 16 13 19 25
16 5 7 20
22 9 25 11 24 24 25 13 8 15 23 16 6 3 18 23 16 23 8 20 4 1 1 14 16 ^
43 1 12 24 14 14 26 23 13 26 14 21 25 25 4 9 13 5 8 9 21 8 12 26 24 9
24 15 1
23 22 16 14 8 22 15 19 24 20 7 8 15 24 12 18 ^
57 0 4 23 21 13 19 15 21 7 15 7 3 21 20 8 22 14 23 26 19 2 10 18 3 5 3
1 9 15
15 3 7 13 23 9 7 1 13 17 14 25 9 16 2 2 6 13 7 19 25 17 1 5 21 2 7 22
5 ^
55 1 6 25 3 12 19 6 2 4 24 17 9 18 20 19 4 11 14 1 6 8 26 6 9 22 4 10
2 7 21 9
8 24 25 14 22 12 22 3 23 3 3 20 6 11 23 6 1 7 5 18 5 15 25 26 23 ^
53 0 1 10 11 11 4 12 11 20 3 14 2 3 2 23 12 12 17 24 11 8 6 24 16 15
22 21 14 6
12 20 19 5 5 12 11 6 23 2 16 23 7 24 6 21 2 17 17 5 25 11 25 20 26 ^
54 0 24 18 6 12 19 25 7 6 5 2 14 12 16 1 15 7 1 26 19 19 13 20 11 17 6
20 5 24
24 1 21 11 9 20 21 15 10 19 26 3 2 6 7 12 9 10 8 14 10 15 5 17 8 23 ^
58 0 1 20 25 6 19 8 3 22 16 16 16 17 13 11 10 17 15 12 6 13 14 17 4 12
10 24 5
13 24 3 5 2 5 11 14 8 5 10 17 16 8 4 14 21 15 3 6 17 25 8 2 3 3 19 10
13 22 19 ^
52 1 8 2 13 25 17 2 1 19 1 14 20 2 5 4 15 8 14 20 7 25 20 26 20 16 7
17 17 22 1
13 6 5 1 18 14 15 23 15 10 5 19 18 18 26 12 13 3 25 12 21 16 22 ^
56 1 4 16 3 6 26 26 10 20 13 1 20 24 10 21 7 25 2 25 22 9 15 16 26 5
12 1 7 4
24 7 14 20 1 4 7 3 10 10 15 6 13 6 22 6 20 22 2 26 9 6 14 1 16 3 24 12
^
52 0 24 17 12 14 18 18 9 9 11 22 13 9 2 22 15 20 20 18 8 13 13 21 8 11
5 26 8
19 17 4 2 15 26 13 24 26 11 16 9 1 8 18 21 6 1 8 1 16 4 20 23 5 ^
54 0 24 10 11 12 10 10 25 1 11 25 10 3 26 26 20 15 11 19 16 17 3 16 12
25 15 9
23 1 2 24 8 10 18 2 9 5 4 17 16 9 12 20 20 1 3 11 5 9 3 17 9 17 24 20
^
49 1 3 4 16 2 10 1 25 5 22 16 6 21 19 21 23 15 9 9 21 17 11 1 15 17 23
24 11 7
10 26 15 8 22 22 18 26 11 17 5 26 7 23 19 3 24 17 2 19 8 ^
49 0 26 21 12 5 18 22 20 7 15 6 4 5 20 13 10 23 18 16 25 18 15 2 20 14
24 22 17
3 15 16 1 10 2 12 2 21 1 21 21 4 18 13 1 24 18 23 17 19 23 ^
50 1 19 9 26 25 23 23 14 13 13 20 12 13 4 21 10 22 21 22 1 12 12 7 13
16 18 8 9
4 3 10 18 12 4 5 19 19 12 24 13 18 1 22 8 1 17 18 24 3 22 21 ^

48 1 7 19 26 13 26 6 26 6 18 12 5 26 17 18 19 22 10 12 8 23 11 17 23
25 5 5 11
4 13 7 18 4 19 23 3 13 15 4 3 9 24 14 5 25 21 22 18 25 ^
53 1 12 15 14 16 10 14 9 19 7 16 3 23 4 26 18 5 16 21 24 2 8 8 8 7 11
13 14 7 9
12 18 11 10 7 20 12 14 19 11 21 21 22 19 13 10 16 19 14 7 8 16 20 7 ^
49 0 3 9 23 18 7 26 16 22 21 21 5 14 11 2 2 16 22 11 26 7 22 23 5 23
10 4 22 24
19 20 9 16 10 24 4 19 26 4 25 4 8 17 7 7 2 23 11 18 19 ^
49 0 23 6 10 9 1 9 14 19 5 6 6 25 12 11 21 26 12 3 8 20 8 21 18 25 25
14 4 11
24 23 4 14 3 17 26 22 26 2 15 9 8 25 19 12 23 17 20 17 10 ^
54 0 4 23 20 17 21 14 14 9 18 11 18 2 18 4 2 22 25 15 18 10 23 8 21 26
7 14 10
6 6 15 15 13 2 11 15 22 2 9 4 4 6 16 11 11 3 16 13 19 25 16 5 7 20 23
^
47 0 9 25 11 24 24 25 13 8 15 23 16 6 3 18 23 16 23 8 20 4 1 1 14 13
12 24 14
14 26 23 13 26 14 21 25 25 4 9 13 5 8 9 21 8 12 26 15 ^
51 1 9 24 15 1 23 22 16 14 8 22 15 19 24 20 7 8 15 24 12 4 4 23 21 13
19 15 21
7 15 7 3 21 20 8 22 14 23 26 19 2 10 18 3 5 3 1 9 15 15 3 24 ^
60 0 13 23 9 7 1 13 17 14 25 9 16 2 2 6 13 7 19 25 17 1 5 21 2 7 22 5
6 25 3 12
19 6 2 4 24 17 9 18 20 19 4 11 14 1 6 8 26 6 9 22 4 10 2 7 21 9 8 24
25 10 ^
59 0 22 12 22 3 23 3 3 20 6 11 23 6 1 7 5 18 5 15 25 26 1 1 10 11 11 4
12 11 20
3 14 2 3 2 23 12 12 17 24 11 8 6 24 16 15 22 21 14 6 12 20 19 5 5 12
11 6 23 8 ^
50 0 16 23 7 24 6 21 2 17 17 5 25 11 25 20 25 24 18 6 12 19 25 7 6 5 2
14 12 16
1 15 7 1 26 19 19 13 20 11 17 6 20 5 24 24 1 21 11 9 20 14 ^
60 0 15 10 19 26 3 2 6 7 12 9 10 8 14 10 15 5 17 8 21 1 20 25 6 19 8 3
22 16 16
16 17 13 11 10 17 15 12 6 13 14 17 4 12 10 24 5 13 24 3 5 2 5 11 14 8
5 10 17
16 13 ^
57 0 4 14 21 15 3 6 17 25 8 2 3 3 19 10 13 22 22 8 2 13 25 17 2 1 19 1
14 20 2
5 4 15 8 14 20 7 25 20 26 20 16 7 17 17 22 1 13 6 5 1 18 14 15 23 15
10 21 ^
52 1 19 18 18 26 12 13 3 25 12 21 16 24 4 16 3 6 26 26 10 20 13 1 20
24 10 21 7
25 2 25 22 9 15 16 26 5 12 1 7 4 24 7 14 20 1 4 7 3 10 10 15 19 ^
51 0 13 6 22 6 20 22 2 26 9 6 14 1 16 3 24 25 24 17 12 14 18 18 9 9 11
22 13 9
2 22 15 20 20 18 8 13 13 21 8 11 5 26 8 19 17 4 2 15 26 13 21 ^
52 1 24 26 11 16 9 1 8 18 21 6 1 8 1 16 4 20 23 23 24 10 11 12 10 10
25 1 11 25
10 3 26 26 20 15 11 19 16 17 3 16 12 25 15 9 23 1 2 24 8 10 18 15 ^

55 1 9 5 4 17 16 9 12 20 20 1 3 11 5 9 3 17 9 17 24 21 3 4 16 2 10 1
25 5 22 16
6 21 19 21 23 15 9 9 21 17 11 1 15 17 23 24 11 7 10 26 15 8 22 22 11 ^
50 1 26 11 17 5 26 7 23 19 3 24 17 2 19 18 26 21 12 5 18 22 20 7 15 6
4 5 20 13
10 23 18 16 25 18 15 2 20 14 24 22 17 3 15 16 1 10 2 12 2 25 ^
51 0 1 21 21 4 18 13 1 24 18 23 17 19 6 19 9 26 25 23 23 14 13 13 20
12 13 4 21
10 22 21 22 1 12 12 7 13 16 18 8 9 4 3 10 18 12 4 5 19 19 12 24 ^
53 1 13 18 1 22 8 1 17 18 24 3 22 7 7 19 26 13 26 6 26 6 18 12 5 26 17
18 19 22
10 12 8 23 11 17 23 25 5 5 11 4 13 7 18 4 19 23 3 13 15 4 3 9 18 ^
51 1 14 5 25 21 22 18 17 12 15 14 16 10 14 9 19 7 16 3 23 4 26 18 5 16
21 24 2
8 8 8 7 11 13 14 7 9 12 18 11 10 7 20 12 14 19 11 21 21 22 19 26 ^
49 0 10 16 19 14 7 8 16 20 10 3 9 23 18 7 26 16 22 21 21 5 14 11 2 2
16 22 11
26 7 22 23 5 23 10 4 22 24 19 20 9 16 10 24 4 19 26 4 25 14 ^
52 0 8 17 7 7 2 23 11 18 19 23 6 10 9 1 9 14 19 5 6 6 25 12 11 21 26
12 3 8 20
8 21 18 25 25 14 4 11 24 23 4 14 3 17 26 22 26 2 15 9 8 25 24 ^
55 1 12 23 17 20 17 10 4 23 20 17 21 14 14 9 18 11 18 2 18 4 2 22 25
15 18 10
23 8 21 26 7 14 10 6 6 15 15 13 2 11 15 22 2 9 4 4 6 16 11 11 3 16 13
19 15 ^
49 0 16 5 7 20 22 9 25 11 24 24 25 13 8 15 23 16 6 3 18 23 16 23 8 20
4 1 1 14
13 12 24 14 14 26 23 13 26 14 21 25 25 4 9 13 5 8 9 21 9 ^
55 0 22 24 9 27 23 1 13 18 24 1 8 24 8 6 14 1 4 5 15 27 12 20 1 17 5 4
14 25 14
3 19 24 24 14 8 13 12 19 8 3 6 13 7 10 23 19 10 17 17 11 6 13 8 18 18
^
51 0 7 3 25 1 17 8 26 25 11 23 14 27 1 22 11 9 7 19 4 15 18 2 27 20 11
5 20 22
19 15 5 6 17 8 1 19 22 9 23 10 26 27 26 14 12 3 14 15 11 2 26 ^
50 0 6 22 20 8 25 4 25 24 23 20 3 3 16 2 2 27 2 15 14 27 3 17 12 27 27
7 26 2
22 11 9 10 24 15 1 27 11 27 4 1 25 25 16 6 2 22 12 17 21 14 ^
52 0 8 9 22 17 12 7 18 14 23 2 6 4 23 9 4 22 16 18 23 8 11 25 17 27 16
10 3 24
8 20 20 18 18 13 4 2 18 22 15 25 26 9 3 1 2 19 15 1 24 17 16 18 ^
49 0 9 14 11 8 16 19 16 22 10 16 24 25 7 22 12 20 15 23 22 18 22 14 6
26 21 10
21 7 13 2 16 7 24 16 22 13 10 7 11 16 5 9 25 13 9 14 14 17 14 ^
54 1 2 10 5 25 5 7 10 22 10 1 24 3 11 7 23 5 17 14 3 18 5 13 26 15 5
13 23 18
17 3 18 3 27 15 6 25 24 22 27 17 4 25 23 15 9 15 21 8 7 7 15 3 13 25 ^
47 0 25 3 4 10 24 12 7 16 25 18 21 21 20 27 13 14 23 27 1 21 9 21 14
22 9 9 5
15 13 17 3 4 9 16 1 8 23 17 23 26 8 24 10 20 25 27 25 ^
58 1 8 26 8 10 4 3 8 14 5 5 7 11 13 11 26 11 4 26 17 20 19 11 10 3 10
14 9 6 9

7 16 10 4 4 19 19 2 26 13 19 17 15 24 15 4 21 22 13 13 12 22 2 14 20 5
18 7 25 ^
58 0 17 24 20 13 6 17 9 20 2 10 19 3 22 4 1 11 3 5 3 21 11 15 12 23 26
5 2 27 6
5 16 6 3 2 23 5 3 20 20 4 24 2 18 21 7 14 10 27 23 6 24 6 19 23 3 9 22
15 ^
54 0 21 17 19 25 17 7 21 19 6 16 15 15 20 14 2 25 19 14 18 19 7 9 1 14
11 10 16
3 23 14 26 10 11 1 18 1 12 24 19 19 1 7 2 3 24 7 12 9 2 8 16 20 24 25
^
49 1 26 26 4 9 2 7 25 9 8 12 18 14 26 7 17 18 4 20 1 16 14 21 26 4 6 8
24 11 25
15 24 16 23 4 10 23 21 24 15 10 9 26 7 14 24 21 6 20 19 ^
56 1 17 16 17 1 3 12 1 4 13 3 9 21 23 7 18 11 1 19 20 23 12 12 27 13
13 15 16
13 1 16 15 12 26 3 16 16 8 17 13 21 4 6 5 19 14 16 4 16 11 14 18 18 27
9 13 22 ^
51 1 3 26 22 3 7 6 4 26 3 15 8 25 1 20 13 9 11 20 6 11 21 27 25 20 7 4
18 26 16
27 5 12 19 7 23 6 25 25 2 11 13 25 21 18 17 6 12 14 13 24 13 ^
48 1 14 19 26 27 25 6 1 15 4 7 27 24 27 21 5 27 19 18 8 21 3 23 20 21
25 21 1 9
17 22 5 22 8 2 13 27 8 19 27 21 2 5 8 4 27 10 6 25 ^
52 0 2 6 1 1 26 18 13 5 10 18 10 8 5 21 15 5 14 19 12 4 8 20 21 26 1
12 25 10 2
17 10 15 2 20 25 26 23 25 12 16 15 9 18 15 19 21 16 26 11 18 20 26 ^
53 0 6 22 7 8 8 12 2 21 18 12 11 5 19 3 19 27 18 25 22 22 23 16 2 1 20
16 14 22
6 15 22 1 21 6 6 10 2 26 12 11 6 27 14 7 20 26 20 17 7 19 5 23 14 ^
61 0 16 12 7 15 17 7 5 11 12 15 7 1 10 10 8 4 2 4 17 4 11 17 26 23 24
4 5 12 5
1 6 9 6 8 27 5 11 11 24 24 13 17 2 24 26 19 11 22 21 2 2 14 21 13 1 23
10 25 8
15 13 ^
52 0 19 23 2 9 26 6 23 13 5 24 12 21 11 14 4 12 10 12 22 4 3 20 17 10
13 16 9
10 27 9 12 13 20 10 12 23 23 11 19 13 25 22 8 1 5 16 20 9 19 9 24 26 ^
51 0 12 15 6 21 27 22 26 24 18 9 15 8 22 24 9 27 23 1 13 18 24 1 8 24
8 6 14 1
4 5 15 27 12 20 1 17 5 4 14 25 14 3 19 24 24 14 8 13 12 19 22 ^
56 0 3 6 13 7 10 23 19 10 17 17 11 6 13 8 18 6 7 3 25 1 17 8 26 25 11
23 14 27
1 22 11 9 7 19 4 15 18 2 27 20 11 5 20 22 19 15 5 6 17 8 1 19 22 9 23
17 ^
50 0 26 27 26 14 12 3 14 15 11 2 12 6 22 20 8 25 4 25 24 23 20 3 3 16
2 2 27 2
15 14 27 3 17 12 27 27 7 26 2 22 11 9 10 24 15 1 27 11 27 21 ^
52 0 1 25 25 16 6 2 22 12 17 21 12 8 9 22 17 12 7 18 14 23 2 6 4 23 9
4 22 16
18 23 8 11 25 17 27 16 10 3 24 8 20 20 18 18 13 4 2 18 22 15 25 10 ^
52 0 9 3 1 2 19 15 1 24 17 16 18 9 14 11 8 16 19 16 22 10 16 24 25 7
22 12 20

15 23 22 18 22 14 6 26 21 10 21 7 13 2 16 7 24 16 22 13 10 7 11 16 13
^
53 1 9 25 13 9 14 14 17 25 2 10 5 25 5 7 10 22 10 1 24 3 11 7 23 5 17
14 3 18 5
13 26 15 5 13 23 18 17 3 18 3 27 15 6 25 24 22 27 17 4 25 23 15 15 ^
52 1 15 21 8 7 7 15 3 13 22 25 3 4 10 24 12 7 16 25 18 21 21 20 27 13
14 23 27
1 21 9 21 14 22 9 9 5 15 13 17 3 4 9 16 1 8 23 17 23 26 8 24 14 ^
58 0 20 25 27 3 8 26 8 10 4 3 8 14 5 5 7 11 13 11 26 11 4 26 17 20 19
11 10 3
10 14 9 6 9 7 16 10 4 4 19 19 2 26 13 19 17 15 24 15 4 21 22 13 13 12
22 2 14
18 ^
60 0 5 18 7 17 24 20 13 6 17 9 20 2 10 19 3 22 4 1 11 3 5 3 21 11 15
12 23 26 5
2 27 6 5 16 6 3 2 23 5 3 20 20 4 24 2 18 21 7 14 10 27 23 6 24 6 19 23
3 9 25 ^
55 0 16 21 17 19 25 17 7 21 19 6 16 15 15 20 14 2 25 19 14 18 19 7 9 1
14 11 10
16 3 23 14 26 10 11 1 18 1 12 24 19 19 1 7 2 3 24 7 12 9 2 8 16 20 24
27 ^
51 1 26 26 4 9 2 7 25 9 8 12 18 14 26 7 17 18 4 20 1 16 14 21 26 4 6 8
24 11 25
15 24 16 23 4 10 23 21 24 15 10 9 26 7 14 24 21 6 20 5 17 15 ^
57 0 17 1 3 12 1 4 13 3 9 21 23 7 18 11 1 19 20 23 12 12 27 13 13 15
16 13 1 16
15 12 26 3 16 16 8 17 13 21 4 6 5 19 14 16 4 16 11 14 18 18 27 9 13 21
3 26 23 ^
52 0 3 7 6 4 26 3 15 8 25 1 20 13 9 11 20 6 11 21 27 25 20 7 4 18 26
16 27 5 12
19 7 23 6 25 25 2 11 13 25 21 18 17 6 12 14 13 24 11 14 19 26 12 ^
55 1 25 6 1 15 4 7 27 24 27 21 5 27 19 18 8 21 3 23 20 21 25 21 1 9 17
22 5 22
8 2 13 27 8 19 27 21 2 5 8 4 27 10 6 18 2 6 1 1 26 18 13 5 10 18 11 ^
52 0 8 5 21 15 5 14 19 12 4 8 20 21 26 1 12 25 10 2 17 10 15 2 20 25
26 23 25
12 16 15 9 18 15 19 21 16 26 11 18 20 6 22 7 8 8 12 2 21 18 12 11 27 ^
53 1 19 3 19 27 18 25 22 22 23 16 2 1 20 16 14 22 6 15 22 1 21 6 6 10
2 26 12
11 6 27 14 7 20 26 20 17 7 19 5 23 18 16 12 7 15 17 7 5 11 12 15 7 22
^
57 1 10 10 8 4 2 4 17 4 11 17 26 23 24 4 5 12 5 1 6 9 6 8 27 5 11 11
24 24 13
17 2 24 26 19 11 22 21 2 2 14 21 13 1 23 10 25 8 15 26 19 23 2 9 26 6
23 22 ^
52 1 5 24 12 21 11 14 4 12 10 12 22 4 3 20 17 10 13 16 9 10 27 9 12 13
20 10 12
23 23 11 19 13 25 22 8 1 5 16 20 9 19 9 24 20 12 15 6 21 27 22 26 16 ^
58 0 18 9 15 8 22 24 9 27 23 1 13 18 24 1 8 24 8 6 14 1 4 5 15 27 12
20 1 17 5
4 14 25 14 3 19 24 24 14 8 13 12 19 8 3 6 13 7 10 23 19 10 17 17 11 6
13 8 22 ^

55 0 6 7 3 25 1 17 8 26 25 11 23 14 27 1 22 11 9 7 19 4 15 18 2 27 20
11 5 20
22 19 15 5 6 17 8 1 19 22 9 23 10 26 27 26 14 12 3 14 15 11 2 12 6 22
16 ^
53 0 8 25 4 25 24 23 20 3 3 16 2 2 27 2 15 14 27 3 17 12 27 27 7 26 2
22 11 9
10 24 15 1 27 11 27 4 1 25 25 16 6 2 22 12 17 21 12 8 9 22 17 12 18 ^
54 0 7 18 14 23 2 6 4 23 9 4 22 16 18 23 8 11 25 17 27 16 10 3 24 8 20
20 18 18
13 4 2 18 22 15 25 26 9 3 1 2 19 15 1 24 17 16 18 9 14 11 8 16 19 27 ^
52 0 22 10 16 24 25 7 22 12 20 15 23 22 18 22 14 6 26 21 10 21 7 13 2
16 7 24
16 22 13 10 7 11 16 5 9 25 13 9 14 14 17 25 2 10 5 25 5 7 10 22 10 22
^
54 1 24 3 11 7 23 5 17 14 3 18 5 13 26 15 5 13 23 18 17 3 18 3 27 15 6
25 24 22
27 17 4 25 23 15 9 15 21 8 7 7 15 3 13 22 25 3 4 10 24 12 7 16 25 10 ^
56 0 21 21 20 27 13 14 23 27 1 21 9 21 14 22 9 9 5 15 13 17 3 4 9 16 1
8 23 17
23 26 8 24 10 20 25 27 3 8 26 8 10 4 3 8 14 5 5 7 11 13 11 26 11 4 26
2 ^
61 1 20 19 11 10 3 10 14 9 6 9 7 16 10 4 4 19 19 2 26 13 19 17 15 24
15 4 21 22
13 13 12 22 2 14 20 5 18 7 17 24 20 13 6 17 9 20 2 10 19 3 22 4 1 11 3
5 3 21
11 15 22 ^
53 0 23 26 5 2 27 6 5 16 6 3 2 23 5 3 20 20 4 24 2 18 21 7 14 10 27 23
6 24 6
19 23 3 9 22 16 21 17 19 25 17 7 21 19 6 16 15 15 20 14 2 25 19 25 ^
60 0 18 19 7 9 1 14 11 10 16 3 23 14 26 10 11 1 18 1 12 24 19 19 1 7 2
3 24 7
12 9 2 8 16 20 24 5 26 26 4 9 2 7 25 9 8 12 18 14 26 7 17 18 4 20 1 16
14 21 26
18 ^
54 0 6 8 24 11 25 15 24 16 23 4 10 23 21 24 15 10 9 26 7 14 24 21 6 20
5 17 16
17 1 3 12 1 4 13 3 9 21 23 7 18 11 1 19 20 23 12 12 27 13 13 15 16 13
24 ^
57 0 1 16 15 12 26 3 16 16 8 17 13 21 4 6 5 19 14 16 4 16 11 14 18 18
27 9 13
21 3 26 22 3 7 6 4 26 3 15 8 25 1 20 13 9 11 20 6 11 21 27 25 20 7 4
18 26 10 ^
48 0 27 5 12 19 7 23 6 25 25 2 11 13 25 21 18 17 6 12 14 13 24 11 14
19 26 27
25 6 1 15 4 7 27 24 27 21 5 27 19 18 8 21 3 23 20 21 25 8 ^
60 1 1 9 17 22 5 22 8 2 13 27 8 19 27 21 2 5 8 4 27 10 6 18 2 6 1 1 26
18 13 5
10 18 10 8 5 21 15 5 14 19 12 4 8 20 21 26 1 12 25 10 2 17 10 15 2 20
25 26 23
21 ^
54 1 12 16 15 9 18 15 19 21 16 26 11 18 20 6 22 7 8 8 12 2 21 18 12 11
5 19 3

19 27 18 25 22 22 23 16 2 1 20 16 14 22 6 15 22 1 21 6 6 10 2 26 12 11
24 ^
60 0 27 14 7 20 26 20 17 7 19 5 23 18 16 12 7 15 17 7 5 11 12 15 7 1
10 10 8 4
2 4 17 4 11 17 26 23 24 4 5 12 5 1 6 9 6 8 27 5 11 11 24 24 13 17 2 24
26 19 11
22 ^
55 0 21 2 2 14 21 13 1 23 10 25 8 15 26 19 23 2 9 26 6 23 13 5 24 12
21 11 14 4
12 10 12 22 4 3 20 17 10 13 16 9 10 27 9 12 13 20 10 12 23 23 11 19 13
25 13 ^
55 0 8 1 5 16 20 9 19 9 24 20 12 15 6 21 27 22 26 24 18 9 15 8 22 24 9
27 23 1
13 18 24 1 8 24 8 6 14 1 4 5 15 27 12 20 1 17 5 4 14 25 14 3 19 24 16
^
60 0 14 8 13 12 19 8 3 6 13 7 10 23 19 10 17 17 11 6 13 8 18 6 7 3 25
1 17 8 26
25 11 23 14 27 1 22 11 9 7 19 4 15 18 2 27 20 11 5 20 22 19 15 5 6 17
8 1 19 22
10 ^
55 1 15 9 11 26 19 22 27 2 21 8 20 23 26 12 10 21 9 15 13 25 7 26 1 13
5 9 1 3
9 21 22 7 1 23 28 1 2 8 22 12 18 28 5 18 14 7 11 17 20 20 7 21 13 8 22
^
52 1 21 22 2 16 20 15 28 9 3 22 13 10 23 4 16 11 14 1 10 8 14 14 15 18
13 12 21
18 25 15 27 17 5 1 23 7 17 20 11 14 10 4 21 17 19 25 17 22 18 21 19 17
^
57 0 10 15 26 20 8 13 19 7 8 18 7 12 14 6 8 22 21 11 10 14 23 25 9 15
22 18 6
17 24 7 13 6 12 7 2 18 18 26 13 13 16 8 24 10 3 26 9 20 1 12 2 14 13
23 12 25 5
^
57 0 24 11 6 3 8 27 8 21 26 3 11 19 18 5 5 25 26 24 4 23 15 15 23 21
24 5 15 21
3 3 18 2 24 27 11 20 10 11 4 5 12 2 23 6 5 3 1 14 26 15 12 6 28 12 19
20 9 ^
55 0 6 23 4 7 7 3 10 24 28 6 10 9 6 20 14 27 10 19 11 1 15 21 5 10 7
11 8 16 9
27 5 2 6 6 27 20 24 21 2 20 16 14 24 17 15 3 2 26 25 25 14 24 20 28 28
^
59 1 17 5 9 10 25 20 12 9 18 3 16 4 10 7 18 23 17 2 8 1 28 7 2 15 5 18
11 23 2
1 18 22 14 7 20 21 2 11 12 3 15 12 4 1 27 21 16 27 19 18 13 21 14 5 20
9 19 27
25 ^
55 0 3 5 16 16 22 2 13 7 4 13 19 27 15 7 16 25 3 4 4 17 12 13 11 12 26
6 1 10
15 18 4 19 10 27 9 4 25 13 21 25 26 21 20 12 25 21 7 17 28 27 3 13 14
19 18 ^
53 1 23 19 4 28 3 20 18 9 7 15 4 26 5 6 20 5 23 28 10 24 19 13 28 1 9
12 23 16

10 13 2 19 21 6 25 22 28 23 16 7 20 4 2 22 11 26 22 6 9 17 11 21 10 ^
47 0 1 3 28 11 27 25 12 8 26 26 28 24 13 22 17 6 4 24 15 17 18 4 22 23
9 24 12
27 27 25 9 13 15 11 9 22 28 26 11 18 15 10 6 16 17 14 24 ^
55 1 25 27 16 28 22 24 1 1 11 7 24 3 28 7 5 28 16 15 1 7 14 9 7 6 15
25 19 9 3
19 23 1 26 20 17 17 1 16 24 9 16 8 15 6 21 17 22 9 28 8 8 5 4 24 26 ^
47 1 14 4 21 26 23 7 8 27 23 23 22 19 24 25 16 19 12 22 10 20 3 7 14 1
20 6 26
23 27 4 16 20 18 12 20 15 19 16 18 20 27 11 11 17 24 18 16 ^
50 0 19 1 25 23 2 15 16 26 27 23 14 12 28 22 15 8 19 2 20 13 1 24 2 25
1 6 19
19 8 11 24 24 21 13 27 5 11 28 17 7 25 6 23 24 14 25 12 5 13 25 ^
60 0 2 5 8 10 16 11 5 26 18 19 21 3 12 11 13 4 14 22 22 14 16 13 3 22
16 23 5
19 6 13 10 26 17 27 26 4 3 25 6 14 2 3 5 7 23 11 22 8 25 2 9 25 18 17
8 2 14 4
19 22 ^
54 1 5 27 13 9 2 27 28 2 17 23 10 27 18 26 7 22 16 3 27 1 26 21 28 10
3 6 2 2
10 17 13 16 6 17 21 23 13 20 22 5 6 11 12 12 8 23 13 17 9 23 20 3 28
21 ^
56 0 12 17 25 28 19 5 21 4 27 8 1 19 14 20 6 7 9 1 6 22 3 19 26 14 8 6
7 19 15
23 1 17 16 6 26 14 5 22 25 4 7 10 16 21 10 18 19 24 16 23 8 3 17 28 18
13 ^
58 0 2 5 3 21 21 15 6 24 1 4 24 18 10 22 1 21 12 5 4 4 20 25 24 26 8
25 11 2 7
27 22 19 4 18 27 10 28 4 12 24 8 16 12 11 16 17 25 8 12 16 1 9 9 10 5
24 23 11 ^
62 1 5 14 18 8 4 9 5 17 8 28 1 22 4 11 3 2 17 3 14 9 27 13 18 24 9 8 7
28 25 14
21 27 24 6 18 16 2 12 15 9 14 10 1 8 17 4 6 15 26 11 15 2 28 20 26 16
3 7 5 8 9
24 ^
50 1 10 12 25 11 9 13 24 15 20 2 4 8 2 22 20 19 4 15 14 28 13 25 10 10
12 28 24
22 26 28 15 9 11 26 19 22 27 2 21 8 20 23 26 12 10 21 9 15 13 17 ^
60 1 7 26 1 13 5 9 1 3 9 21 22 7 1 23 28 1 2 8 22 12 18 28 5 18 14 7
11 17 20
20 7 21 13 8 28 21 22 2 16 20 15 28 9 3 22 13 10 23 4 16 11 14 1 10 8
14 14 15
18 17 ^
52 0 12 21 18 25 15 27 17 5 1 23 7 17 20 11 14 10 4 21 17 19 25 17 22
18 21 19
17 10 15 26 20 8 13 19 7 8 18 7 12 14 6 8 22 21 11 10 14 23 25 9 15 19
^
57 1 18 6 17 24 7 13 6 12 7 2 18 18 26 13 13 16 8 24 10 3 26 9 20 1 12
2 14 13
23 12 25 3 24 11 6 3 8 27 8 21 26 3 11 19 18 5 5 25 26 24 4 23 15 15
23 21 12 ^

67 1 5 15 21 3 3 18 2 24 27 11 20 10 11 4 5 12 2 23 6 5 3 1 14 26 15
12 6 28 12
19 20 6 23 4 7 7 3 10 24 28 6 10 9 6 20 14 27 10 19 11 1 15 21 5 10 7
11 8 16 9
27 5 2 6 6 27 2 ^
56 1 24 21 2 20 16 14 24 17 15 3 2 26 25 25 14 24 20 28 14 17 5 9 10
25 20 12 9
18 3 16 4 10 7 18 23 17 2 8 1 28 7 2 15 5 18 11 23 2 1 18 22 14 7 20
21 24 ^
60 1 11 12 3 15 12 4 1 27 21 16 27 19 18 13 21 14 5 20 9 19 27 6 3 5
16 16 22 2
13 7 4 13 19 27 15 7 16 25 3 4 4 17 12 13 11 12 26 6 1 10 15 18 4 19
10 27 9 4
25 27 ^
51 0 21 25 26 21 20 12 25 21 7 17 28 27 3 13 14 19 3 23 19 4 28 3 20
18 9 7 15
4 26 5 6 20 5 23 28 10 24 19 13 28 1 9 12 23 16 10 13 2 19 21 23 ^
49 1 25 22 28 23 16 7 20 4 2 22 11 26 22 6 9 17 11 21 5 1 3 28 11 27
25 12 8 26
26 28 24 13 22 17 6 4 24 15 17 18 4 22 23 9 24 12 27 27 9 ^
56 0 9 13 15 11 9 22 28 26 11 18 15 10 6 16 17 14 11 25 27 16 28 22 24
1 1 11 7
24 3 28 7 5 28 16 15 1 7 14 9 7 6 15 25 19 9 3 19 23 1 26 20 17 17 1
16 16 ^
51 0 9 16 8 15 6 21 17 22 9 28 8 8 5 4 24 27 14 4 21 26 23 7 8 27 23
23 22 19
24 25 16 19 12 22 10 20 3 7 14 1 20 6 26 23 27 4 16 20 18 12 22 ^
51 0 15 19 16 18 20 27 11 11 17 24 18 1 19 1 25 23 2 15 16 26 27 23 14
12 28 22
15 8 19 2 20 13 1 24 2 25 1 6 19 19 8 11 24 24 21 13 27 5 11 28 16 ^
59 0 7 25 6 23 24 14 25 12 5 13 26 2 5 8 10 16 11 5 26 18 19 21 3 12
11 13 4 14
22 22 14 16 13 3 22 16 23 5 19 6 13 10 26 17 27 26 4 3 25 6 14 2 3 5 7
23 11 22
10 ^
59 0 25 2 9 25 18 17 8 2 14 4 19 1 5 27 13 9 2 27 28 2 17 23 10 27 18
26 7 22
16 3 27 1 26 21 28 10 3 6 2 2 10 17 13 16 6 17 21 23 13 20 22 5 6 11
12 12 8 23
7 ^
56 1 17 9 23 20 3 28 27 12 17 25 28 19 5 21 4 27 8 1 19 14 20 6 7 9 1
6 22 3 19
26 14 8 6 7 19 15 23 1 17 16 6 26 14 5 22 25 4 7 10 16 21 10 18 19 24
16 ^
57 1 23 8 3 17 28 18 10 2 5 3 21 21 15 6 24 1 4 24 18 10 22 1 21 12 5
4 4 20 25
24 26 8 25 11 2 7 27 22 19 4 18 27 10 28 4 12 24 8 16 12 11 16 17 25 8
12 18 ^
63 0 1 9 9 10 5 24 23 18 5 14 18 8 4 9 5 17 8 28 1 22 4 11 3 2 17 3 14
9 27 13
18 24 9 8 7 28 25 14 21 27 24 6 18 16 2 12 15 9 14 10 1 8 17 4 6 15 26
11 15 2

28 20 16 ^
52 1 16 3 7 5 8 9 26 10 12 25 11 9 13 24 15 20 2 4 8 2 22 20 19 4 15
14 28 13
25 10 10 12 28 24 22 26 28 15 9 11 26 19 22 27 2 21 8 20 23 26 12 28 ^
60 0 21 9 15 13 25 7 26 1 13 5 9 1 3 9 21 22 7 1 23 28 1 2 8 22 12 18
28 5 18
14 7 11 17 20 20 7 21 13 8 28 21 22 2 16 20 15 28 9 3 22 13 10 23 4 16
11 14 1
10 20 ^
54 1 14 14 15 18 13 12 21 18 25 15 27 17 5 1 23 7 17 20 11 14 10 4 21
17 19 25
17 22 18 21 19 17 10 15 26 20 8 13 19 7 8 18 7 12 14 6 8 22 21 11 10
14 23 11 ^
58 0 9 15 22 18 6 17 24 7 13 6 12 7 2 18 18 26 13 13 16 8 24 10 3 26 9
20 1 12
2 14 13 23 12 25 3 24 11 6 3 8 27 8 21 26 3 11 19 18 5 5 25 26 24 4 23
15 15 27
^
64 1 21 24 5 15 21 3 3 18 2 24 27 11 20 10 11 4 5 12 2 23 6 5 3 1 14
26 15 12 6
28 12 19 20 6 23 4 7 7 3 10 24 28 6 10 9 6 20 14 27 10 19 11 1 15 21 5
10 7 11
8 16 9 27 20 ^
58 1 2 6 6 27 20 24 21 2 20 16 14 24 17 15 3 2 26 25 25 14 24 20 28 14
17 5 9
10 25 20 12 9 18 3 16 4 10 7 18 23 17 2 8 1 28 7 2 15 5 18 11 23 2 1
18 22 14
28 ^
62 0 20 21 2 11 12 3 15 12 4 1 27 21 16 27 19 18 13 21 14 5 20 9 19 27
6 3 5 16
16 22 2 13 7 4 13 19 27 15 7 16 25 3 4 4 17 12 13 11 12 26 6 1 10 15
18 4 19 10
27 9 4 26 ^
52 0 13 21 25 26 21 20 12 25 21 7 17 28 27 3 13 14 19 3 23 19 4 28 3
20 18 9 7
15 4 26 5 6 20 5 23 28 10 24 19 13 28 1 9 12 23 16 10 13 2 19 21 27 ^
49 1 25 22 28 23 16 7 20 4 2 22 11 26 22 6 9 17 11 21 5 1 3 28 11 27
25 12 8 26
26 28 24 13 22 17 6 4 24 15 17 18 4 22 23 9 24 12 27 27 26 ^
57 0 9 13 15 11 9 22 28 26 11 18 15 10 6 16 17 14 11 25 27 16 28 22 24
1 1 11 7
24 3 28 7 5 28 16 15 1 7 14 9 7 6 15 25 19 9 3 19 23 1 26 20 17 17 1
16 24 9 ^
52 1 9 16 8 15 6 21 17 22 9 28 8 8 5 4 24 27 14 4 21 26 23 7 8 27 23
23 22 19
24 25 16 19 12 22 10 20 3 7 14 1 20 6 26 23 27 4 16 20 18 12 20 19 ^
52 1 19 16 18 20 27 11 11 17 24 18 1 19 1 25 23 2 15 16 26 27 23 14 12
28 22 15
8 19 2 20 13 1 24 2 25 1 6 19 19 8 11 24 24 21 13 27 5 11 28 17 7 24 ^
59 0 6 23 24 14 25 12 5 13 26 2 5 8 10 16 11 5 26 18 19 21 3 12 11 13
4 14 22

22 14 16 13 3 22 16 23 5 19 6 13 10 26 17 27 26 4 3 25 6 14 2 3 5 7 23
11 22 8
25 26 ^
59 0 9 25 18 17 8 2 14 4 19 1 5 27 13 9 2 27 28 2 17 23 10 27 18 26 7
22 16 3
27 1 26 21 28 10 3 6 2 2 10 17 13 16 6 17 21 23 13 20 22 5 6 11 12 12
8 23 13
17 21 ^
56 1 23 20 3 28 27 12 17 25 28 19 5 21 4 27 8 1 19 14 20 6 7 9 1 6 22
3 19 26
14 8 6 7 19 15 23 1 17 16 6 26 14 5 22 25 4 7 10 16 21 10 18 19 24 16
23 20 ^
60 0 3 17 28 18 10 2 5 3 21 21 15 6 24 1 4 24 18 10 22 1 21 12 5 4 4
20 25 24
26 8 25 11 2 7 27 22 19 4 18 27 10 28 4 12 24 8 16 12 11 16 17 25 8 12
16 1 9 9
10 21 ^
60 0 24 23 18 5 14 18 8 4 9 5 17 8 28 1 22 4 11 3 2 17 3 14 9 27 13 18
24 9 8 7
28 25 14 21 27 24 6 18 16 2 12 15 9 14 10 1 8 17 4 6 15 26 11 15 2 28
20 26 16
25 ^
53 1 7 5 8 9 26 10 12 25 11 9 13 24 15 20 2 4 8 2 22 20 19 4 15 14 28
13 25 10
10 12 28 24 22 26 28 15 9 11 26 19 22 27 2 21 8 20 23 26 12 10 21 9 24
^
62 1 13 25 7 26 1 13 5 9 1 3 9 21 22 7 1 23 28 1 2 8 22 12 18 28 5 18
14 7 11
17 20 20 7 21 13 8 28 21 22 2 16 20 15 28 9 3 22 13 10 23 4 16 11 14 1
10 8 14
14 15 18 13 ^
54 0 12 21 18 25 15 27 17 5 1 23 7 17 20 11 14 10 4 21 17 19 25 17 22
18 21 19
17 10 15 26 20 8 13 19 7 8 18 7 12 14 6 8 22 21 11 10 14 23 25 9 15 22
18 13 ^
58 0 17 24 7 13 6 12 7 2 18 18 26 13 13 16 8 24 10 3 26 9 20 1 12 2 14
13 23 12
25 3 24 11 6 3 8 27 8 21 26 3 11 19 18 5 5 25 26 24 4 23 15 15 23 21
24 5 15 26
^
67 1 3 3 18 2 24 27 11 20 10 11 4 5 12 2 23 6 5 3 1 14 26 15 12 6 28
12 19 20 6
23 4 7 7 3 10 24 28 6 10 9 6 20 14 27 10 19 11 1 15 21 5 10 7 11 8 16
9 27 5 2
6 6 27 20 24 21 12 ^
62 0 20 16 14 24 17 15 3 2 26 25 25 14 24 20 28 14 17 5 9 10 25 20 12
9 18 3 16
4 10 7 18 23 17 2 8 1 28 7 2 15 5 18 11 23 2 1 18 22 14 7 20 21 2 11
12 3 15 12
4 1 27 18 ^
61 1 4 9 14 16 15 8 11 21 20 10 10 21 23 20 2 11 23 1 11 1 5 3 23 16
15 27 14 5

16 3 22 2 3 24 3 19 29 4 4 10 8 20 14 15 1 26 12 27 25 4 28 22 11 19
19 24 5 20
8 9 26 ^
55 1 25 19 17 19 15 7 24 24 21 3 20 16 8 3 17 28 18 29 9 23 9 10 29 4
12 24 15
5 8 22 17 29 12 3 8 29 15 21 21 4 7 20 7 10 7 26 10 16 24 6 7 12 8 12
28 ^
52 1 24 9 17 11 28 12 26 26 6 29 13 10 20 6 23 10 4 3 26 26 14 20 20
25 14 13
15 24 14 11 4 23 27 24 20 9 16 17 24 13 12 6 1 14 26 25 7 8 21 1 19 27
^
65 0 3 2 2 17 21 13 5 9 21 28 9 1 13 2 21 8 24 1 3 27 13 18 19 4 15 12
5 2 16
27 16 6 2 1 23 17 14 28 19 23 6 22 16 4 19 6 16 28 3 3 1 15 13 22 6 15
11 6 4
24 25 15 19 23 12 ^
55 0 1 9 15 10 12 7 25 11 25 2 27 6 6 18 4 13 1 4 11 25 17 6 29 21 24
27 21 17
17 26 24 13 4 22 24 29 2 11 8 17 4 23 11 26 1 25 28 15 19 10 3 9 29 27
24 ^
53 1 22 3 27 21 2 26 19 9 19 27 23 3 12 17 10 4 8 27 19 16 1 21 23 28
23 18 8
25 18 26 20 8 17 19 2 4 11 7 18 6 24 2 25 25 5 27 20 3 28 20 18 21 11
^
55 0 25 10 1 28 28 29 16 20 19 26 16 21 6 23 15 28 26 28 7 15 18 11 5
17 20 26
1 10 14 8 4 16 26 17 8 10 22 22 18 1 20 24 7 1 9 7 19 5 22 1 4 26 11
10 20 ^
58 0 5 14 2 6 4 2 9 22 15 13 20 22 26 18 16 9 2 15 20 27 8 6 17 5 15 9
25 10 26
4 5 27 5 19 21 5 6 11 29 19 8 2 24 26 22 19 18 17 23 14 21 14 22 13 20
5 29 12 ^
54 0 5 26 6 21 29 3 5 12 16 11 13 16 18 18 10 11 20 21 12 11 12 22 14
19 3 21
15 6 25 26 29 24 19 17 18 25 22 12 13 23 23 12 10 13 1 22 10 2 22 2 28
10 17 28
^
56 0 3 6 19 28 17 11 1 9 14 4 29 29 23 25 13 19 18 2 7 21 6 8 29 5 11
22 14 14
9 28 13 25 24 28 8 14 9 3 14 18 26 9 9 12 13 17 25 16 5 8 4 24 10 24
23 25 ^
49 0 29 21 9 18 28 25 11 3 16 4 21 25 3 4 18 12 12 29 26 13 26 23 26 7
23 10 9
27 26 21 25 21 18 9 24 16 22 5 21 7 7 27 4 20 23 13 27 10 27 ^
58 1 29 7 19 13 13 8 19 20 18 2 19 22 29 3 10 6 9 13 24 17 16 9 20 21
27 2 19 7
8 26 25 12 2 14 3 1 1 16 12 20 28 24 12 5 24 25 20 1 12 25 9 2 6 20 15
14 29 20
^
56 0 21 28 1 17 6 17 27 1 2 10 16 19 5 14 10 27 11 17 3 29 14 27 17 5
2 18 20

11 27 18 22 27 22 9 8 18 3 14 10 5 2 24 18 23 21 26 8 22 8 1 6 27 27
15 29 18 ^
64 0 2 29 22 8 16 20 4 12 9 6 12 16 16 7 9 20 29 11 9 4 1 15 25 16 29
10 22 7 2
8 5 18 14 23 24 4 6 26 3 11 6 12 1 7 14 24 14 6 10 21 16 23 29 25 6 14
17 2 12
10 5 10 15 25 ^
62 1 8 15 3 7 13 25 16 14 1 29 22 26 15 27 9 1 8 8 28 6 13 5 13 3 15 5
23 8 23
2 5 5 4 17 13 14 7 17 12 27 3 18 5 7 5 26 18 15 22 28 16 13 7 2 23 19
25 29 17
7 16 22 ^
55 1 23 11 11 15 2 13 9 26 2 24 26 7 28 11 2 29 7 22 23 5 28 19 1 27
29 1 24 11
18 20 3 13 11 7 3 15 17 24 1 18 13 6 3 25 27 16 28 18 24 8 23 7 29 28
15 ^
52 0 28 14 12 28 27 22 4 14 25 1 3 9 7 11 14 15 16 10 19 12 19 11 20
13 28 4 27
28 7 27 12 4 28 21 17 22 20 17 15 15 23 22 13 12 21 22 21 11 12 14 12
28 ^
62 0 27 8 7 4 9 14 16 15 8 11 21 20 10 10 21 23 20 2 11 23 1 11 1 5 3
23 16 15
27 14 5 16 3 22 2 3 24 3 19 29 4 4 10 8 20 14 15 1 26 12 27 25 4 28 22
11 19 19
24 5 20 18 ^
58 0 9 5 25 19 17 19 15 7 24 24 21 3 20 16 8 3 17 28 18 29 9 23 9 10
29 4 12 24
15 5 8 22 17 29 12 3 8 29 15 21 21 4 7 20 7 10 7 26 10 16 24 6 7 12 8
12 15 16 ^
56 0 9 17 11 28 12 26 26 6 29 13 10 20 6 23 10 4 3 26 26 14 20 20 25
14 13 15
24 14 11 4 23 27 24 20 9 16 17 24 13 12 6 1 14 26 25 7 8 21 1 19 3 2 2
17 21 23
^
64 1 5 9 21 28 9 1 13 2 21 8 24 1 3 27 13 18 19 4 15 12 5 2 16 27 16 6
2 1 23
17 14 28 19 23 6 22 16 4 19 6 16 28 3 3 1 15 13 22 6 15 11 6 4 24 25
15 19 23
28 1 9 15 10 24 ^
54 0 7 25 11 25 2 27 6 6 18 4 13 1 4 11 25 17 6 29 21 24 27 21 17 17
26 24 13 4
22 24 29 2 11 8 17 4 23 11 26 1 25 28 15 19 10 3 9 29 27 7 22 3 27 29
^
54 1 2 26 19 9 19 27 23 3 12 17 10 4 8 27 19 16 1 21 23 28 23 18 8 25
18 26 20
8 17 19 2 4 11 7 18 6 24 2 25 25 5 27 20 3 28 20 18 21 10 25 10 1 28
27 ^
60 0 29 16 20 19 26 16 21 6 23 15 28 26 28 7 15 18 11 5 17 20 26 1 10
14 8 4 16
26 17 8 10 22 22 18 1 20 24 7 1 9 7 19 5 22 1 4 26 11 10 28 5 14 2 6 4
2 9 22
15 22 ^

54 1 20 22 26 18 16 9 2 15 20 27 8 6 17 5 15 9 25 10 26 4 5 27 5 19 21
5 6 11
29 19 8 2 24 26 22 19 18 17 23 14 21 14 22 13 20 5 29 6 5 26 6 21 29
28 ^
57 1 5 12 16 11 13 16 18 18 10 11 20 21 12 11 12 22 14 19 3 21 15 6 25
26 29 24
19 17 18 25 22 12 13 23 23 12 10 13 1 22 10 2 22 2 28 10 17 18 3 6 19
28 17 11
1 9 23 ^
54 0 14 4 29 29 23 25 13 19 18 2 7 21 6 8 29 5 11 22 14 14 9 28 13 25
24 28 8
14 9 3 14 18 26 9 9 12 13 17 25 16 5 8 4 24 10 24 23 4 29 21 9 18 28
27 ^
52 1 11 3 16 4 21 25 3 4 18 12 12 29 26 13 26 23 26 7 23 10 9 27 26 21
25 21 18
9 24 16 22 5 21 7 7 27 4 20 23 13 27 10 23 29 7 19 13 13 8 19 20 23 ^
61 0 2 19 22 29 3 10 6 9 13 24 17 16 9 20 21 27 2 19 7 8 26 25 12 2 14
3 1 1 16
12 20 28 24 12 5 24 25 20 1 12 25 9 2 6 20 15 14 29 18 21 28 1 17 6 17
27 1 2
10 16 19 ^
57 1 5 14 10 27 11 17 3 29 14 27 17 5 2 18 20 11 27 18 22 27 22 9 8 18
3 14 10
5 2 24 18 23 21 26 8 22 8 1 6 27 27 15 29 20 2 29 22 8 16 20 4 12 9 6
12 16 24 ^
64 0 7 9 20 29 11 9 4 1 15 25 16 29 10 22 7 2 8 5 18 14 23 24 4 6 26 3
11 6 12
1 7 14 24 14 6 10 21 16 23 29 25 6 14 17 2 12 10 5 10 15 25 8 15 3 7
13 25 16
14 1 29 22 26 10 ^
63 1 27 9 1 8 8 28 6 13 5 13 3 15 5 23 8 23 2 5 5 4 17 13 14 7 17 12
27 3 18 5
7 5 26 18 15 22 28 16 13 7 2 23 19 25 29 17 7 16 25 23 11 11 15 2 13 9
26 2 24
26 7 28 11 ^
56 1 2 29 7 22 23 5 28 19 1 27 29 1 24 11 18 20 3 13 11 7 3 15 17 24 1
18 13 6
3 25 27 16 28 18 24 8 23 7 29 28 6 28 14 12 28 27 22 4 14 25 1 3 9 7
11 29 ^
54 0 15 16 10 19 12 19 11 20 13 28 4 27 28 7 27 12 4 28 21 17 22 20 17
15 15 23
22 13 12 21 22 21 11 12 14 12 18 27 8 7 4 9 14 16 15 8 11 21 20 10 10
21 23 22 ^
63 0 2 11 23 1 11 1 5 3 23 16 15 27 14 5 16 3 22 2 3 24 3 19 29 4 4 10
8 20 14
15 1 26 12 27 25 4 28 22 11 19 19 24 5 20 8 9 5 25 19 17 19 15 7 24 24
21 3 20
16 8 3 17 19 ^
59 0 18 29 9 23 9 10 29 4 12 24 15 5 8 22 17 29 12 3 8 29 15 21 21 4 7
20 7 10
7 26 10 16 24 6 7 12 8 12 15 24 9 17 11 28 12 26 26 6 29 13 10 20 6 23
10 4 3

26 10 ^
62 0 14 20 20 25 14 13 15 24 14 11 4 23 27 24 20 9 16 17 24 13 12 6 1
14 26 25
7 8 21 1 19 3 2 2 17 21 13 5 9 21 28 9 1 13 2 21 8 24 1 3 27 13 18 19
4 15 12 5
2 16 27 29 ^
62 0 6 2 1 23 17 14 28 19 23 6 22 16 4 19 6 16 28 3 3 1 15 13 22 6 15
11 6 4 24
25 15 19 23 28 1 9 15 10 12 7 25 11 25 2 27 6 6 18 4 13 1 4 11 25 17 6
29 21 24
27 21 18 ^
54 1 17 26 24 13 4 22 24 29 2 11 8 17 4 23 11 26 1 25 28 15 19 10 3 9
29 27 7
22 3 27 21 2 26 19 9 19 27 23 3 12 17 10 4 8 27 19 16 1 21 23 28 23 18
17 ^
52 0 25 18 26 20 8 17 19 2 4 11 7 18 6 24 2 25 25 5 27 20 3 28 20 18
21 10 25
10 1 28 28 29 16 20 19 26 16 21 6 23 15 28 26 28 7 15 18 11 5 17 20 13
^
68 0 1 10 14 8 4 16 26 17 8 10 22 22 18 1 20 24 7 1 9 7 19 5 22 1 4 26
11 10 28
5 14 2 6 4 2 9 22 15 13 20 22 26 18 16 9 2 15 20 27 8 6 17 5 15 9 25
10 26 4 5
27 5 19 21 5 6 11 19 ^
54 1 19 8 2 24 26 22 19 18 17 23 14 21 14 22 13 20 5 29 6 5 26 6 21 29
3 5 12
16 11 13 16 18 18 10 11 20 21 12 11 12 22 14 19 3 21 15 6 25 26 29 24
19 17 24 ^
59 0 25 22 12 13 23 23 12 10 13 1 22 10 2 22 2 28 10 17 18 3 6 19 28
17 11 1 9
14 4 29 29 23 25 13 19 18 2 7 21 6 8 29 5 11 22 14 14 9 28 13 25 24 28
8 14 9 3
14 16 ^
53 0 26 9 9 12 13 17 25 16 5 8 4 24 10 24 23 4 29 21 9 18 28 25 11 3
16 4 21 25
3 4 18 12 12 29 26 13 26 23 26 7 23 10 9 27 26 21 25 21 18 9 24 16 17
^
59 0 5 21 7 7 27 4 20 23 13 27 10 23 29 7 19 13 13 8 19 20 18 2 19 22
29 3 10 6
9 13 24 17 16 9 20 21 27 2 19 7 8 26 25 12 2 14 3 1 1 16 12 20 28 24
12 5 24 25
19 ^
60 0 1 12 25 9 2 6 20 15 14 29 18 21 28 1 17 6 17 27 1 2 10 16 19 5 14
10 27 11
17 3 29 14 27 17 5 2 18 20 11 27 18 22 27 22 9 8 18 3 14 10 5 2 24 18
23 21 26
8 22 13 ^
62 0 1 6 27 27 15 29 20 2 29 22 8 16 20 4 12 9 6 12 16 16 7 9 20 29 11
9 4 1 15
25 16 29 10 22 7 2 8 5 18 14 23 24 4 6 26 3 11 6 12 1 7 14 24 14 6 10
21 16 23
29 25 24 ^

67 0 14 17 2 12 10 5 10 15 25 8 15 3 7 13 25 16 14 1 29 22 26 15 27 9
1 8 8 28
6 13 5 13 3 15 5 23 8 23 2 5 5 4 17 13 14 7 17 12 27 3 18 5 7 5 26 18
15 22 28
16 13 7 2 23 19 25 14 ^
58 1 17 7 16 25 23 11 11 15 2 13 9 26 2 24 26 7 28 11 2 29 7 22 23 5
28 19 1 27
29 1 24 11 18 20 3 13 11 7 3 15 17 24 1 18 13 6 3 25 27 16 28 18 24 8
23 7 29
11 ^
55 0 6 28 14 12 28 27 22 4 14 25 1 3 9 7 11 14 15 16 10 19 12 19 11 20
13 28 4
27 28 7 27 12 4 28 21 17 22 20 17 15 15 23 22 13 12 21 22 21 11 12 14
12 18 27
10 ^
65 0 7 4 9 14 16 15 8 11 21 20 10 10 21 23 20 2 11 23 1 11 1 5 3 23 16
15 27 14
5 16 3 22 2 3 24 3 19 29 4 4 10 8 20 14 15 1 26 12 27 25 4 28 22 11 19
19 24 5
20 8 9 5 25 19 20 ^
59 1 19 15 7 24 24 21 3 20 16 8 3 17 28 18 29 9 23 9 10 29 4 12 24 15
5 8 22 17
29 12 3 8 29 15 21 21 4 7 20 7 10 7 26 10 16 24 6 7 12 8 12 15 24 9 17
11 28 12
23 ^
61 0 26 6 29 13 10 20 6 23 10 4 3 26 26 14 20 20 25 14 13 15 24 14 11
4 23 27
24 20 9 16 17 24 13 12 6 1 14 26 25 7 8 21 1 19 3 2 2 17 21 13 5 9 21
28 9 1 13
2 21 8 29 ^
68 0 1 3 27 13 18 19 4 15 12 5 2 16 27 16 6 2 1 23 17 14 28 19 23 6 22
16 4 19
6 16 28 3 3 1 15 13 22 6 15 11 6 4 24 25 15 19 23 28 1 9 15 10 12 7 25
11 25 2
27 6 6 18 4 13 1 4 11 26 ^
54 1 17 6 29 21 24 27 21 17 17 26 24 13 4 22 24 29 2 11 8 17 4 23 11
26 1 25 28
15 19 10 3 9 29 27 7 22 3 27 21 2 26 19 9 19 27 23 3 12 17 10 4 8 27
20 ^
52 1 16 1 21 23 28 23 18 8 25 18 26 20 8 17 19 2 4 11 7 18 6 24 2 25
25 5 27 20
3 28 20 18 21 10 25 10 1 28 28 29 16 20 19 26 16 21 6 23 15 28 26 12 ^
67 0 7 15 18 11 5 17 20 26 1 10 14 8 4 16 26 17 8 10 22 22 18 1 20 24
7 1 9 7
19 5 22 1 4 26 11 10 28 5 14 2 6 4 2 9 22 15 13 20 22 26 18 16 9 2 15
20 27 8 6
17 5 15 9 25 10 26 19 ^
58 0 5 27 5 19 21 5 6 11 29 19 8 2 24 26 22 19 18 17 23 14 21 14 22 13
20 5 29
6 5 26 6 21 29 3 5 12 16 11 13 16 18 18 10 11 20 21 12 11 12 22 14 19
3 21 15 6
25 27 ^

57 0 29 24 19 17 18 25 22 12 13 23 23 12 10 13 1 22 10 2 22 2 28 10 17
18 3 6
19 28 17 11 1 9 14 4 29 29 23 25 13 19 18 2 7 21 6 8 29 5 11 22 14 14
9 28 13
25 25 ^
52 0 15 29 9 28 20 2 30 26 21 17 8 28 17 22 29 24 8 11 18 29 15 6 7 27
27 17 24
18 23 11 19 8 30 5 24 22 12 25 15 28 23 5 10 21 5 8 7 3 10 19 17 18 ^
62 0 9 15 29 10 7 4 1 16 21 16 29 13 18 5 3 8 15 8 21 29 20 5 27 2 13
27 7 7 30
2 18 26 10 2 5 29 21 15 25 26 24 8 12 20 3 9 10 30 7 12 24 5 13 1 6 10
25 13 9
29 30 7 ^
62 0 29 17 26 6 25 4 7 25 8 15 26 30 11 2 9 6 22 6 21 28 3 3 6 12 13
15 7 17 16
26 3 19 3 5 2 29 16 16 22 26 13 12 9 19 16 27 25 16 7 15 22 22 5 6 13
4 12 11
27 5 19 15 ^
57 0 9 30 9 1 28 3 21 22 7 17 2 25 1 28 21 11 17 20 28 12 10 10 3 15
13 20 1 20
11 1 30 22 23 15 26 12 21 7 27 9 27 29 5 26 16 11 23 28 18 1 13 23 30
6 3 11 25
^
61 1 16 10 13 13 19 15 18 7 22 10 29 3 25 30 8 22 13 4 4 27 22 18 21
23 17 15
16 19 20 17 10 17 21 20 6 14 14 1 10 18 4 5 5 9 17 18 14 5 12 22 22 11
8 18 24
21 3 7 21 9 22 ^
58 1 3 4 23 14 17 25 27 10 5 16 30 23 2 4 25 3 16 28 4 6 6 17 12 18 2
23 7 29
23 29 27 19 7 20 21 25 5 17 9 7 1 21 20 19 29 19 14 8 23 23 19 3 18 17
3 29 14
17 ^
62 1 23 28 5 22 8 10 1 1 11 27 18 3 2 22 28 14 24 12 1 3 30 29 11 22
19 7 16 1
11 4 28 8 20 25 2 6 8 5 14 11 21 9 16 8 25 6 26 27 1 29 12 8 16 4 17 5
30 30 24
12 19 21 ^
57 1 25 11 13 12 27 16 14 3 24 20 24 23 22 9 14 19 26 8 13 6 13 25 19
5 1 19 1
17 28 30 13 21 11 4 10 23 25 7 5 11 20 7 24 29 27 13 16 17 1 15 12 29
4 16 27
12 21 ^
65 0 2 2 24 17 13 22 20 22 4 20 16 2 17 3 11 29 1 5 10 14 23 3 14 9 2
7 18 20
13 2 16 25 24 4 4 14 7 13 8 20 12 23 1 26 7 9 19 6 16 3 8 26 21 10 26
23 24 28
16 12 14 25 22 16 15 ^
53 0 21 24 27 30 15 10 15 2 18 13 26 28 23 16 23 10 2 11 26 26 4 29 18
4 24 24
1 24 13 5 21 29 26 2 10 16 6 5 7 23 19 11 28 22 21 1 24 19 7 26 18 30
26 ^

58 0 21 6 18 22 2 1 14 2 14 5 25 1 27 24 6 23 16 5 1 20 29 22 25 9 25
10 3 28
28 25 19 18 16 24 14 15 5 28 12 28 26 29 2 15 15 9 5 18 19 22 12 15 4
6 15 24
16 22 ^
59 0 4 26 25 18 27 19 20 4 26 12 3 22 1 22 30 3 28 10 9 24 14 29 6 30
3 10 20
14 6 3 19 21 21 28 16 18 11 30 11 20 30 1 9 8 11 5 19 10 24 4 22 4 2
26 5 15 20
8 25 ^
55 1 13 30 18 8 1 25 28 20 20 15 21 18 18 12 16 13 24 9 21 2 28 6 1 23
9 18 27
27 4 9 13 10 8 14 16 15 12 11 14 21 14 10 11 25 17 17 30 21 13 27 26
26 22 14
21 ^
57 0 17 21 19 9 9 20 23 13 7 10 28 24 10 22 27 23 27 8 17 14 6 4 21 26
15 1 8
29 27 6 28 15 3 27 25 25 14 19 13 29 8 24 2 8 2 4 12 19 11 10 6 26 14
22 24 30
2 ^
56 1 11 12 2 12 17 23 8 8 12 28 13 14 28 2 17 4 8 3 26 9 23 21 30 30
20 4 13 28
29 9 3 17 7 19 30 28 1 2 20 9 12 24 15 30 20 27 3 23 11 6 29 25 23 26
17 23 ^
61 1 10 22 15 23 6 25 5 4 30 2 19 23 15 27 14 26 1 1 7 19 12 7 6 20 18
14 4 15
17 28 7 11 7 8 9 22 17 12 5 23 18 25 18 6 12 26 30 12 30 14 3 1 18 10
20 27 21
8 6 24 30 ^
54 0 20 11 24 7 2 4 18 15 14 30 16 19 2 27 15 4 19 25 29 29 7 14 18 9
11 9 27
11 15 29 9 28 20 2 30 26 21 17 8 28 17 22 29 24 8 11 18 29 15 6 7 27
27 7 ^
63 1 24 18 23 11 19 8 30 5 24 22 12 25 15 28 23 5 10 21 5 8 7 3 10 19
17 6 9 15
29 10 7 4 1 16 21 16 29 13 18 5 3 8 15 8 21 29 20 5 27 2 13 27 7 7 30
2 18 26
10 2 5 29 12 ^
63 1 15 25 26 24 8 12 20 3 9 10 30 7 12 24 5 13 1 6 10 25 13 9 29 30
17 29 17
26 6 25 4 7 25 8 15 26 30 11 2 9 6 22 6 21 28 3 3 6 12 13 15 7 17 16
26 3 19 3
5 2 29 16 17 ^
60 1 22 26 13 12 9 19 16 27 25 16 7 15 22 22 5 6 13 4 12 11 27 5 19 28
9 30 9 1
28 3 21 22 7 17 2 25 1 28 21 11 17 20 28 12 10 10 3 15 13 20 1 20 11 1
30 22 23
15 26 6 ^
56 0 21 7 27 9 27 29 5 26 16 11 23 28 18 1 13 23 30 6 3 11 11 16 10 13
13 19 15
18 7 22 10 29 3 25 30 8 22 13 4 4 27 22 18 21 23 17 15 16 19 20 17 10
17 21 20

11 ^
64 1 14 14 1 10 18 4 5 5 9 17 18 14 5 12 22 22 11 8 18 24 21 3 7 21 9
4 3 4 23
14 17 25 27 10 5 16 30 23 2 4 25 3 16 28 4 6 6 17 12 18 2 23 7 29 23
29 27 19 7
20 21 25 5 30 ^
63 1 9 7 1 21 20 19 29 19 14 8 23 23 19 3 18 17 3 29 14 27 23 28 5 22
8 10 1 1
11 27 18 3 2 22 28 14 24 12 1 3 30 29 11 22 19 7 16 1 11 4 28 8 20 25
2 6 8 5
14 11 21 9 29 ^
57 0 8 25 6 26 27 1 29 12 8 16 4 17 5 30 30 24 12 19 24 25 11 13 12 27
16 14 3
24 20 24 23 22 9 14 19 26 8 13 6 13 25 19 5 1 19 1 17 28 30 13 21 11 4
10 23 25
6 ^
68 0 5 11 20 7 24 29 27 13 16 17 1 15 12 29 4 16 27 12 6 2 2 24 17 13
22 20 22
4 20 16 2 17 3 11 29 1 5 10 14 23 3 14 9 2 7 18 20 13 2 16 25 24 4 4
14 7 13 8
20 12 23 1 26 7 9 19 6 30 ^
54 1 3 8 26 21 10 26 23 24 28 16 12 14 25 22 16 2 21 24 27 30 15 10 15
2 18 13
26 28 23 16 23 10 2 11 26 26 4 29 18 4 24 24 1 24 13 5 21 29 26 2 10
16 6 27 ^
56 0 7 23 19 11 28 22 21 1 24 19 7 26 18 30 25 21 6 18 22 2 1 14 2 14
5 25 1 27
24 6 23 16 5 1 20 29 22 25 9 25 10 3 28 28 25 19 18 16 24 14 15 5 28
12 28 9 ^
58 0 29 2 15 15 9 5 18 19 22 12 15 4 6 15 24 16 9 4 26 25 18 27 19 20
4 26 12 3
22 1 22 30 3 28 10 9 24 14 29 6 30 3 10 20 14 6 3 19 21 21 28 16 18 11
30 11 20
29 ^
65 0 1 9 8 11 5 19 10 24 4 22 4 2 26 5 15 20 8 3 13 30 18 8 1 25 28 20
20 15 21
18 18 12 16 13 24 9 21 2 28 6 1 23 9 18 27 27 4 9 13 10 8 14 16 15 12
11 14 21
14 10 11 25 17 17 20 ^
55 0 21 13 27 26 26 22 14 13 17 21 19 9 9 20 23 13 7 10 28 24 10 22 27
23 27 8
17 14 6 4 21 26 15 1 8 29 27 6 28 15 3 27 25 25 14 19 13 29 8 24 2 8 2
4 30 ^
57 1 19 11 10 6 26 14 22 24 30 10 11 12 2 12 17 23 8 8 12 28 13 14 28
2 17 4 8
3 26 9 23 21 30 30 20 4 13 28 29 9 3 17 7 19 30 28 1 2 20 9 12 24 15
30 20 27
30 ^
59 0 23 11 6 29 25 23 26 17 20 10 22 15 23 6 25 5 4 30 2 19 23 15 27
14 26 1 1
7 19 12 7 6 20 18 14 4 15 17 28 7 11 7 8 9 22 17 12 5 23 18 25 18 6 12
26 30 12

30 18 ^
56 0 3 1 18 10 20 27 21 8 6 24 26 20 11 24 7 2 4 18 15 14 30 16 19 2
27 15 4 19
25 29 29 7 14 18 9 11 9 27 11 15 29 9 28 20 2 30 26 21 17 8 28 17 22
29 24 7 ^
62 0 11 18 29 15 6 7 27 27 17 24 18 23 11 19 8 30 5 24 22 12 25 15 28
23 5 10
21 5 8 7 3 10 19 17 6 9 15 29 10 7 4 1 16 21 16 29 13 18 5 3 8 15 8 21
29 20 5
27 2 13 27 7 ^
62 1 7 30 2 18 26 10 2 5 29 21 15 25 26 24 8 12 20 3 9 10 30 7 12 24 5
13 1 6
10 25 13 9 29 30 17 29 17 26 6 25 4 7 25 8 15 26 30 11 2 9 6 22 6 21
28 3 3 6
12 13 15 26 ^
62 0 17 16 26 3 19 3 5 2 29 16 16 22 26 13 12 9 19 16 27 25 16 7 15 22
22 5 6
13 4 12 11 27 5 19 28 9 30 9 1 28 3 21 22 7 17 2 25 1 28 21 11 17 20
28 12 10
10 3 15 13 20 19 ^
56 1 20 11 1 30 22 23 15 26 12 21 7 27 9 27 29 5 26 16 11 23 28 18 1
13 23 30 6
3 11 11 16 10 13 13 19 15 18 7 22 10 29 3 25 30 8 22 13 4 4 27 22 18
21 23 17
22 ^
66 1 16 19 20 17 10 17 21 20 6 14 14 1 10 18 4 5 5 9 17 18 14 5 12 22
22 11 8
18 24 21 3 7 21 9 4 3 4 23 14 17 25 27 10 5 16 30 23 2 4 25 3 16 28 4
6 6 17 12
18 2 23 7 29 23 29 24 ^
62 0 19 7 20 21 25 5 17 9 7 1 21 20 19 29 19 14 8 23 23 19 3 18 17 3
29 14 27
23 28 5 22 8 10 1 1 11 27 18 3 2 22 28 14 24 12 1 3 30 29 11 22 19 7
16 1 11 4
28 8 20 25 7 ^
59 0 6 8 5 14 11 21 9 16 8 25 6 26 27 1 29 12 8 16 4 17 5 30 30 24 12
19 24 25
11 13 12 27 16 14 3 24 20 24 23 22 9 14 19 26 8 13 6 13 25 19 5 1 19 1
17 28 30
13 26 ^
70 0 11 4 10 23 25 7 5 11 20 7 24 29 27 13 16 17 1 15 12 29 4 16 27 12
6 2 2 24
17 13 22 20 22 4 20 16 2 17 3 11 29 1 5 10 14 23 3 14 9 2 7 18 20 13 2
16 25 24
4 4 14 7 13 8 20 12 23 1 26 7 ^
54 1 9 19 6 16 3 8 26 21 10 26 23 24 28 16 12 14 25 22 16 2 21 24 27
30 15 10
15 2 18 13 26 28 23 16 23 10 2 11 26 26 4 29 18 4 24 24 1 24 13 5 21
29 26 27 ^
58 1 10 16 6 5 7 23 19 11 28 22 21 1 24 19 7 26 18 30 25 21 6 18 22 2
1 14 2 14

5 25 1 27 24 6 23 16 5 1 20 29 22 25 9 25 10 3 28 28 25 19 18 16 24 14
15 5 28
28 ^
58 1 28 26 29 2 15 15 9 5 18 19 22 12 15 4 6 15 24 16 9 4 26 25 18 27
19 20 4
26 12 3 22 1 22 30 3 28 10 9 24 14 29 6 30 3 10 20 14 6 3 19 21 21 28
16 18 11
30 22 ^
65 1 20 30 1 9 8 11 5 19 10 24 4 22 4 2 26 5 15 20 8 3 13 30 18 8 1 25
28 20 20
15 21 18 18 12 16 13 24 9 21 2 28 6 1 23 9 18 27 27 4 9 13 10 8 14 16
15 12 11
14 21 14 10 11 25 20 ^
53 0 17 30 21 13 27 26 26 22 14 13 17 21 19 9 9 20 23 13 7 10 28 24 10
22 27 23
27 8 17 14 6 4 21 26 15 1 8 29 27 6 28 15 3 27 25 25 14 19 13 29 8 24
15 ^
61 0 8 2 4 12 19 11 10 6 26 14 22 24 30 10 11 12 2 12 17 23 8 8 12 28
13 14 28
2 17 4 8 3 26 9 23 21 30 30 20 4 13 28 29 9 3 17 7 19 30 28 1 2 20 9
12 24 15
30 20 27 20 ^
60 0 23 11 6 29 25 23 26 17 20 10 22 15 23 6 25 5 4 30 2 19 23 15 27
14 26 1 1
7 19 12 7 6 20 18 14 4 15 17 28 7 11 7 8 9 22 17 12 5 23 18 25 18 6 12
26 30 12
30 14 20 ^
56 0 3 1 18 10 20 27 21 8 6 24 26 20 11 24 7 2 4 18 15 14 30 16 19 2
27 15 4 19
25 29 29 7 14 18 9 11 9 27 11 15 29 9 28 20 2 30 26 21 17 8 28 17 22
29 24 23 ^
62 0 11 18 29 15 6 7 27 27 17 24 18 23 11 19 8 30 5 24 22 12 25 15 28
23 5 10
21 5 8 7 3 10 19 17 6 9 15 29 10 7 4 1 16 21 16 29 13 18 5 3 8 15 8 21
29 20 5
27 2 13 27 23 ^
64 1 7 30 2 18 26 10 2 5 29 21 15 25 26 24 8 12 20 3 9 10 30 7 12 24 5
13 1 6
10 25 13 9 29 30 17 29 17 26 6 25 4 7 25 8 15 26 30 11 2 9 6 22 6 21
28 3 3 6
12 13 15 7 17 18 ^
65 0 26 3 19 3 5 2 29 16 16 22 26 13 12 9 19 16 27 25 16 7 15 22 22 5
6 13 4 12
11 27 5 19 28 9 30 9 1 28 3 21 22 7 17 2 25 1 28 21 11 17 20 28 12 10
10 3 15
13 20 1 20 11 1 30 5 ^
57 0 23 15 26 12 21 7 27 9 27 29 5 26 16 11 23 28 18 1 13 23 30 6 3 11
11 16 10
13 13 19 15 18 7 22 10 29 3 25 30 8 22 13 4 4 27 22 18 21 23 17 15 16
19 20 17
10 25 ^

66 1 21 20 6 14 14 1 10 18 4 5 5 9 17 18 14 5 12 22 22 11 8 18 24 21 3
7 21 9 4
3 4 23 14 17 25 27 10 5 16 30 23 2 4 25 3 16 28 4 6 6 17 12 18 2 23 7
29 23 29
27 19 7 20 21 25 20 ^
65 1 17 9 7 1 21 20 19 29 19 14 8 23 23 19 3 18 17 3 29 14 27 23 28 5
22 8 10 1
1 11 27 18 3 2 22 28 14 24 12 1 3 30 29 11 22 19 7 16 1 11 4 28 8 20
25 2 6 8 5
14 11 21 9 16 28 ^
59 1 25 6 26 27 1 29 12 8 16 4 17 5 30 30 24 12 19 24 25 11 13 12 27
16 14 3 24
20 24 23 22 9 14 19 26 8 13 6 13 25 19 5 1 19 1 17 28 30 13 21 11 4 10
23 25 7
5 11 23 ^
70 0 20 7 24 29 27 13 16 17 1 15 12 29 4 16 27 12 6 2 2 24 17 13 22 20
22 4 20
16 2 17 3 11 29 1 5 10 14 23 3 14 9 2 7 18 20 13 2 16 25 24 4 4 14 7
13 8 20 12
23 1 26 7 9 19 6 16 3 8 26 25 ^
56 0 10 26 23 24 28 16 12 14 25 22 16 2 21 24 27 30 15 10 15 2 18 13
26 28 23
16 23 10 2 11 26 26 4 29 18 4 24 24 1 24 13 5 21 29 26 2 10 16 6 5 7
23 19 11
28 24 ^
59 1 21 1 24 19 7 26 18 30 25 21 6 18 22 2 1 14 2 14 5 25 1 27 24 6 23
16 5 1
20 29 22 25 9 25 10 3 28 28 25 19 18 16 24 14 15 5 28 12 28 26 29 2 15
15 9 5
18 19 13 ^
63 1 12 15 4 6 15 24 16 9 4 26 25 18 27 19 20 4 26 12 3 22 1 22 30 3
28 10 9 24
14 29 6 30 3 10 20 14 6 3 19 21 21 28 16 18 11 30 11 20 30 1 9 8 11 5
19 10 24
4 22 4 2 26 20 ^
59 1 15 20 8 3 13 30 18 8 1 25 28 20 20 15 21 18 18 12 16 13 24 9 21 2
28 6 1
23 9 18 27 27 4 9 13 10 8 14 16 15 12 11 14 21 14 10 11 25 17 17 30 21
13 27 26
26 22 14 23 ^
61 0 16 21 30 25 8 9 18 30 17 4 6 31 26 16 25 18 23 9 20 8 7 25 26 11
27 17 28
2 23 2 12 24 2 8 6 3 12 20 16 4 10 17 30 12 11 5 31 19 19 19 30 13 18
2 3 8 2
17 8 24 28 ^
69 0 22 2 7 26 1 13 26 6 6 6 18 24 7 6 7 30 19 12 22 24 25 3 11 17 3
10 12 11
11 25 7 13 31 4 7 27 13 10 30 16 30 16 29 4 27 5 6 27 8 12 29 9 1 10 9
26 4 19
1 3 3 4 16 13 17 6 16 14 28 ^
62 0 3 20 3 2 1 30 14 14 21 29 13 11 10 29 20 19 26 27 19 6 17 26 26 7
9 13 16

9 31 2 20 28 10 10 4 28 3 24 21 6 29 16 1 22 31 1 24 9 16 22 28 11 7 7
3 17 13
22 4 17 9 27 ^
60 0 21 23 17 24 16 24 11 26 5 31 30 2 24 14 9 23 28 18 4 13 23 4 3 9
9 14 12
13 8 20 12 18 6 21 12 25 3 27 8 21 13 5 26 21 18 19 23 16 12 14 20 17
16 7 16
19 22 9 15 25 ^
64 0 31 7 18 29 5 7 2 5 16 18 10 7 11 21 21 9 8 18 25 19 3 6 19 29 10
3 5 23 15
16 27 26 12 2 19 23 1 22 3 19 28 5 2 9 4 21 11 18 1 23 6 30 23 30 26
20 11 22
19 27 2 16 10 31 ^
60 0 4 24 17 20 25 20 15 8 23 23 20 30 18 16 3 30 15 26 23 28 7 21 8 7
31 31 14
26 18 3 1 26 28 15 25 11 31 3 25 9 21 20 2 6 14 4 9 5 28 8 17 22 1 4 8
7 10 14
19 19 ^
62 0 14 8 27 9 24 26 4 30 11 8 19 5 21 7 2 27 20 16 20 20 22 14 13 16
26 14 10
3 25 22 25 23 21 10 15 15 29 8 13 4 2 13 22 20 7 4 20 31 16 2 2 28 13
19 14 12
23 27 6 2 14 24 ^
62 0 22 6 25 30 29 31 13 14 16 31 12 16 30 5 14 31 11 4 1 1 25 21 13
26 22 21 5
22 14 29 1 21 3 14 30 4 2 29 12 15 23 3 15 5 1 6 23 22 13 1 14 27 25 5
15 6 13
3 22 11 23 21 ^
56 1 24 6 5 20 4 14 3 29 8 29 19 7 29 23 25 28 19 11 15 27 21 14 1 19
20 26 12
7 12 1 18 13 29 28 23 29 14 23 7 1 9 29 24 5 30 18 5 25 25 31 25 13 7
24 25 11 ^
54 0 1 12 19 9 7 6 28 20 14 28 21 19 31 20 20 6 24 18 27 24 4 18 21 1
31 15 1
15 2 27 4 26 25 4 23 19 2 31 22 30 21 22 5 27 12 30 28 28 27 15 18 14
25 23 ^
59 1 17 7 28 11 28 29 30 1 17 12 10 2 18 20 21 2 11 12 5 4 12 25 14 5
5 24 22
18 31 15 22 29 11 3 21 31 21 27 3 28 7 10 25 2 15 30 9 30 7 22 15 9 3
20 24 24
28 14 25 ^
58 0 9 27 14 22 27 31 10 8 14 7 15 7 20 5 26 1 29 7 17 17 8 3 13 27 18
8 31 27
28 22 22 17 19 18 18 11 19 13 25 10 19 6 28 4 31 23 10 18 26 31 5 10
13 12 8 15
19 24 ^
56 1 11 14 15 24 15 12 9 22 16 16 24 13 2 26 24 29 26 15 13 21 24 20
10 10 22
23 13 6 12 28 25 12 2 21 26 28 26 8 16 15 4 5 19 24 12 31 8 30 31 9 28
12 3 26
29 8 ^

65 0 27 10 20 13 30 8 25 1 8 1 5 11 20 9 12 4 29 15 21 25 27 7 14 11 1
11 16 23
8 8 11 1 13 15 28 1 16 5 8 3 24 10 23 24 17 5 13 28 30 5 3 16 6 20 28
31 6 22
10 11 29 25 17 27 23 ^
60 0 26 3 23 14 4 30 27 23 24 21 17 12 21 17 23 4 22 7 5 27 1 20 23 12
26 10 24
31 29 4 11 15 11 6 9 17 18 15 5 17 21 28 6 9 6 8 10 21 16 11 2 23 18
27 29 18 9
11 24 24 ^
58 1 11 15 29 29 3 31 18 12 17 31 19 8 4 25 29 17 29 9 25 6 1 2 18 17
15 19 15
1 26 17 20 27 30 30 6 10 18 5 9 10 26 9 17 2 30 10 28 17 1 27 29 2 19
16 8 28
16 28 ^
61 1 30 25 8 9 18 30 17 4 6 31 26 16 25 18 23 9 20 8 7 25 26 11 27 17
28 2 23 2
12 24 2 8 6 3 12 20 16 4 10 17 30 12 11 5 31 19 19 19 30 13 18 2 3 8 2
17 8 24
30 22 29 ^
70 0 7 26 1 13 26 6 6 6 18 24 7 6 7 30 19 12 22 24 25 3 11 17 3 10 12
11 11 25
7 13 31 4 7 27 13 10 30 16 30 16 29 4 27 5 6 27 8 12 29 9 1 10 9 26 4
19 1 3 3
4 16 13 17 6 16 14 24 3 20 21 ^
62 1 2 1 30 14 14 21 29 13 11 10 29 20 19 26 27 19 6 17 26 26 7 9 13
16 9 31 2
20 28 10 10 4 28 3 24 21 6 29 16 1 22 31 1 24 9 16 22 28 11 7 7 3 17
13 22 4 17
9 4 21 23 21 ^
62 1 24 16 24 11 26 5 31 30 2 24 14 9 23 28 18 4 13 23 4 3 9 9 14 12
13 8 20 12
18 6 21 12 25 3 27 8 21 13 5 26 21 18 19 23 16 12 14 20 17 16 7 16 19
22 9 15
10 31 7 18 29 7 ^
66 1 7 2 5 16 18 10 7 11 21 21 9 8 18 25 19 3 6 19 29 10 3 5 23 15 16
27 26 12
2 19 23 1 22 3 19 28 5 2 9 4 21 11 18 1 23 6 30 23 30 26 20 11 22 19
27 2 16 10
6 4 24 17 20 25 20 21 ^
62 1 8 23 23 20 30 18 16 3 30 15 26 23 28 7 21 8 7 31 31 14 26 18 3 1
26 28 15
25 11 31 3 25 9 21 20 2 6 14 4 9 5 28 8 17 22 1 4 8 7 10 14 19 10 14 8
27 9 24
26 4 30 8 ^
62 1 8 19 5 21 7 2 27 20 16 20 20 22 14 13 16 26 14 10 3 25 22 25 23
21 10 15
15 29 8 13 4 2 13 22 20 7 4 20 31 16 2 2 28 13 19 14 12 23 27 6 2 14 2
22 6 25
30 29 31 13 14 21 ^
65 0 31 12 16 30 5 14 31 11 4 1 1 25 21 13 26 22 21 5 22 14 29 1 21 3
14 30 4 2

29 12 15 23 3 15 5 1 6 23 22 13 1 14 27 25 5 15 6 13 3 22 11 23 31 24
6 5 20 4
14 3 29 8 29 19 31 ^
55 1 29 23 25 28 19 11 15 27 21 14 1 19 20 26 12 7 12 1 18 13 29 28 23
29 14 23
7 1 9 29 24 5 30 18 5 25 25 31 25 13 7 24 25 24 1 12 19 9 7 6 28 20 14
28 27 ^
53 1 19 31 20 20 6 24 18 27 24 4 18 21 1 31 15 1 15 2 27 4 26 25 4 23
19 2 31
22 30 21 22 5 27 12 30 28 28 27 15 18 14 25 15 17 7 28 11 28 29 30 1
17 21 ^
61 0 10 2 18 20 21 2 11 12 5 4 12 25 14 5 5 24 22 18 31 15 22 29 11 3
21 31 21
27 3 28 7 10 25 2 15 30 9 30 7 22 15 9 3 20 24 24 28 14 18 9 27 14 22
27 31 10
8 14 7 15 19 ^
59 1 20 5 26 1 29 7 17 17 8 3 13 27 18 8 31 27 28 22 22 17 19 18 18 11
19 13 25
10 19 6 28 4 31 23 10 18 26 31 5 10 13 12 8 15 19 17 11 14 15 24 15 12
9 22 16
16 24 13 23 ^
57 0 26 24 29 26 15 13 21 24 20 10 10 22 23 13 6 12 28 25 12 2 21 26
28 26 8 16
15 4 5 19 24 12 31 8 30 31 9 28 12 3 26 29 22 27 10 20 13 30 8 25 1 8
1 5 11 20
16 ^
61 1 12 4 29 15 21 25 27 7 14 11 1 11 16 23 8 8 11 1 13 15 28 1 16 5 8
3 24 10
23 24 17 5 13 28 30 5 3 16 6 20 28 31 6 22 10 11 29 25 17 27 22 26 3
23 14 4 30
27 23 24 31 ^
61 1 17 12 21 17 23 4 22 7 5 27 1 20 23 12 26 10 24 31 29 4 11 15 11 6
9 17 18
15 5 17 21 28 6 9 6 8 10 21 16 11 2 23 18 27 29 18 9 11 24 27 11 15 29
29 3 31
18 12 17 31 12 ^
60 1 8 4 25 29 17 29 9 25 6 1 2 18 17 15 19 15 1 26 17 20 27 30 30 6
10 18 5 9
10 26 9 17 2 30 10 28 17 1 27 29 2 19 16 8 28 16 21 30 25 8 9 18 30 17
4 6 31
26 16 18 ^
67 1 18 23 9 20 8 7 25 26 11 27 17 28 2 23 2 12 24 2 8 6 3 12 20 16 4
10 17 30
12 11 5 31 19 19 19 30 13 18 2 3 8 2 17 8 24 30 22 2 7 26 1 13 26 6 6
6 18 24 7
6 7 30 19 12 22 24 28 ^
69 1 3 11 17 3 10 12 11 11 25 7 13 31 4 7 27 13 10 30 16 30 16 29 4 27
5 6 27 8
12 29 9 1 10 9 26 4 19 1 3 3 4 16 13 17 6 16 14 24 3 20 3 2 1 30 14 14
21 29 13
11 10 29 20 19 26 27 19 6 28 ^

62 1 26 26 7 9 13 16 9 31 2 20 28 10 10 4 28 3 24 21 6 29 16 1 22 31 1
24 9 16
22 28 11 7 7 3 17 13 22 4 17 9 4 21 23 17 24 16 24 11 26 5 31 30 2 24
14 9 23
28 18 4 13 26 ^
70 1 4 3 9 9 14 12 13 8 20 12 18 6 21 12 25 3 27 8 21 13 5 26 21 18 19
23 16 12
14 20 17 16 7 16 19 22 9 15 10 31 7 18 29 5 7 2 5 16 18 10 7 11 21 21
9 8 18 25
19 3 6 19 29 10 3 5 23 15 16 17 ^
60 1 26 12 2 19 23 1 22 3 19 28 5 2 9 4 21 11 18 1 23 6 30 23 30 26 20
11 22 19
27 2 16 10 6 4 24 17 20 25 20 15 8 23 23 20 30 18 16 3 30 15 26 23 28
7 21 8 7
31 31 7 ^
66 0 26 18 3 1 26 28 15 25 11 31 3 25 9 21 20 2 6 14 4 9 5 28 8 17 22
1 4 8 7
10 14 19 10 14 8 27 9 24 26 4 30 11 8 19 5 21 7 2 27 20 16 20 20 22 14
13 16 26
14 10 3 25 22 25 23 17 ^
64 1 10 15 15 29 8 13 4 2 13 22 20 7 4 20 31 16 2 2 28 13 19 14 12 23
27 6 2 14
2 22 6 25 30 29 31 13 14 16 31 12 16 30 5 14 31 11 4 1 1 25 21 13 26
22 21 5 22
14 29 1 21 3 14 27 ^
64 0 4 2 29 12 15 23 3 15 5 1 6 23 22 13 1 14 27 25 5 15 6 13 3 22 11
23 31 24
6 5 20 4 14 3 29 8 29 19 7 29 23 25 28 19 11 15 27 21 14 1 19 20 26 12
7 12 1
18 13 29 28 23 29 13 ^
58 0 23 7 1 9 29 24 5 30 18 5 25 25 31 25 13 7 24 25 24 1 12 19 9 7 6
28 20 14
28 21 19 31 20 20 6 24 18 27 24 4 18 21 1 31 15 1 15 2 27 4 26 25 4 23
19 2 31
28 ^
57 0 30 21 22 5 27 12 30 28 28 27 15 18 14 25 15 17 7 28 11 28 29 30 1
17 12 10
2 18 20 21 2 11 12 5 4 12 25 14 5 5 24 22 18 31 15 22 29 11 3 21 31 21
27 3 28
7 26 ^
59 0 25 2 15 30 9 30 7 22 15 9 3 20 24 24 28 14 18 9 27 14 22 27 31 10
8 14 7
15 7 20 5 26 1 29 7 17 17 8 3 13 27 18 8 31 27 28 22 22 17 19 18 18 11
19 13 25
10 19 19 ^
59 0 28 4 31 23 10 18 26 31 5 10 13 12 8 15 19 17 11 14 15 24 15 12 9
22 16 16
24 13 2 26 24 29 26 15 13 21 24 20 10 10 22 23 13 6 12 28 25 12 2 21
26 28 26 8
16 15 4 5 31 ^
66 1 24 12 31 8 30 31 9 28 12 3 26 29 22 27 10 20 13 30 8 25 1 8 1 5
11 20 9 12

4 29 15 21 25 27 7 14 11 1 11 16 23 8 8 11 1 13 15 28 1 16 5 8 3 24 10
23 24 17
5 13 28 30 5 3 16 21 ^
59 0 20 28 31 6 22 10 11 29 25 17 27 22 26 3 23 14 4 30 27 23 24 21 17
12 21 17
23 4 22 7 5 27 1 20 23 12 26 10 24 31 29 4 11 15 11 6 9 17 18 15 5 17
21 28 6 9
6 8 26 ^
59 0 21 16 11 2 23 18 27 29 18 9 11 24 27 11 15 29 29 3 31 18 12 17 31
19 8 4
25 29 17 29 9 25 6 1 2 18 17 15 19 15 1 26 17 20 27 30 30 6 10 18 5 9
10 26 9
17 2 30 24 ^
61 0 28 17 1 27 29 2 19 16 8 28 16 21 30 25 8 9 18 30 17 4 6 31 26 16
25 18 23
9 20 8 7 25 26 11 27 17 28 2 23 2 12 24 2 8 6 3 12 20 16 4 10 17 30 12
11 5 31
19 19 19 25 ^
74 0 13 18 2 3 8 2 17 8 24 30 22 2 7 26 1 13 26 6 6 6 18 24 7 6 7 30
19 12 22
24 25 3 11 17 3 10 12 11 11 25 7 13 31 4 7 27 13 10 30 16 30 16 29 4
27 5 6 27
8 12 29 9 1 10 9 26 4 19 1 3 3 4 16 16 ^
64 1 17 6 16 14 24 3 20 3 2 1 30 14 14 21 29 13 11 10 29 20 19 26 27
19 6 17 26
26 7 9 13 16 9 31 2 20 28 10 10 4 28 3 24 21 6 29 16 1 22 31 1 24 9 16
22 28 11
7 7 3 17 13 22 27 ^
65 0 17 9 4 21 23 17 24 16 24 11 26 5 31 30 2 24 14 9 23 28 18 4 13 23
4 3 9 9
14 12 13 8 20 12 18 6 21 12 25 3 27 8 21 13 5 26 21 18 19 23 16 12 14
20 17 16
7 16 19 22 9 15 10 31 1 ^
68 1 18 29 5 7 2 5 16 18 10 7 11 21 21 9 8 18 25 19 3 6 19 29 10 3 5
23 15 16
27 26 12 2 19 23 1 22 3 19 28 5 2 9 4 21 11 18 1 23 6 30 23 30 26 20
11 22 19
27 2 16 10 6 4 24 17 20 25 20 ^
63 0 15 8 23 23 20 30 18 16 3 30 15 26 23 28 7 21 8 7 31 31 14 26 18 3
1 26 28
15 25 11 31 3 25 9 21 20 2 6 14 4 9 5 28 8 17 22 1 4 8 7 10 14 19 10
14 8 27 9
24 26 4 30 24 ^
64 1 8 19 5 21 7 2 27 20 16 20 20 22 14 13 16 26 14 10 3 25 22 25 23
21 10 15
15 29 8 13 4 2 13 22 20 7 4 20 31 16 2 2 28 13 19 14 12 23 27 6 2 14 2
22 6 25
30 29 31 13 14 16 31 5 ^
67 0 16 30 5 14 31 11 4 1 1 25 21 13 26 22 21 5 22 14 29 1 21 3 14 30
4 2 29 12
15 23 3 15 5 1 6 23 22 13 1 14 27 25 5 15 6 13 3 22 11 23 31 24 6 5 20
4 14 3

29 8 29 19 7 29 23 25 21 ^
58 0 19 11 15 27 21 14 1 19 20 26 12 7 12 1 18 13 29 28 23 29 14 23 7
1 9 29 24
5 30 18 5 25 25 31 25 13 7 24 25 24 1 12 19 9 7 6 28 20 14 28 21 19 31
20 20 6
24 22 ^
60 0 27 24 4 18 21 1 31 15 1 15 2 27 4 26 25 4 23 19 2 31 22 30 21 22
5 27 12
30 28 28 27 15 18 14 25 15 17 7 28 11 28 29 30 1 17 12 10 2 18 20 21 2
11 12 5
4 12 25 14 22 ^
60 1 5 24 22 18 31 15 22 29 11 3 21 31 21 27 3 28 7 10 25 2 15 30 9 30
7 22 15
9 3 20 24 24 28 14 18 9 27 14 22 27 31 10 8 14 7 15 7 20 5 26 1 29 7
17 17 8 3
13 27 31 ^
58 0 8 31 27 28 22 22 17 19 18 18 11 19 13 25 10 19 6 28 4 31 23 10 18
26 31 5
10 13 12 8 15 19 17 11 14 15 24 15 12 9 22 16 16 24 13 2 26 24 29 26
15 13 21
24 20 10 10 25 ^
65 0 23 13 6 12 28 25 12 2 21 26 28 26 8 16 15 4 5 19 24 12 31 8 30 31
9 28 12
3 26 29 22 27 10 20 13 30 8 25 1 8 1 5 11 20 9 12 4 29 15 21 25 27 7
14 11 1 11
16 23 8 8 11 1 13 21 ^
60 1 28 1 16 5 8 3 24 10 23 24 17 5 13 28 30 5 3 16 6 20 28 31 6 22 10
11 29 25
17 27 22 26 3 23 14 4 30 27 23 24 21 17 12 21 17 23 4 22 7 5 27 1 20
23 12 26
10 24 31 11 ^
64 1 4 11 15 11 6 9 17 18 15 5 17 21 28 6 9 6 8 10 21 16 11 2 23 18 27
29 18 9
11 24 27 11 15 29 29 3 31 18 12 17 31 19 8 4 25 29 17 29 9 25 6 1 2 18
17 15 19
15 1 26 17 20 27 25 ^
64 0 30 6 10 18 5 9 10 26 9 17 2 30 10 28 17 1 27 29 2 19 16 8 28 16
21 30 25 8
9 18 30 17 4 6 31 26 16 25 18 23 9 20 8 7 25 26 11 27 17 28 2 23 2 12
24 2 8 6
3 12 20 16 4 31 ^
60 0 28 5 24 16 10 27 28 19 29 19 22 10 23 17 25 7 26 5 5 27 1 20 24
13 27 16
24 32 31 5 12 21 13 6 11 21 20 15 3 14 18 32 2 9 6 10 13 23 16 17 3 20
19 27 29
21 5 11 28 14 ^
<D

H>SHS Type 2 Strings<H

D>

60 1 16 13 19 5 16 14 27 2 22 7 5 28 15 18 25 17 13 12 29 24 21 29 25
16 5 17

22 24 8 8 18 3 13 10 28 3 23 28 12 8 3 25 5 25 24 7 14 30 27 32 30 23
8 15 27
28 13 8 12 21 ^
86 1 33 5 40 26 3 19 12 36 43 5 35 37 5 14 11 45 35 16 10 8 32 4 15 35
26 2 39
22 37 22 30 29 28 36 45 14 36 4 29 37 30 34 36 20 9 42 29 9 11 11 18
42 13 6 20
42 20 30 22 35 34 6 14 41 19 30 17 35 33 26 41 15 5 44 39 8 20 3 44 32
15 18 10
15 10 36 ^
112 0 10 44 24 53 35 22 40 20 15 51 22 18 22 42 6 54 49 38 21 7 13 30
16 7 52
16 22 13 38 7 11 44 33 9 25 13 37 42 14 45 53 30 38 5 25 5 35 38 22 28
10 29 52
31 39 35 23 5 17 14 31 46 5 41 54 43 33 13 23 46 45 3 46 3 55 15 6 53
28 52 22
27 45 32 53 51 15 37 15 49 36 35 35 15 3 13 22 25 3 4 10 24 12 7 44 25
18 21 21
20 55 24 ^
123 1 43 9 1 52 4 21 49 61 18 50 23 13 46 62 23 45 62 9 56 18 23 31 8
30 27 36
13 38 4 58 53 47 24 18 41 58 19 12 18 52 42 29 44 45 26 63 34 32 41 64
15 26 55
19 2 49 6 30 53 13 54 12 53 37 12 37 29 60 53 21 47 42 45 52 40 10 28
47 43 17
16 17 61 36 12 34 60 37 13 52 58 37 39 29 34 11 45 8 36 1 50 50 5 51
13 15 32
51 33 61 16 31 50 58 63 16 5 30 17 24 10 26 52 ^
144 1 64 62 63 30 15 1 35 28 16 40 20 14 50 33 19 38 30 27 55 10 16 46
47 7 55
12 53 26 56 33 29 55 25 17 48 43 21 43 18 24 63 27 68 46 38 33 35 10
18 11 27 5
9 58 35 70 36 36 39 47 2 10 66 47 5 18 21 44 71 51 57 3 22 7 56 55 28
25 14 40
16 24 48 37 66 50 24 45 18 39 53 29 64 60 26 50 6 47 61 46 44 60 17 4
58 45 26
20 26 32 8 7 68 37 3 24 44 47 64 54 71 57 12 68 27 58 29 54 43 42 60
13 14 56
49 1 32 9 61 43 36 33 20 59 ^
166 0 7 38 57 33 61 73 7 64 1 49 35 76 14 27 21 45 68 38 58 73 13 72
47 73 33 8
66 23 38 4 56 77 47 10 71 13 20 31 41 6 51 3 18 17 61 47 14 48 76 46
28 34 43 1
56 4 25 7 65 41 1 34 37 23 59 59 27 26 13 15 14 75 60 14 1 28 59 26 65
61 16 23
17 28 6 19 2 35 49 30 29 48 2 63 73 59 1 3 76 41 11 19 18 43 54 63 67
51 4 9 78
60 8 72 31 35 77 54 58 20 36 58 51 76 40 20 37 49 3 75 1 29 20 57 64
22 25 6 25
59 32 43 58 40 51 77 1 2 36 10 12 28 39 11 46 2 41 27 27 53 76 13 66
69 76 28 ^

186 1 5 73 38 30 30 48 21 75 80 40 21 8 53 9 26 30 34 81 71 71 51 23
75 33 41
23 32 5 8 66 40 72 40 16 66 45 14 48 34 21 41 27 3 55 27 37 23 41 65 4
57 51 74
22 19 75 42 16 19 46 16 10 48 20 19 37 41 14 57 9 17 55 38 5 60 7 46
20 43 36
39 52 20 10 62 45 23 46 7 35 75 29 70 35 36 34 25 12 15 84 26 10 6 71
29 79 33
32 25 59 76 82 64 58 7 8 19 41 74 2 53 65 24 1 55 51 36 21 79 28 5 17
41 36 63
51 18 48 25 46 45 36 77 76 22 57 14 44 63 56 45 56 20 1 64 49 9 12 28
30 72 82
73 33 3 22 30 43 56 67 62 2 57 13 50 20 57 38 43 61 9 8 57 65 76 9 ^
183 1 54 34 83 10 33 51 86 81 82 69 18 8 22 64 19 86 62 58 33 37 17 34
5 29 83
42 76 50 54 66 39 9 1 36 43 17 65 6 35 56 72 71 83 88 10 1 8 87 22 6
21 78 25
89 43 62 40 55 85 31 89 74 63 46 28 24 26 31 17 7 8 27 19 12 85 17 20
27 77 10
2 54 80 17 52 74 76 69 78 11 20 80 4 29 24 85 75 18 39 23 70 83 29 57
67 72 70
33 4 15 46 42 2 69 13 53 33 69 64 33 64 14 40 69 59 78 71 19 85 16 65
38 82 55
40 56 40 24 84 29 25 44 68 13 85 76 64 32 18 25 45 23 6 80 31 30 30 58
14 13 72
57 14 41 28 52 73 26 76 83 56 54 2 36 11 65 66 38 50 8 41 88 77 ^
200 1 25 57 40 46 78 57 34 78 61 36 66 57 38 80 22 32 68 71 30 74 37
81 66 77
66 55 2 51 24 93 61 40 68 45 61 12 63 24 89 59 52 72 43 20 20 69 36 40
88 46 9
62 55 77 84 20 18 6 77 15 52 39 75 3 26 4 85 17 62 29 11 92 46 58 29
59 28 42
80 71 96 2 49 85 37 63 4 61 14 2 53 87 25 86 6 75 76 93 41 39 93 92 42
56 41 63
26 28 18 77 11 50 78 79 1 12 12 91 29 13 58 5 56 92 66 59 4 39 47 95 5
5 62 33
13 80 69 60 43 51 22 30 32 42 5 38 3 50 7 48 90 56 96 68 47 41 61 60
96 20 14 4
19 19 23 25 13 9 38 52 17 11 96 48 19 25 88 42 7 74 59 48 70 55 24 18
28 44 46
3 24 38 56 19 96 22 50 60 17 96 ^
195 1 25 51 45 35 66 19 61 60 9 31 93 64 70 30 42 86 53 1 71 46 42 22
38 96 10
99 34 76 26 55 73 63 63 97 23 92 81 64 46 1 30 31 35 86 91 88 64 87 16
37 69 84
94 60 100 3 47 52 8 71 87 57 29 76 43 18 45 46 15 65 12 44 42 66 60 15
68 19 58
39 62 76 9 92 101 57 32 4 34 15 41 62 32 89 71 43 35 31 41 21 17 82 33
96 27 62
29 82 57 46 62 15 24 99 37 83 40 52 46 56 80 98 3 91 74 6 27 7 58 94
10 41 79

97 84 77 74 26 99 95 61 19 17 24 55 15 85 84 30 45 93 44 100 44 39 59
46 18 95
3 41 15 82 77 16 48 18 51 83 20 82 60 53 53 12 99 92 54 5 99 51 39 100
61 2 31
52 49 86 94 16 78 84 91 42 ^
212 0 56 98 16 98 78 22 72 33 103 104 52 84 12 65 15 85 101 97 84 31
51 26 100
100 38 80 13 2 78 7 24 44 84 103 27 7 28 16 33 99 25 103 54 14 42 62
87 92 27
22 42 5 52 100 84 73 72 63 24 48 56 52 23 5 17 76 31 1 95 58 43 60 50
62 30 23
35 79 20 35 3 72 32 45 51 87 41 84 27 79 77 70 102 15 54 15 100 8 52
69 105 3
30 84 42 93 66 89 69 74 24 33 42 97 4 38 99 106 13 93 6 106 74 100 54
45 21 59
56 37 9 50 32 75 79 31 77 9 61 1 8 68 6 60 81 7 100 99 14 61 48 25 73
26 70 72
94 3 11 98 31 5 67 7 39 92 73 88 101 32 105 34 82 81 11 72 48 10 59 71
51 71 35
33 27 49 49 19 98 47 43 75 64 17 15 24 94 49 4 84 58 58 74 39 47 104
82 5 63 69
90 19 ^
236 1 29 79 41 40 72 51 12 92 34 52 44 69 104 21 97 89 96 48 21 4 61
40 28 67
34 23 85 44 22 62 52 33 84 23 30 73 74 4 79 12 81 47 80 53 47 89 40 19
80 62 34
61 29 41 95 43 1 70 63 55 53 18 19 13 48 10 19 89 49 4 52 53 56 76 10
8 104 77
15 28 38 75 109 3 85 90 8 40 8 93 90 43 39 14 60 17 36 78 56 105 80 35
75 36 58
82 50 100 98 45 74 13 66 95 72 71 95 34 14 98 72 33 38 37 52 6 14 107
59 3 29
61 67 98 92 5 93 17 98 36 87 41 75 71 57 88 17 25 91 84 3 58 20 92 69
51 50 36
31 14 25 18 30 18 1 41 104 30 82 59 87 70 34 96 28 47 62 81 103 109
103 100 7
17 16 104 1 12 34 5 102 42 16 22 71 39 5 73 45 45 54 41 13 2 26 41 22
100 109
30 44 4 90 108 53 38 104 68 28 56 47 51 94 22 7 81 85 26 50 9 72 98 54
59 96 52
^
232 1 92 72 47 37 33 38 92 17 8 28 88 22 62 69 32 89 75 3 72 96 85 13
105 24 38
37 94 115 83 72 108 114 24 93 76 103 60 99 102 9 43 10 59 95 46 33 93
15 26 69
44 2 86 107 55 45 61 65 92 66 9 55 39 70 83 29 98 67 13 111 15 20 31
62 8 2 51
20 19 33 44 14 115 71 112 97 10 41 28 53 51 26 57 15 38 98 55 106 22
56 31 50
95 107 110 84 70 10 108 96 73 100 25 36 55 88 71 63 96 30 90 96 79 22
7 30 23

28 59 89 8 51 99 47 86 34 18 43 65 98 104 107 49 7 79 71 8 57 21 29 80
2 74 78
44 57 9 61 22 13 68 52 91 74 98 43 30 58 68 95 101 72 102 76 42 99 108
47 25 73
27 104 34 59 52 36 104 20 98 14 97 4 50 46 106 83 17 22 105 79 77 46
39 15 113
20 37 100 106 65 66 76 7 12 72 35 8 22 50 40 68 101 23 54 96 44 51 87
18 6 46
110 ^
245 0 70 30 81 46 53 119 85 6 104 47 92 72 70 5 70 15 115 68 105 33 97
13 85
106 14 61 29 22 86 45 57 69 91 38 38 28 66 13 60 95 103 3 15 5 113 38
23 62 5
65 94 107 73 104 37 47 102 117 3 78 35 7 95 56 78 45 52 28 46 43 37 32
53 19 55
29 47 97 76 115 83 71 11 45 62 73 99 116 2 24 116 7 28 41 2 29 37 52
23 5 118
79 31 57 89 61 24 101 78 50 93 73 41 7 33 45 47 24 1 48 73 36 3 25 87
46 28 108
54 68 53 67 119 28 36 118 104 42 88 27 112 4 74 85 1 63 39 97 71 74 75
76 10 49
12 79 11 50 103 118 94 117 118 37 27 12 94 60 28 51 47 82 110 17 15
105 23 52
43 12 21 22 81 41 12 74 90 42 108 117 98 67 4 69 44 76 105 38 101 21
80 70 10
111 23 110 32 101 83 91 101 31 5 3 83 76 15 90 57 74 95 46 33 112 62
63 54 63
49 59 4 64 40 68 110 74 105 91 86 102 27 55 34 88 55 ^
240 0 55 79 75 84 113 22 4 113 109 31 33 17 96 11 29 63 98 103 107 116
34 14 9
95 38 18 51 75 33 109 118 55 66 4 76 7 75 70 82 74 23 1 26 69 40 112
99 47 65
31 70 119 52 103 88 85 86 28 16 12 76 25 22 78 64 21 86 27 61 77 72
108 2 18
106 119 121 54 16 85 72 2 73 26 88 66 60 80 35 24 117 63 24 44 67 52
122 119 33
72 16 99 98 69 54 19 42 28 53 114 32 117 81 100 57 49 123 56 21 68 80
53 95 1
45 95 107 98 87 1 27 24 99 116 16 67 1 113 91 84 25 40 25 72 3 28 90
87 112 80
16 117 45 77 36 90 105 59 88 122 64 108 108 71 98 18 50 115 93 105 77
35 6 46
55 47 102 4 26 87 111 120 81 113 4 57 105 3 84 94 115 61 3 121 72 71
24 62 14
60 99 61 96 103 51 107 6 65 81 114 83 121 109 49 104 30 73 82 92 94 9
8 11 98
19 10 55 97 22 97 95 88 67 15 20 39 65 13 123 77 ^
253 1 45 7 15 24 51 5 98 115 24 49 90 104 117 66 128 94 64 80 12 43 91
46 111
59 58 77 30 14 88 60 123 68 41 44 68 40 104 118 41 43 93 90 105 92 16
127 26 54

125 114 79 71 24 48 21 25 118 40 103 49 91 44 67 65 25 119 109 18 48
23 69 112
38 61 64 87 84 104 119 110 122 92 22 1 8 83 34 100 32 62 41 46 112 34
102 76 56
39 4 127 30 13 19 110 124 7 16 128 95 4 124 11 104 116 126 49 95 3 55
96 70 90
101 4 122 96 75 118 39 128 99 92 18 42 20 87 83 35 75 111 61 67 71 28
101 9 56
34 105 95 71 23 73 71 26 57 15 23 76 55 99 89 128 98 117 68 43 88 62
38 62 39 2
83 36 15 26 60 128 96 73 74 10 1 12 42 22 2 77 33 33 32 57 13 14 82 57
12 39 3
58 80 14 87 85 44 69 109 19 117 67 68 44 63 106 37 88 64 82 56 37 89 4
126 55
98 114 84 99 91 43 92 21 26 58 84 102 75 116 124 5 101 5 97 46 109 15
29 79 72
93 92 110 ^
289 1 83 5 76 20 32 15 10 1 103 18 22 116 98 9 51 104 102 44 33 15 12
24 31 89
1 6 28 101 8 64 72 106 30 5 52 89 111 39 108 64 85 17 57 124 22 105 78
115 3 40
108 66 108 77 128 103 44 35 38 13 95 10 111 63 98 117 61 51 126 69 96
70 70 59
39 13 97 33 112 2 77 7 123 70 83 29 66 67 49 79 19 104 115 14 60 2 55
40 71 33
28 114 51 91 17 46 45 128 57 87 62 25 115 38 50 55 90 74 8 51 102 79
43 94 36
122 94 12 41 36 25 104 91 24 7 99 80 30 126 32 63 122 107 114 27 28 79
41 12 35
51 115 122 70 22 79 65 2 88 27 17 59 15 23 44 57 5 65 6 26 78 80 125
93 84 100
45 22 129 68 36 111 74 118 11 50 42 120 47 21 8 86 112 26 67 60 99 45
93 47 8
38 59 52 56 124 20 82 18 117 24 18 46 106 19 117 26 41 47 45 130 7 15
1 4 5 100
10 85 50 44 11 48 92 119 108 42 118 24 20 69 107 90 96 48 103 7 2 90
50 18 130
126 45 106 17 35 25 102 68 23 68 120 58 41 34 75 103 20 45 117 37 61
73 34 62 1
125 58 74 21 21 129 7 86 110 76 66 124 ^
263 1 100 135 7 75 23 5 81 110 31 118 29 1 62 11 41 88 109 119 102 37
3 30 123
47 31 56 134 29 124 116 118 99 21 56 77 91 23 37 135 81 44 51 67 95 51
133 30
57 67 116 122 48 100 7 132 97 106 69 93 4 95 125 102 103 119 81 57 133
96 37
118 50 117 113 81 127 17 45 103 32 121 129 60 43 65 127 30 36 132 110
52 53 35
71 12 76 22 72 130 112 99 76 26 21 73 63 63 97 23 58 115 132 114 1 132
31 35 18
23 54 30 53 118 37 35 84 94 60 100 3 47 18 110 105 87 57 63 76 43 52
45 46 49

65 12 10 42 66 60 117 34 19 92 5 28 76 9 126 101 125 32 38 34 15 7 62
32 21 3
43 69 31 109 123 51 116 135 130 129 130 63 14 57 80 62 15 126 31 105
83 108 120
80 124 46 98 105 91 6 6 27 7 58 128 78 7 79 63 84 77 74 128 65 61 95
121 17 24
123 117 51 50 64 79 59 44 134 78 73 59 114 18 27 3 109 49 82 43 118
116 52 119
15 122 82 26 87 53 114 133 92 88 39 99 17 107 134 95 104 31 52 49 94 ^
286 0 120 16 104 136 117 82 138 32 65 114 119 137 121 8 12 46 126 26
119 73 130
60 76 113 100 14 133 26 116 34 120 80 95 84 53 15 24 44 51 4 10 23 77
24 99 66
37 54 63 42 136 21 34 76 5 17 128 101 1 59 40 113 112 32 97 31 93 105
79 91 18
39 1 103 132 51 68 124 111 13 97 43 128 69 84 85 72 15 12 26 87 16 16
92 101 13
77 4 118 89 103 56 42 16 60 44 39 126 46 18 83 93 41 105 3 82 106 115
91 6 4 54
115 15 120 109 113 48 41 9 95 20 62 67 105 111 25 132 7 116 46 138 44
83 61 124
131 35 107 6 109 81 114 67 41 137 77 56 74 73 34 12 14 69 52 11 98 47
54 83 81
6 1 15 88 35 139 80 83 49 89 27 47 130 92 133 87 51 112 76 49 109 49
57 93 73
22 117 50 64 58 97 139 36 131 111 133 58 33 8 88 55 38 90 46 30 118 57
29 82 74
41 117 38 46 94 92 5 105 15 117 70 103 68 60 120 48 21 110 85 40 81 38
58 129
56 10 99 86 76 81 26 23 25 68 16 28 127 65 18 71 75 77 100 26 11 130
104 28 34
56 106 7 8 120 112 12 58 21 130 124 67 61 19 87 119 ^
286 1 35 94 61 130 98 35 88 34 65 104 56 126 118 50 87 10 81 109 90 86
118 32 6
114 88 39 38 39 62 3 12 134 72 137 35 75 81 115 106 140 112 11 123 41
103 45 95
84 71 107 13 26 110 96 62 16 109 84 59 53 38 27 8 28 13 32 137 17 138
41 122 36
99 65 99 83 36 112 29 49 70 96 126 136 131 116 3 18 17 126 142 14 37
141 141
123 42 13 20 83 42 139 83 54 49 58 42 7 137 29 48 16 121 127 34 52 140
106 128
58 36 124 83 24 69 54 61 112 17 6 95 97 24 57 86 124 59 71 119 67 1
109 54 68
49 57 132 32 5 71 113 40 80 104 75 106 133 31 126 130 104 62 9 39 44
66 116 141
135 96 132 19 41 121 126 124 77 8 4 60 82 6 101 124 89 51 123 48 40 85
77 21
112 10 69 66 115 87 16 108 30 84 65 80 103 32 131 134 73 47 10 63 39
50 93 37
135 114 69 48 34 58 23 27 133 37 9 40 98 41 115 99 70 83 29 42 67 133
55 79 80

91 122 12 2 115 112 23 33 16 102 3 19 60 101 130 105 116 117 27 2 1
103 38 14
55 78 26 116 123 54 67 139 82 74 70 84 72 17 132 25 68 31 120 103 140
^
285 1 65 30 51 119 48 107 92 84 69 28 136 143 54 20 6 70 47 142 64 4
65 59 73
99 134 146 102 125 116 57 137 137 72 48 128 78 5 80 63 54 85 30 22 129
68 21 21
74 28 128 107 27 60 2 93 95 71 37 11 37 15 39 102 3 104 65 80 59 52
113 34 20
67 60 27 81 135 46 106 106 102 68 128 17 15 100 124 15 43 136 122 100
67 142 35
14 53 120 2 89 93 99 73 9 122 39 77 15 96 90 43 79 134 60 92 105 55 96
31 119
77 97 72 23 140 38 30 43 83 136 88 107 117 72 109 118 58 91 119 73 95
100 59
138 123 54 49 143 50 133 66 106 45 80 88 42 93 5 59 77 101 74 110 104
40 92 19
77 76 86 102 129 3 144 101 139 134 56 90 18 91 94 85 55 10 137 11 58 1
107 113
70 22 7 56 29 143 111 8 46 45 116 122 129 89 7 121 53 95 14 49 118 62
125 91 37
97 15 35 100 63 140 63 50 51 58 26 127 6 45 59 102 121 114 85 141 135
10 72 19
106 66 66 41 53 13 38 1 21 103 50 108 46 66 70 43 140 124 40 63 144 97
137 47
60 88 103 115 132 145 116 143 115 10 133 9 118 37 105 51 94 60 67 117
116 ^
305 1 109 110 75 42 12 139 137 43 128 106 107 19 126 12 101 148 127 15
117 125
125 62 96 13 76 70 96 101 110 138 8 95 76 143 17 32 97 79 149 39 31 94
123 21
41 135 55 84 70 33 135 118 50 62 121 81 1 45 144 93 60 5 64 137 8 105
91 82 67
27 113 119 53 18 98 79 48 84 32 135 128 5 1 20 76 17 85 108 72 36 141
140 49
150 105 104 3 149 14 54 18 148 64 49 125 37 28 28 101 22 104 91 32 82
117 12
114 69 58 2 58 115 9 108 47 59 65 14 92 7 4 86 98 16 82 92 95 38 94 10
10 48 97
104 66 115 97 142 115 122 119 40 97 16 32 47 34 88 89 26 50 12 76 80
51 40 9
133 24 44 40 122 84 108 22 142 140 99 44 15 54 8 42 125 150 130 21 79
124 62 46
119 15 29 91 57 150 42 138 71 61 68 80 114 6 1 70 121 18 35 113 56 87
86 10 73
14 29 41 72 89 1 133 87 101 123 59 90 142 77 133 52 78 48 34 138 134
27 17 60
131 147 61 93 148 39 132 49 62 71 36 91 4 139 49 100 120 43 113 144 30
94 73
127 40 34 50 100 110 88 114 94 35 138 33 72 61 58 65 24 107 6 54 81
114 56 137

71 11 23 19 73 1 108 110 25 148 11 98 132 150 17 70 146 50 ^
314 1 99 88 7 139 24 13 35 47 97 145 74 36 119 3 51 84 48 119 53 49 15
79 17
120 103 148 64 30 41 97 120 75 111 63 58 131 134 18 13 10 48 18 16 48
43 15 54
18 41 47 122 144 80 92 145 77 1 33 89 54 46 78 48 21 54 43 40 53 24 16
73 42 94
29 44 34 151 152 23 123 12 142 140 43 37 88 29 19 35 72 96 151 130 62
112 34 36
91 120 50 112 138 2 105 60 68 137 131 5 17 19 139 74 11 120 78 149 58
128 15
104 16 126 78 20 57 134 71 49 90 76 108 126 100 54 68 39 132 153 42
147 146 124
62 87 35 75 61 65 46 100 82 105 113 31 63 5 95 54 71 77 127 150 80 36
144 2 130
59 74 39 3 152 121 122 18 117 12 117 141 118 135 62 36 69 5 39 53 150
52 153
143 30 66 96 126 131 56 137 8 7 86 142 14 7 111 141 93 136 137 134 43
12 89 23
44 9 152 146 121 97 19 38 110 91 67 14 32 110 66 68 8 130 84 73 118 59
24 41 72
121 150 55 37 138 27 104 66 124 9 51 109 47 125 109 148 8 29 47 72 146
149 61
93 10 20 54 15 76 133 125 106 110 94 32 9 29 148 36 66 121 125 56 112
9 1 111
116 94 57 8 108 50 62 140 71 74 89 1 83 152 20 85 17 145 102 39 56 115
47 150
108 64 98 ^
335 1 28 103 19 92 95 152 152 10 11 13 155 67 11 83 101 69 153 152 45
141 14
120 129 140 119 59 2 89 73 70 83 29 16 67 81 29 1 54 65 96 117 2 37 47
128 33 3
89 108 98 139 49 78 27 103 39 119 94 132 90 38 132 55 65 131 90 58 2
54 100 69
118 22 44 19 7 148 93 25 29 123 81 64 131 55 30 1 89 38 97 82 64 9 28
86 123
151 10 133 40 154 102 4 111 65 9 63 59 124 116 72 105 76 57 137 97 32
145 108
78 112 50 43 34 75 20 22 129 68 11 118 74 125 118 57 17 20 129 53 65
61 144 1
17 142 156 52 100 54 15 20 59 52 63 131 20 57 124 31 125 46 106 76 92
8 98 154
152 80 114 15 140 136 112 100 17 92 25 151 150 80 99 69 83 49 43 156
102 19 57
122 96 30 3 39 134 40 32 75 5 76 127 138 99 17 57 52 150 130 18 127 33
23 116
107 78 77 77 42 69 68 48 41 69 33 75 40 49 128 103 4 146 93 10 83 66
96 152 30
38 12 33 5 39 47 41 34 60 74 20 42 156 67 46 56 102 89 3 124 81 99 104
56 50 8
61 74 55 15 87 108 28 138 47 93 60 2 124 46 126 103 91 145 36 25 116
122 79 79

7 111 13 85 121 9 98 32 125 81 134 77 112 15 70 33 110 13 50 148 8 133
127 153
45 19 42 121 94 45 91 95 117 22 9 46 6 26 31 53 150 8 148 128 53 141 ^
326 0 80 123 24 147 1 112 82 159 63 74 97 109 33 151 32 89 87 132 117
46 129 59
115 91 114 118 37 21 9 94 60 25 89 47 79 110 55 12 143 99 87 43 88 56
57 160 76
12 71 128 77 146 117 95 105 42 66 3 76 20 76 101 100 118 149 45 26 143
148 32
57 39 129 19 31 84 123 1 152 135 5 54 30 13 125 68 30 62 101 51 142 5
94 83 20
116 24 107 109 105 91 42 17 27 93 69 3 139 68 79 38 84 2 85 128 126
122 131 46
17 35 98 42 26 111 100 29 120 55 84 114 109 145 14 18 138 14 9 85 7 18
129 91 2
94 51 133 82 87 123 64 39 8 103 38 75 110 78 7 9 45 115 42 138 135 86
78 16 62
52 75 159 54 151 121 149 77 74 16 85 47 102 105 82 119 10 67 137 153
148 135 28
49 26 151 153 36 80 11 130 113 24 44 30 102 24 58 133 122 140 99 24
156 54 119
42 115 140 90 132 19 94 2 157 99 136 19 71 7 130 153 108 51 21 58 70
74 137 1
40 111 149 5 103 6 27 76 141 23 125 140 1 72 29 152 103 87 51 93 29 80
132 77
123 153 68 159 14 98 114 158 121 158 81 137 41 93 118 140 122 19 22 31
147 41 4
89 150 50 120 144 113 104 20 44 53 97 20 4 90 60 48 84 74 25 138 23 62
61 159 5
14 107 6 4 81 114 46 97 11 112 154 130 73 132 60 ^
325 0 58 137 96 11 98 67 98 103 57 146 21 59 88 151 139 148 127 25 17
47 115 34
160 109 107 51 64 28 69 13 49 149 69 141 90 93 118 64 10 1 67 80 35
111 13 58
101 124 132 147 154 18 162 6 162 33 5 34 142 41 161 82 114 70 92 145
57 155 137
114 79 44 36 48 48 21 14 13 40 33 14 150 33 32 54 143 14 4 101 142 23
93 136
132 120 147 17 38 163 143 5 52 46 151 130 32 72 34 124 150 51 100 112
128 126
65 10 28 87 81 159 131 19 99 54 125 110 58 119 28 78 129 104 140 126
38 154 27
114 61 153 90 66 98 76 50 158 48 39 82 123 22 147 136 114 52 37 35 75
41 15 150
60 52 55 103 21 23 129 95 24 71 47 97 130 50 140 144 106 100 9 64 19
117 122 71
92 8 77 156 97 121 98 85 2 36 39 109 143 23 120 156 133 93 154 36 66
116 131
160 127 162 161 46 142 14 141 81 141 63 86 117 104 3 146 39 127 34 133
102 106
91 57 9 28 60 61 7 158 12 80 26 8 122 80 44 63 68 49 158 21 32 81 150
15 141

108 161 64 46 124 123 31 99 27 105 109 98 112 9 37 12 116 149 51 73
144 124 4
119 46 133 75 86 90 84 2 9 19 108 6 16 101 115 16 92 163 125 101 106
64 37 8 68
40 42 130 41 24 89 115 43 112 85 121 125 92 154 9 46 115 7 140 108 158
^
336 0 38 42 155 103 9 62 65 102 122 10 138 160 125 47 158 43 91 69 123
132 35
121 4 110 89 130 69 29 139 69 53 70 83 29 163 67 41 9 108 34 45 76 87
2 144 164
98 33 160 79 78 48 89 9 38 134 93 146 79 54 122 80 38 112 55 55 101 70
8 129 44
70 59 98 149 24 136 124 138 63 25 166 83 51 34 91 45 30 118 59 28 87
72 44 116
28 36 103 101 113 10 114 62 111 71 65 126 53 19 114 86 42 85 36 57 137
57 159
95 88 78 72 20 23 14 65 10 22 129 68 1 68 74 75 108 7 7 147 109 13 35
51 104
158 164 122 126 2 50 4 132 127 59 52 13 81 20 47 107 74 148 115 46 106
46 82
115 68 144 142 60 104 15 90 136 102 100 134 42 15 141 100 40 49 49 73
166 13
156 82 166 37 82 96 137 130 166 134 20 139 45 122 56 107 98 79 124 17
32 130
120 165 77 23 130 96 67 68 47 37 12 29 18 38 158 19 160 55 147 39 118
83 121 96
43 137 33 66 86 112 147 155 149 140 5 19 17 148 161 10 44 159 146 57
16 26 102
49 3 104 61 59 74 56 10 165 31 54 25 142 157 37 58 165 128 154 73 50
149 94 36
76 63 71 135 26 5 116 122 29 69 7 101 140 75 81 136 78 2 125 71 84 57
62 162 40
3 80 130 50 98 125 93 127 153 45 146 149 121 74 5 41 55 77 139 166 153
113 153
21 53 140 145 148 88 162 ^
341 0 4 62 78 167 102 144 94 55 141 63 29 97 91 24 115 166 80 69 132
99 1 120
23 88 64 87 118 37 137 152 94 60 168 71 47 52 110 37 155 125 63 42 43
52 11 12
151 31 12 44 110 32 128 117 68 87 24 39 164 76 145 58 101 91 100 140
151 143
130 32 21 3 111 1 31 75 123 153 116 135 130 27 164 165 116 23 12 62 83
24 133
139 49 74 154 80 158 80 64 105 91 6 142 27 75 24 128 112 41 79 29 84
145 40 128
99 95 95 19 17 160 89 15 17 84 64 11 93 10 66 78 73 127 148 18 129 139
143 49
150 9 84 82 154 85 15 88 82 60 87 19 12 133 58 20 39 65 51 141 134 27
70 167
120 117 86 60 16 44 16 57 132 18 142 85 104 59 47 141 58 2 66 96 46
119 153 40

110 126 103 90 144 13 26 106 144 80 145 134 103 95 24 44 21 84 140 13
97 104
140 99 6 147 54 83 42 106 131 54 96 135 67 118 121 81 109 10 53 132
112 117 81
33 155 49 61 38 119 1 13 102 131 148 94 131 143 67 123 148 89 104 135
72 145
152 76 87 6 66 2 71 123 77 114 108 59 123 166 62 96 140 94 149 116 36
128 23 93
91 95 113 162 156 165 111 166 4 44 105 5 120 99 113 68 11 169 35 70 2
147 125
81 15 12 57 56 16 138 14 53 61 114 121 5 107 6 129 81 114 37 61 127 67
136 94
73 114 32 34 119 72 11 98 37 74 73 51 146 161 35 146 ^
349 0 142 139 121 91 16 163 2 88 171 133 100 62 51 46 10 24 150 49 131
60 114
63 84 91 64 165 138 40 44 172 111 141 58 74 115 96 129 145 164 153 170
126 24
169 16 115 41 125 46 87 61 92 145 39 155 92 78 70 35 27 21 48 21 151
159 40 15
5 132 170 23 18 107 160 150 56 133 23 66 109 123 102 102 172 166 145
116 151 34
1 151 130 5 36 34 97 114 15 82 128 112 119 99 29 138 165 42 36 159 95
19 63 36
89 101 40 92 1 33 93 104 113 126 2 136 96 52 108 90 57 89 31 5 113 30
39 37 96
4 147 127 105 43 165 35 75 23 143 105 24 25 10 94 12 160 102 95 170 71
20 70
112 23 95 144 61 73 137 55 1 81 95 26 65 172 41 147 79 103 80 40 121
36 12 64
98 169 93 111 115 48 127 9 39 107 131 115 118 162 161 10 142 14 123 54
141 36
41 99 77 140 128 167 82 25 106 57 70 64 21 19 15 34 126 149 167 53 163
127 86
35 8 54 23 40 140 3 169 45 150 152 96 81 143 28 28 124 87 13 90 9 87
109 53 67
164 28 131 89 149 42 55 126 79 132 74 19 133 30 68 72 75 148 9 10 72
152 144 83
106 153 74 163 98 92 97 37 19 8 32 31 24 121 14 152 89 79 7 76 155 85
76 107 83
154 155 37 115 144 131 108 116 26 36 137 103 3 44 47 72 104 10 120 154
107 35
152 19 85 69 105 120 29 109 171 104 65 124 39 11 127 57 41 70 83 29
157 67 164 ^
370 1 167 63 16 27 58 60 2 99 128 71 33 160 70 51 3 44 149 2 89 84 101
43 18
113 71 38 94 55 46 74 52 139 102 35 43 50 80 122 6 100 88 129 36 25
148 47 24 7
55 36 30 82 32 19 78 63 26 71 28 167 85 56 167 95 159 78 26 66 35 65
90 44 159
105 59 15 67 57 137 21 132 50 70 78 36 169 5 172 56 1 22 129 68 168 23
74 30 99

138 174 120 91 153 8 42 68 158 155 104 99 133 5 135 96 82 59 52 144 36
20 38 62
29 112 106 46 106 19 73 70 41 135 133 42 95 15 45 136 93 100 98 173 6
132 55 4
4 31 64 130 162 156 64 157 19 46 96 92 103 139 134 2 94 18 86 38 89 62
61 79
157 14 112 111 156 32 14 85 78 31 59 20 1 161 169 149 29 122 150 133
37 102 30
109 65 85 51 174 110 164 66 77 76 111 119 131 95 5 1 166 103 134 141
17 158 123
137 48 165 175 102 13 3 86 43 23 47 56 150 165 4 36 174 115 157 168 13
147 119
109 55 41 140 67 27 31 27 53 126 17 163 116 122 160 60 7 92 113 66 45
109 60
151 125 62 39 39 17 153 13 152 53 94 50 53 89 57 127 153 45 119 104
121 56 145
172 19 41 103 166 108 68 126 12 53 131 127 148 52 134 168 50 48 155 72
132 82
37 129 63 175 97 79 18 91 148 74 57 132 87 147 114 175 70 46 69 118 37
107 140
94 60 156 59 47 34 110 25 143 113 39 12 43 28 157 158 145 1 12 26 98 2
116 117
50 75 12 21 164 76 121 46 101 85 88 134 146 127 143 79 ^
364 0 32 164 146 93 162 31 66 123 144 80 135 94 137 156 107 157 173 62
65 176
124 112 4 65 127 44 131 53 19 105 91 149 106 27 57 158 92 85 14 79 20
84 127
174 128 72 68 59 171 17 124 80 167 8 57 28 172 66 144 48 42 37 109 121
18 120
103 116 13 132 39 73 145 76 158 43 82 33 51 153 164 97 13 2 3 20 24
114 98 9 25
131 102 99 86 42 16 26 159 39 105 161 133 49 59 41 20 105 31 136 30 87
10 119
135 13 83 99 58 45 99 156 26 61 135 143 80 118 107 76 77 24 44 12 66
95 147 61
86 140 99 167 138 54 47 42 97 122 18 60 90 40 73 85 63 82 1 35 96 94
81 54 15
128 40 52 2 101 1 165 93 113 130 85 95 98 58 105 112 53 68 108 72 100
152 49 87
140 39 154 62 114 77 105 63 50 87 157 26 78 122 67 140 71 170 119 5 93
64 50
104 144 129 138 75 130 4 178 60 139 120 54 113 32 2 133 17 43 163 129
89 72 149
155 30 38 7 138 5 44 61 69 76 175 107 6 93 81 114 28 25 82 22 118 58
73 96 175
177 92 36 11 98 171 38 28 42 146 125 178 88 136 139 103 67 10 151 151
70 153
115 94 32 51 34 177 173 132 49 119 54 96 45 78 73 64 159 120 22 20 154
111 117
58 56 109 72 117 139 152 147 170 102 18 169 4 97 41 101 22 69 55 92
145 27 155

62 54 64 29 21 3 48 21 133 147 40 3 178 120 152 17 173 83 148 138 26
127 23 48
91 117 90 72 166 142 133 98 139 22 150 151 130 140 ^
372 0 34 70 78 161 64 92 112 110 72 175 102 138 179 173 159 59 19 27
18 53 92
22 65 156 170 57 104 86 126 148 118 155 78 43 63 90 48 80 168 142 68
12 39 174
69 168 147 118 96 34 129 35 75 5 107 60 170 180 147 85 3 133 75 95 152
71 175
43 94 178 50 144 16 46 101 46 165 45 68 163 38 172 5 138 61 85 62 177
76 36 167
19 53 151 66 66 97 3 100 164 12 98 131 70 109 162 161 156 142 14 105
27 141 9
178 81 50 113 110 131 37 16 79 12 34 37 167 173 10 152 7 81 140 158 26
136 82
50 172 154 45 160 31 122 167 142 9 150 125 51 54 125 174 10 124 51 177
81 173
69 109 8 22 155 19 86 62 149 33 37 108 34 96 29 174 133 167 50 54 66
130 9 1 36
134 108 65 97 126 56 163 71 83 88 10 1 8 178 22 6 112 169 116 89 43
153 40 146
85 31 89 74 154 137 28 115 117 122 108 98 8 27 110 103 176 17 20 27 77
10 93
145 80 17 143 165 76 69 78 102 20 91 171 95 29 115 176 166 109 39 23
70 83 29
148 67 163 161 33 4 15 46 42 2 69 104 53 33 160 64 33 155 14 131 160
59 78 71
19 176 107 65 38 82 55 40 56 40 115 84 29 25 44 68 104 176 76 64 123
18 25 136
23 6 171 31 30 30 58 14 13 72 57 14 41 28 143 73 26 167 83 147 54 2 36
11 65 66
38 141 99 41 179 55 158 57 137 179 114 20 58 78 12 157 175 166 50 177
22 129 68
168 175 74 93 114 174 102 79 135 172 36 44 158 149 92 81 109 157 111
72 119 ^
372 0 59 52 112 180 20 30 22 173 80 98 46 106 179 65 30 17 127 125 26
87 15 5
136 85 100 66 141 182 124 15 156 148 15 56 98 146 156 48 149 3 14 96
52 79 115
134 170 54 178 54 22 73 30 45 39 133 182 96 103 148 176 6 45 62 183 51
180 153
145 145 117 21 90 118 109 21 62 22 101 49 53 11 142 86 132 66 69 44 79
87 115
55 5 169 150 63 110 109 177 150 91 129 40 149 159 102 165 3 70 27 175
23 56 126
165 164 20 158 91 157 136 157 131 111 69 39 33 132 43 19 175 179 37
118 9 155
116 122 128 52 7 84 89 58 13 85 44 135 125 54 183 23 161 145 173 136
29 62 50
13 57 25 127 153 45 95 64 121 40 121 140 171 9 71 166 68 28 102 4 53
123 111

148 20 102 160 34 8 139 32 116 66 13 113 63 143 97 63 10 59 124 66 41
132 71
115 106 151 46 22 45 118 37 67 124 94 60 140 43 47 10 110 9 127 97 7
156 43 180
125 126 137 145 12 2 82 146 100 117 26 59 180 181 164 76 89 30 101 77
72 126
114 95 143 102 32 149 131 83 157 31 61 123 139 60 135 74 169 122 151
102 137
168 62 55 166 119 97 163 60 112 24 116 38 178 105 91 134 86 27 47 138
72 70 183
79 15 84 117 154 128 57 53 39 161 17 104 75 157 3 42 8 167 51 124 38
22 17 99
106 18 115 83 101 177 122 179 14 68 140 71 143 18 82 18 31 133 154 77
172 176
167 179 9 99 78 183 111 92 89 86 32 16 16 144 29 90 146 128 29 34 31 5
85 16
116 10 82 174 119 125 79 ^
391 0 59 75 18 5 59 132 26 21 127 119 80 94 83 52 61 24 44 4 50 55 115
29 70
140 99 159 130 54 15 42 89 114 173 28 50 16 33 53 47 58 180 19 64 78
49 30 186
104 32 44 157 85 1 149 85 97 114 77 63 58 50 89 80 21 36 84 72 60 152
25 87 108
15 138 54 106 77 97 23 42 55 149 181 62 106 43 132 31 138 111 176 93
40 10 96
128 105 114 43 98 4 146 20 107 120 14 113 181 101 1 19 155 113 57 64
117 131 6
22 186 138 184 36 61 29 36 175 107 6 61 81 114 20 180 42 169 102 26 73
80 151
153 68 4 11 98 139 6 175 34 146 93 154 88 128 139 79 35 2 135 119 46
129 91 86
179 51 18 169 141 108 49 103 46 72 21 70 49 64 151 96 185 175 130 111
85 58 32
101 40 101 131 136 139 170 70 10 169 175 73 41 69 177 45 47 92 145 11
155 22 22
56 21 13 166 48 21 109 131 40 174 178 104 128 9 149 51 132 122 173 119
23 24 67
109 74 32 158 110 117 74 123 6 118 151 130 150 167 34 55 58 146 54 72
112 105
57 160 82 123 159 153 159 39 19 7 8 33 87 12 50 146 150 37 104 71 126
133 108
145 68 38 38 90 43 75 148 122 43 2 39 154 54 163 147 113 91 29 109 35
75 182 87
35 155 170 127 80 185 118 60 95 142 71 165 28 84 168 25 144 178 31 81
41 160 25
53 143 23 172 172 133 51 75 52 157 51 36 157 181 28 141 51 41 87 165
85 154 184
93 131 45 104 162 161 141 142 14 95 12 141 181 158 71 35 98 100 111 12
11 64
174 14 22 152 173 5 132 179 56 135 153 11 121 57 30 152 139 40 140 53
^

384 1 106 159 118 167 150 101 11 30 109 150 184 124 19 169 73 165 53
109 158
172 147 11 46 38 149 25 21 92 184 64 179 158 133 135 34 38 58 114 9
183 4 118
76 49 89 102 40 163 47 75 80 176 175 8 154 14 180 104 153 84 89 11 129
8 138 85
181 73 66 154 121 20 115 93 114 108 82 182 19 86 103 176 183 186 177
53 10 69
137 56 1 135 141 68 69 54 86 12 75 171 87 187 107 144 150 93 23 7 70
83 29 140
67 139 153 183 178 189 30 18 2 29 72 29 33 160 56 9 123 164 107 136 19
70 31
177 152 99 57 38 66 55 32 32 24 83 60 21 1 36 52 80 168 44 32 115 184
25 120
181 172 155 189 22 30 26 180 5 64 49 188 1 28 111 57 176 167 67 131 22
160 186
169 65 34 30 117 91 17 163 39 134 57 137 155 90 170 42 78 170 141 167
158 42
177 22 129 68 168 143 74 150 85 82 174 78 63 111 156 28 12 158 141 76
57 77 125
79 40 12 59 52 88 156 20 24 182 149 56 92 46 106 167 59 189 121 119 14
81 15
165 136 79 100 42 117 182 118 175 138 124 3 50 74 134 156 36 143 181
180 96 22
61 97 134 164 24 166 30 10 61 6 33 9 115 176 84 97 142 152 15 50 165
45 168 135
133 127 93 15 66 94 91 9 32 16 95 37 29 171 118 68 108 66 63 20 55 63
103 25 5
163 138 33 92 85 165 144 67 123 34 137 147 102 147 3 58 15 157 5 56
108 165 152
8 146 73 157 112 133 119 105 39 27 27 126 25 13 151 161 25 112 3 149
116 122
104 46 7 78 71 52 179 67 32 123 125 48 159 11 137 139 161 110 ^
394 0 2 26 50 161 21 182 127 153 45 68 19 121 22 94 104 144 166 35 166
23 176
75 188 53 114 93 148 177 66 151 16 156 121 180 98 48 179 95 63 107 97
45 1 23
97 57 23 132 53 79 97 124 19 188 18 118 37 22 106 94 60 122 25 47 176
110 184
109 79 164 120 43 153 89 90 128 109 12 168 64 110 82 117 192 41 171
163 164 76
53 12 101 68 54 117 78 59 143 84 32 122 104 65 148 31 52 123 130 24
135 38 151
95 142 93 101 159 62 37 148 110 70 127 51 85 181 89 11 142 105 91 107
50 27 29
102 36 43 165 79 6 84 99 118 128 30 26 3 143 17 68 66 139 187 15 165
158 24 88
20 179 174 81 79 18 106 47 74 150 104 179 162 59 131 62 116 166 82 184
188 97
136 41 136 167 140 143 175 72 42 174 148 75 74 71 86 14 16 191 117 11
63 119

119 186 182 13 171 49 182 80 167 73 147 119 107 164 41 57 181 168 29
114 26 184
121 101 80 76 65 34 49 24 44 191 38 25 91 5 58 140 99 153 124 54 184
42 83 108
155 4 20 191 3 29 35 40 180 7 40 66 25 12 180 86 26 38 139 73 1 137 79
85 102
71 39 28 44 77 56 190 12 66 72 30 152 7 87 84 190 126 48 100 77 91 186
36 31
143 163 50 94 25 126 1 114 105 170 93 22 173 90 116 87 96 19 74 4 122
183 83
120 177 113 169 181 77 182 1 149 101 33 58 93 113 181 10 186 138 184
30 61 192
6 175 107 6 37 81 114 14 162 12 145 90 2 73 68 133 135 50 173 11 98
115 175 151
28 146 69 136 88 122 139 61 11 189 123 95 28 111 73 80 155 51 6 163
117 90 49
91 40 54 189 ^
390 0 62 25 64 143 72 169 151 106 111 53 58 8 93 8 85 123 120 131 170
38 2 169
167 49 41 37 153 21 39 92 145 190 155 177 185 48 13 5 150 48 21 85 115
40 166
178 88 104 1 125 19 116 106 141 111 23 43 101 58 187 150 78 101 50 107
185 86
151 130 134 143 34 31 26 122 38 40 112 97 33 136 50 99 127 121 159 7
19 170 187
1 79 191 26 130 118 5 104 47 126 109 92 129 52 30 193 90 35 67 116 90
3 181 39
122 30 155 147 105 83 21 77 35 75 174 55 190 131 154 95 72 185 94 36
95 126 71
149 4 68 152 180 144 146 7 49 33 152 188 29 111 194 172 148 125 35 59
36 125 11
36 141 149 183 125 27 1 71 133 61 138 168 85 131 5 96 162 161 117 142
14 79 183
141 165 126 55 11 74 84 79 167 3 40 142 177 193 128 173 192 100 163 16
127 145
182 97 17 193 120 115 32 108 18 96 154 103 152 150 86 181 15 99 135
179 124 194
164 68 160 43 109 138 152 142 6 21 23 149 20 11 82 164 44 159 148 133
115 24 28
53 104 9 183 179 108 56 39 84 87 30 163 32 70 75 166 170 8 139 9 175
99 143 64
89 186 114 183 133 85 161 63 61 154 111 15 115 78 109 108 72 177 14 71
103 176
173 176 157 38 10 54 132 41 186 130 126 63 69 39 76 7 65 171 82 172
102 124 140
83 13 192 70 83 29 135 67 124 148 163 173 184 20 3 2 4 52 14 33 160 51
189 103
144 92 121 189 65 6 162 137 94 52 38 56 55 27 17 14 63 45 16 181 31 42
65 163
24 12 110 174 25 110 166 162 145 174 17 30 6 170 59 44 183 171 28 91
47 156 172
^

400 1 51 115 188 136 154 145 65 2 22 93 83 191 147 23 110 57 137 131
66 138 26
78 146 125 159 150 34 177 22 129 68 168 111 74 118 77 50 174 54 47 87
140 20
178 158 133 60 33 45 93 47 8 170 59 52 56 124 20 16 150 117 24 84 46
106 151 51
158 173 113 111 196 73 15 133 136 71 100 10 85 182 110 143 114 92 185
42 42 118
156 20 135 173 156 96 180 37 73 134 156 182 150 196 192 45 172 17 167
91 168 68
89 134 120 190 173 34 141 37 152 111 117 103 61 7 34 62 67 191 190 8
87 21 195
139 86 44 76 66 55 186 23 31 87 183 5 155 122 191 68 53 149 136 35 115
26 121
131 102 123 3 42 197 133 179 56 84 165 136 190 130 49 157 80 101 103
97 197 11
19 118 1 5 119 137 9 104 193 141 116 122 72 38 7 70 47 44 155 43 16
107 125 40
127 193 105 131 145 108 185 6 50 141 1 167 127 153 45 53 192 121 12 79
84 129
151 15 166 196 156 60 188 53 109 83 148 162 46 146 6 136 111 160 88 38
169 85
63 87 97 35 194 3 82 52 13 132 43 59 92 109 4 178 3 118 37 195 96 94
60 112 15
47 166 110 179 99 69 149 100 43 138 69 70 123 89 12 158 54 90 72 117
182 31 166
153 164 76 33 2 101 63 44 112 58 39 143 74 32 107 89 55 143 31 47 123
125 4 135
18 141 80 137 88 81 154 62 27 138 105 55 107 46 70 166 74 194 122 105
91 92 30
27 19 82 16 28 155 79 1 84 89 98 128 15 11 181 133 17 48 61 129 187
150 153 9
68 10 164 159 71 64 18 101 27 59 135 94 179 142 54 126 57 101 146 82
174 173 77
126 21 116 162 125 123 165 57 22 169 153 ^
394 0 47 60 57 86 16 184 96 197 42 98 112 165 154 199 157 21 168 52
146 66 126
119 93 150 20 36 153 140 194 93 26 156 114 80 80 55 44 13 35 24 44 191
24 190
63 177 44 140 99 146 117 54 163 42 76 101 134 176 185 177 168 1 21 19
180 193
12 52 197 191 173 65 19 31 118 59 1 123 72 71 88 64 11 193 37 63 28
169 184 45
72 195 152 186 87 56 176 112 41 93 77 84 158 29 3 136 142 36 80 4 119
166 86 98
163 93 1 145 83 102 66 75 191 46 4 94 155 55 120 149 113 148 181 49
175 180 142
87 5 51 65 92 167 196 186 138 184 23 61 164 171 175 107 6 9 81 114 7
141 177
117 76 174 73 54 112 114 29 152 11 98 87 154 123 21 146 41 115 88 115
139 40

183 189 109 67 7 90 52 73 127 51 192 156 89 69 49 77 33 33 182 57 10
64 138 57
159 136 91 111 33 58 193 88 188 75 118 110 126 170 18 197 169 162 34
41 17 138
6 34 92 145 185 155 157 170 43 8 140 48 21 70 105 40 161 178 78 89 196
110 199
106 96 121 106 23 185 28 96 48 167 145 58 91 35 97 180 66 151 130 124
128 34 16
6 107 28 20 112 92 18 121 30 84 107 101 159 187 19 155 182 181 74 186
11 120 98
185 104 32 126 94 82 119 42 25 173 90 30 62 96 70 178 176 39 102 15
150 147 100
78 16 57 35 75 169 35 170 116 144 75 67 185 79 21 95 116 71 139 189 58
142 160
144 126 192 29 28 147 173 14 91 184 172 133 120 25 49 26 105 186 36
131 129 163
115 12 176 61 113 46 128 158 80 131 180 91 162 161 102 142 14 69 173
141 155
106 45 196 59 74 59 147 195 ^
412 0 16 110 153 177 104 173 192 68 147 179 119 137 166 73 180 169 88
91 24 76
10 80 146 79 128 150 62 149 194 83 111 171 124 170 156 60 152 27 109
106 120
134 201 184 202 149 12 198 66 132 12 127 132 133 83 8 12 45 88 9 183
155 92 24
23 76 63 14 163 8 62 67 150 162 8 115 1 167 91 127 32 89 162 90 159
125 85 129
47 53 154 95 7 115 54 101 108 56 169 6 47 103 176 157 160 125 14 10 30
124 17
178 122 102 55 69 15 60 202 49 171 74 148 94 92 124 67 200 184 70 83
29 127 67
100 140 131 165 176 4 182 2 167 20 193 33 160 43 173 71 112 68 97 157
57 169
138 113 86 44 38 40 55 19 196 201 31 21 8 165 23 26 41 155 195 183 102
158 25
94 142 146 129 150 9 30 177 154 195 51 36 175 139 28 59 31 124 167 41
105 173
121 134 130 65 185 17 78 78 181 137 13 95 57 137 116 51 118 16 78 131
115 154
145 29 177 22 129 68 168 91 74 98 72 30 174 39 37 72 130 15 163 158
128 50 18
25 73 27 191 150 59 52 36 104 20 11 130 97 4 79 46 106 141 46 138 163
108 106
191 68 15 113 136 66 100 193 65 182 105 123 99 72 180 37 22 108 156 10
130 168
141 96 160 22 58 134 151 162 140 181 187 35 157 7 147 76 163 58 84 129
100 190
153 24 126 32 142 96 107 88 41 2 14 42 52 186 170 3 82 11 180 119 66
29 56 66
50 171 3 11 77 163 5 150 112 171 53 33 139 131 15 110 21 111 121 102
108 3 32

192 118 169 56 69 165 126 185 120 34 157 60 81 93 92 177 1 14 113 189
99 122
202 99 193 136 116 122 52 33 7 65 32 39 140 28 6 97 125 35 107 188 85
126 135
98 175 189 50 121 184 116 ^
413 1 127 153 45 32 164 121 203 58 56 108 130 192 166 168 128 39 188
53 102 69
148 141 18 139 197 108 97 132 74 24 155 71 63 59 97 21 194 180 61 45
204 132 29
31 85 88 188 164 187 118 37 167 82 94 60 98 1 47 152 110 172 85 55 128
72 43
117 41 42 116 61 12 144 40 62 58 117 168 17 159 139 164 76 5 193 101
56 30 105
30 11 143 60 32 86 68 41 136 31 40 123 118 181 135 195 127 59 130 81
53 147 62
13 124 98 34 79 39 49 145 53 180 94 105 91 71 2 27 5 54 193 7 141 79
199 84 75
70 128 199 195 160 119 17 20 54 115 187 184 129 146 193 40 201 143 138
57 43 18
94 204 38 114 80 179 114 47 119 50 80 118 82 160 152 49 112 198 88 155
104 95
151 36 199 162 100 27 50 47 86 195 16 179 81 192 27 83 107 150 134 194
147 1
158 32 131 61 111 119 83 140 5 21 133 120 174 78 26 136 109 65 80 40
29 203 25
24 44 191 14 170 43 162 34 140 99 141 112 54 148 42 71 96 119 161 165
167 148
186 11 4 180 188 197 42 182 181 168 50 14 26 103 49 1 113 67 61 78 59
196 173
32 53 8 154 169 30 72 175 152 176 87 36 166 102 36 88 77 79 138 24 188
131 127
26 70 194 114 146 66 93 158 93 191 125 78 92 51 60 176 26 4 74 135 35
120 129
113 133 181 29 170 170 137 77 190 46 45 77 157 191 186 138 184 18 61
144 151
175 107 6 194 81 114 2 126 157 97 66 159 73 44 97 99 14 137 11 98 67
139 103 16
146 21 100 88 110 139 25 168 189 99 47 197 75 37 68 107 51 187 151 69
54 49 67
28 18 172 52 200 64 133 42 149 121 76 111 13 58 183 83 173 65 113 100
121 170
203 197 169 157 19 202 ^
434 1 193 114 190 26 92 145 177 155 125 146 35 200 124 48 21 46 89 40
153 178
62 65 196 86 175 90 80 89 98 23 169 4 88 32 135 137 26 75 11 81 172 34
151 130
108 104 34 200 182 83 12 196 112 84 202 97 206 60 75 69 159 163 19 131
174 157
66 178 195 104 66 161 104 8 126 70 66 103 26 17 141 90 22 54 64 38 146
168 39
70 199 142 147 92 70 8 25 35 75 161 3 138 92 128 43 59 185 55 205 95
100 71 123

173 42 126 128 144 94 176 205 20 139 149 198 59 168 172 109 112 9 33
10 73 154
36 115 97 131 99 196 144 45 81 22 112 142 72 131 148 83 162 161 78 142
14 53
157 141 139 74 29 180 35 58 27 115 198 1 90 138 167 89 173 192 48 137
159 114
132 156 58 160 154 68 76 19 56 5 70 141 64 113 150 47 129 184 73 96
166 124 155
151 55 147 17 109 86 100 129 201 164 192 149 7 193 56 112 200 107 122
133 63
206 2 40 78 9 183 140 82 4 13 71 48 4 163 201 57 62 140 157 8 100 204
162 86
117 12 89 147 75 144 120 85 109 37 48 154 85 2 115 39 96 108 46 164 1
32 103
176 147 150 105 207 10 15 119 2 173 117 87 50 69 50 202 39 171 69 133
89 72 114
57 195 179 70 83 29 122 67 85 135 111 160 171 202 172 2 147 183 33 160
38 163
51 92 53 82 137 52 149 123 98 81 39 38 30 55 14 186 196 11 6 3 155 18
16 26 150
180 168 97 148 25 84 127 136 119 135 4 30 162 144 195 46 31 170 119 28
39 21
104 167 31 95 158 106 114 115 65 170 12 63 73 171 127 3 80 57 137 101
36 98 6
78 116 105 149 140 24 177 22 129 68 168 71 74 78 67 10 174 24 27 57
120 10 148
158 123 40 3 5 53 7 176 130 59 52 16 84 20 6 110 77 192 74 46 106 131
41 118
153 144 ^
438 1 99 184 61 15 85 136 59 100 172 37 182 98 95 78 44 173 30 204 94
156 206
123 161 120 96 132 1 37 134 144 134 126 160 180 21 136 203 119 55 156
44 77 122
72 190 125 10 105 25 128 75 93 67 13 205 196 14 31 179 142 206 75 207
159 91 38
8 28 66 43 150 185 193 63 135 5 143 98 143 32 5 125 124 197 103 14 97
107 102
87 3 18 185 97 155 56 48 165 112 178 106 13 157 32 53 79 85 149 197 7
106 175
203 71 101 195 92 193 129 116 122 24 26 7 58 11 32 119 7 202 83 125 28
79 181
57 119 121 84 161 168 50 93 163 131 127 153 45 17 144 121 198 43 36 93
115 177
166 148 108 24 188 53 97 59 148 126 208 134 192 88 87 112 64 14 145 61
63 39 97
11 194 165 46 40 199 132 19 11 80 73 178 154 177 118 37 147 72 94 60
88 201 47
142 110 167 75 45 113 52 43 102 21 22 111 41 12 134 30 42 48 117 158 7
154 129
164 76 195 188 101 51 20 100 10 201 143 50 32 71 53 31 131 31 35 123
113 166

135 180 117 44 125 76 33 142 62 3 114 93 19 59 34 34 130 38 170 74 105
91 56
192 27 205 34 178 202 131 79 199 84 65 50 128 189 185 145 109 17 49
105 187 174
114 141 183 20 196 128 123 47 28 18 89 189 23 99 70 179 94 42 114 45
65 98 82
150 137 29 102 183 68 150 89 75 141 21 184 157 80 7 40 37 86 190 16
174 66 187
12 68 102 135 114 189 137 191 148 12 116 56 96 119 73 130 200 6 113
100 154 63
26 116 104 50 80 25 14 193 15 24 44 191 4 150 23 147 24 140 99 136 107
54 133
42 66 91 104 146 145 157 128 171 1 199 180 183 182 32 167 171 163 35 9
21 88 39
1 103 62 51 68 54 181 153 27 43 198 139 154 15 72 155 152 166 87 16
156 92 31
83 77 74 118 19 173 126 202 ^
431 0 10 54 178 106 114 34 85 150 93 175 93 70 76 27 36 152 207 4 42
103 3 120
97 113 109 181 210 162 154 129 61 166 38 13 53 141 183 186 138 184 10
61 112
119 175 107 6 170 81 114 207 102 125 65 50 135 73 28 73 75 203 113 11
98 35 115
71 8 146 202 76 88 102 139 1 144 189 83 15 181 51 13 60 75 51 179 143
37 30 49
51 20 207 156 44 184 64 125 18 133 97 52 111 194 58 167 75 149 49 105
84 113
170 179 197 169 149 208 41 178 99 180 21 92 145 172 155 105 131 30 208
200 114
48 21 31 79 40 148 178 52 50 196 71 160 80 70 69 93 23 159 202 83 22
115 132 6
65 209 71 167 14 151 130 98 89 34 190 167 68 2 181 112 79 192 82 191
45 55 49
159 148 19 116 169 142 61 173 185 94 46 146 104 206 126 55 56 93 16 12
121 90
17 49 44 18 126 163 39 50 189 137 147 87 65 3 5 35 75 156 196 118 77
118 23 54
185 40 195 95 90 71 113 163 32 116 108 144 74 166 190 15 134 134 188
39 158 172
94 107 212 23 53 134 36 105 77 111 89 186 124 35 61 7 102 132 67 131
128 78 162
161 63 142 14 43 147 141 129 54 19 170 20 48 7 95 198 199 70 123 157
74 173 192
28 127 139 109 127 146 43 140 139 48 61 14 36 60 136 49 98 150 32 109
174 63 81
161 124 140 146 50 142 7 109 66 80 124 201 144 182 149 2 188 46 92 185
87 112
133 43 201 205 35 68 9 183 125 72 197 3 66 33 207 163 191 52 57 130
152 8 85
204 157 81 107 205 89 132 60 129 115 85 89 27 43 154 75 210 115 24 91
108 36

159 209 17 103 176 137 140 85 197 10 114 200 168 112 72 45 69 198 40
202 29 171
64 118 84 52 104 47 190 174 70 83 29 117 67 70 130 91 155 166 197 162
2 127 193
173 82 ^
464 1 160 31 149 23 64 32 61 109 45 121 102 77 74 32 38 16 55 7 172
189 198 200
211 141 11 2 5 143 159 147 90 134 25 70 106 122 105 114 212 30 141 130
195 39
24 163 91 28 11 7 76 167 17 81 137 85 86 94 65 149 5 42 66 157 113 204
59 57
137 80 15 70 207 78 95 91 142 133 17 177 22 129 68 168 43 74 50 60 197
174 3 13
36 106 3 127 158 116 26 197 192 25 194 155 102 59 52 203 56 20 214 82
49 171 67
46 106 117 34 90 139 96 94 179 56 15 65 136 54 100 157 17 182 93 75 63
24 168
25 189 84 156 201 118 156 105 96 112 201 22 134 139 114 116 145 175 11
121 198
99 40 151 34 72 117 52 190 105 90 20 118 60 83 52 208 205 181 209 16
174 122
206 70 202 144 71 18 208 8 66 38 135 170 178 53 115 5 138 88 123 17
200 115 119
182 98 9 87 97 102 72 3 8 180 82 145 56 33 165 102 173 96 213 157 12
33 69 80
129 192 2 101 165 203 51 86 190 87 193 124 116 122 4 21 7 53 211 27
104 207 197
73 125 23 59 176 37 114 111 74 151 153 50 73 148 116 127 153 45 2 124
121 193
28 16 78 100 162 166 128 88 9 188 53 92 49 148 111 193 129 187 68 77
92 54 4
135 51 63 19 97 1 194 150 31 35 194 132 9 206 75 58 168 144 167 118 37
127 62
94 60 78 196 47 132 110 162 65 35 98 32 43 87 1 2 106 21 12 124 20 22
38 117
148 212 149 119 164 76 180 183 101 46 10 95 205 186 143 40 32 56 38 21
126 31
30 123 108 151 135 165 107 29 120 71 13 137 62 208 104 88 4 39 29 19
115 23 160
54 105 91 41 177 27 200 14 163 192 121 79 199 84 55 30 128 179 175 130
99 17
195 44 95 187 164 99 136 173 191 113 108 37 13 18 84 174 8 84 60 179
74 37 109
40 50 78 82 140 122 9 92 168 48 145 74 55 131 6 169 152 60 202 30 27
86 185 16
169 51 182 212 53 97 185 ^
452 1 82 181 121 167 132 198 92 48 72 119 57 114 184 200 81 68 122 39
26 84 96
26 80 1 208 177 217 24 44 191 206 118 209 123 8 140 99 128 99 54 109
42 58 83
80 122 113 141 96 147 203 183 180 175 158 16 143 155 155 11 1 13 64 23
1 87 54

35 52 46 157 121 19 27 174 115 130 209 72 123 152 150 87 202 140 76 23
75 77 66
86 11 149 118 88 44 168 101 94 14 80 145 93 165 73 65 66 12 21 137 192
4 22 83
201 120 77 113 94 181 195 157 144 124 51 151 33 211 38 131 178 186 138
184 5 61
92 99 175 107 6 155 81 114 207 87 105 45 40 120 73 18 58 60 193 98 11
98 15 100
51 3 146 187 61 88 97 139 204 129 189 73 213 171 36 216 55 55 51 174
138 17 15
49 41 15 197 146 39 174 64 120 3 123 82 37 111 179 58 157 70 134 39
100 74 108
170 164 197 169 144 198 41 163 84 170 16 92 145 167 155 85 116 25 208
200 104
48 21 16 69 40 143 178 42 35 196 56 145 70 60 49 88 23 149 192 78 12
95 127 204
55 199 61 162 212 151 130 88 74 34 180 152 53 210 166 112 74 182 67
176 30 35
29 159 133 19 101 164 127 56 168 175 84 26 131 104 196 126 40 46 83 6
7 101 90
12 44 24 216 106 158 39 30 179 132 147 82 60 216 203 35 75 151 181 98
62 108 3
49 185 25 185 95 80 71 103 153 22 106 88 144 54 156 175 10 129 119 178
19 148
172 79 102 207 13 208 33 114 36 95 57 91 79 176 104 25 41 210 92 122
62 131 108
73 162 161 48 142 14 33 137 141 119 34 9 160 5 38 205 75 198 189 50
108 147 59
173 192 8 117 119 104 122 136 28 120 124 28 46 9 16 213 50 131 34 83
150 17 89
164 53 66 156 124 125 141 45 137 215 109 46 60 119 201 124 172 149 215
183 36
72 170 67 102 133 23 196 200 30 58 9 183 110 62 182 211 61 18 202 163
97 ^
463 1 45 50 116 145 8 64 204 150 74 93 184 89 111 39 108 108 85 61 13
36 154 61
210 115 3 84 108 22 152 209 216 103 176 123 126 57 183 10 199 107 186
161 105
51 38 69 184 26 202 15 171 57 97 77 24 90 33 183 167 70 83 29 110 67
49 123 63
148 159 190 148 2 99 172 159 33 160 26 139 3 44 17 46 89 40 101 87 62
69 27 38
6 55 2 162 184 183 190 211 131 6 212 210 138 144 132 85 124 25 60 91
112 95 99
212 30 126 120 195 34 19 158 71 28 211 217 56 167 7 71 122 70 66 79 65
134 27
61 147 103 199 44 57 137 65 50 202 78 80 81 137 128 12 177 22 129 68
168 23 74
30 55 182 174 208 3 21 96 218 112 158 111 16 187 177 5 179 140 82 59
52 188 36
20 214 62 29 156 62 46 106 107 29 70 129 91 89 174 51 15 45 136 49 100
142 217

182 88 55 48 4 163 20 174 74 156 196 113 151 90 96 92 191 7 134 134 94
106 130
170 1 106 193 79 25 146 24 67 112 32 190 85 210 75 15 108 45 73 37 193
205 166
194 1 169 102 206 65 197 129 51 218 198 208 66 33 120 155 163 43 95 5
133 78
103 2 185 105 114 167 93 4 77 87 102 57 3 218 175 67 135 56 18 165 92
168 86
203 157 212 13 59 75 109 187 217 96 155 203 31 71 185 82 193 119 116
122 204 16
7 48 201 22 89 197 192 63 125 18 39 171 17 109 101 64 141 138 50 53
133 101 127
153 45 207 104 121 188 13 216 63 85 147 166 108 68 214 188 53 87 39
148 96 178
124 182 48 67 72 44 214 125 41 63 219 97 211 194 135 16 30 189 132 219
191 70
43 158 134 157 118 37 107 52 94 60 68 191 47 122 110 157 55 25 83 12
43 72 201
202 101 1 12 114 10 2 28 117 138 207 144 109 164 76 165 178 101 41 90
190 171
143 30 32 41 23 11 121 31 25 123 103 136 135 150 97 14 115 66 213 132
62 203 94
83 209 19 24 4 100 8 150 160 ^
474 0 105 91 20 156 27 193 208 142 178 107 79 199 84 41 2 128 165 161
109 85 17
174 37 81 187 150 78 129 159 194 184 92 87 23 214 18 77 153 209 63 46
179 46 30
102 33 29 50 82 126 101 203 78 147 20 138 53 27 117 207 148 145 32 181
16 13 86
178 16 162 30 175 198 32 90 99 66 177 113 155 124 186 80 44 60 119 49
106 176
192 65 52 106 27 26 68 92 14 80 211 200 169 213 24 44 191 202 102 197
111 140
99 124 95 54 97 42 54 79 68 110 97 133 80 135 199 175 180 171 146 8
131 147 151
221 219 9 52 15 1 79 50 27 44 42 145 105 15 19 162 103 118 201 72 107
152 142
87 190 132 68 19 71 77 62 70 7 137 114 76 214 36 160 97 78 220 76 141
93 157 57
61 58 9 125 180 4 6 67 189 120 61 113 82 181 183 153 136 120 43 139 29
199 26
123 174 186 138 184 1 61 76 83 175 107 6 143 81 114 207 75 89 29 32
108 73 10
46 48 185 86 11 98 221 88 35 221 146 175 49 88 93 139 196 117 189 65
201 163 24
208 51 39 51 170 134 1 3 49 33 11 189 138 35 166 64 116 213 115 70 25
111 167
58 149 66 122 31 96 66 104 170 152 197 169 140 190 41 151 72 162 12 92
145 163
155 69 104 21 208 200 96 48 21 4 61 40 139 178 34 23 196 44 133 62 52
33 84 23

141 184 74 4 79 123 192 47 191 53 158 200 151 130 80 62 34 172 140 41
206 154
112 70 174 55 164 18 19 13 159 121 19 89 160 115 52 164 167 76 10 119
104 188
126 28 38 75 220 3 85 90 8 40 8 204 90 154 39 14 171 128 147 78 56 216
191 35
75 147 169 82 50 100 209 45 185 13 177 95 72 71 95 145 14 98 72 144 38
148 163
6 125 107 170 3 140 172 67 98 203 5 204 17 98 36 87 41 75 71 168 88 17
25 202
84 114 58 131 92 69 162 161 36 142 14 25 129 141 111 18 1 152 215 30
193 59 198
181 34 96 139 47 173 192 214 109 103 100 118 128 16 104 112 198 ^
466 0 25 2 213 213 36 124 13 62 150 221 61 150 39 45 149 124 104 134
38 130 208
109 18 32 112 201 96 158 149 215 176 22 44 149 39 88 133 220 189 193
23 44 9
183 89 48 161 204 54 222 195 163 167 40 45 106 140 8 49 204 145 69 83
169 89 96
24 93 103 85 41 3 31 154 51 210 115 213 79 108 12 147 209 206 103 176
113 116
37 173 10 189 102 176 156 100 36 33 69 174 16 202 5 171 52 82 72 4 80
23 178
162 70 83 29 105 67 34 118 43 143 154 185 138 2 79 157 149 33 160 21
129 208 24
2 31 69 35 81 72 47 64 22 38 221 55 222 152 179 168 180 211 121 1 207
200 133
129 117 80 114 25 50 76 102 85 84 212 30 111 110 195 29 14 153 51 28
196 212 36
167 222 61 107 55 46 64 65 119 220 12 56 137 93 194 29 57 137 50 210
30 197 78
65 71 132 123 7 177 22 129 68 168 3 74 10 50 167 174 198 218 6 86 218
97 158
106 6 177 162 210 164 125 62 59 52 173 16 20 214 42 9 141 57 46 106 97
24 50
119 86 84 169 46 15 25 136 44 100 127 202 182 83 35 33 209 158 15 159
64 156
191 108 146 75 96 72 181 217 134 129 74 96 115 165 216 91 188 59 10
141 14 62
107 12 190 65 205 60 10 98 30 63 22 178 205 151 179 211 164 82 206 60
192 114
31 203 188 193 66 28 105 140 148 33 75 5 128 68 83 212 170 95 109 152
88 224 67
77 102 42 3 213 170 52 125 56 3 165 82 163 76 193 157 197 218 49 70 89
182 217
91 145 203 11 56 180 77 193 114 116 122 189 11 7 43 191 17 74 187 187
53 125 13
19 166 222 104 91 54 131 123 50 33 118 86 127 153 45 197 84 121 183
223 201 48
70 132 166 88 48 204 188 53 82 29 148 81 163 119 177 28 57 52 34 209
115 31 63

204 97 206 194 120 1 25 184 132 214 176 65 28 148 124 147 118 37 87 42
94 60 58
186 47 112 110 152 45 15 68 217 43 57 186 187 218 ^
477 1 200 12 100 223 201 14 117 124 200 137 95 164 76 144 171 101 34
213 83 169
150 143 16 32 20 2 224 114 31 18 123 96 115 135 129 83 220 108 59 192
125 62
196 80 76 195 218 17 210 79 214 136 6 105 91 5 141 27 188 193 127 168
97 79 199
84 31 209 128 155 151 94 75 17 159 32 71 187 140 63 124 149 179 179 77
72 13
204 18 72 138 199 48 36 179 26 25 97 28 14 30 82 116 86 188 68 132 133
38 7 107
197 133 140 12 166 6 3 86 173 16 157 15 170 188 17 85 84 46 172 103
140 114 171
65 39 45 119 39 96 166 182 45 32 86 12 26 48 87 226 80 201 190 159 208
24 44
191 197 82 182 96 217 140 99 119 90 54 82 42 49 74 53 95 77 123 60 120
194 165
180 166 131 225 116 137 146 211 219 4 37 5 1 69 45 17 34 37 130 85 10
9 147 88
103 191 72 87 152 132 87 175 122 58 14 66 77 57 50 2 122 109 61 209 26
150 92
58 205 71 136 93 147 37 56 48 212 221 110 165 4 213 47 174 120 41 113
67 181
168 148 126 115 33 124 24 184 11 113 169 186 138 184 223 61 56 63 175
107 6 128
81 114 207 60 69 9 22 93 73 31 33 175 71 11 98 206 73 15 221 146 160
34 88 88
139 186 102 189 55 186 153 9 198 46 19 51 165 129 208 215 49 23 6 179
128 30
156 64 111 203 105 55 10 111 152 58 139 61 107 21 91 56 99 170 137 197
169 135
180 41 136 57 152 7 92 145 158 155 49 89 16 208 200 86 48 21 216 51 40
134 178
24 8 196 29 118 52 42 13 79 23 131 174 69 221 59 118 177 37 181 43 153
185 151
130 70 47 34 162 125 26 201 139 112 65 164 40 149 3 226 220 159 106 19
74 155
100 47 159 157 66 217 104 104 178 126 13 28 65 215 225 65 90 3 35 215
189 70
149 39 221 161 123 147 73 51 216 176 35 75 142 154 62 35 90 194 40 185
225 167
95 62 71 85 135 4 88 52 144 18 138 148 1 120 92 160 210 130 172 52 93
198 222
199 224 19 ^
479 0 36 73 13 47 57 154 60 3 226 188 70 100 51 131 64 62 162 161 15
142 14 11
115 141 97 219 216 138 201 16 172 31 198 167 6 75 125 26 173 192 193
95 75 93
111 114 224 76 91 213 13 227 201 213 28 120 1 50 150 213 45 142 31 33
145 124

92 130 34 126 204 109 2 16 108 201 80 150 149 215 172 14 28 137 23 80
133 208
185 189 19 36 9 183 77 40 149 200 50 214 191 163 159 36 41 98 136 8 37
204 141
65 75 157 89 84 12 81 99 85 25 224 27 154 43 210 115 205 75 108 4 143
209 198
103 176 105 108 21 165 10 181 98 168 152 96 24 29 69 166 8 202 226 171
48 70 68
217 72 15 174 158 70 83 29 101 67 22 114 27 139 150 181 130 2 63 145
141 33 160
17 121 196 8 219 19 53 31 65 60 35 60 18 38 217 55 222 144 175 156 172
211 113
226 203 192 129 117 105 76 106 25 42 64 94 77 72 212 30 99 102 195 25
10 149 35
28 184 208 20 167 218 53 95 43 30 52 65 107 220 52 129 85 190 17 57
137 38 202
14 193 78 53 63 128 119 3 177 22 129 68 168 216 74 223 46 155 174 190
214 223
78 218 85 158 102 227 169 150 198 152 113 46 59 52 161 20 214 26 222
129 53 46
106 89 20 34 111 82 80 165 42 15 9 136 40 100 115 190 182 79 19 21 197
154 11
147 56 156 187 104 142 63 96 56 173 209 134 125 58 88 103 161 212 79
184 43 227
137 6 58 103 225 190 49 201 48 6 90 18 55 10 166 205 139 167 203 160
66 206 56
188 102 15 191 180 181 66 24 93 128 136 25 59 5 124 60 67 204 158 87
105 140 84
224 59 69 102 30 3 209 166 40 117 56 220 165 74 159 68 185 157 185 206
41 66 73
178 217 87 137 203 224 44 176 73 193 110 116 122 177 7 7 39 183 13 62
179 183
45 125 9 3 162 210 100 83 46 123 111 50 17 106 74 127 153 45 189 68
121 179 215
189 36 58 120 166 72 32 196 188 53 78 21 148 69 151 115 173 12 49 36
26 205 107
23 63 192 112 ^
485 1 200 194 102 214 19 178 132 208 158 59 10 136 112 135 118 37 63
30 94 60
46 180 47 100 110 146 33 3 50 199 43 39 168 169 90 188 12 92 219 189 6
117 116
196 133 87 164 76 132 167 101 30 209 79 157 138 143 8 32 8 221 220 110
31 14
123 92 103 135 117 75 212 104 55 180 121 62 192 72 72 187 206 13 202
67 206 128
221 105 91 224 129 27 184 181 115 160 89 79 199 84 23 197 128 147 143
82 67 17
147 28 63 187 132 51 120 141 167 175 65 60 5 196 18 68 126 191 36 28
179 10 21
93 24 2 14 82 108 74 176 60 120 215 129 26 222 99 189 121 136 227 154
229 226

86 169 16 153 3 166 180 5 81 72 30 168 95 128 106 159 53 35 33 119 31
88 158
174 29 16 70 26 32 83 218 80 193 182 151 204 24 44 191 193 66 170 84
213 140 99
115 86 54 70 42 45 70 41 83 61 115 44 108 190 157 180 162 119 221 104
129 142
203 219 25 228 1 61 41 9 26 33 118 69 6 1 135 76 91 183 72 71 152 124
87 163
114 50 10 62 77 53 34 229 110 105 49 205 18 142 88 42 193 67 132 93
139 21 52
40 204 213 98 153 4 201 31 162 120 25 113 55 181 156 144 118 111 25
112 20 172
230 105 165 186 138 184 223 61 40 47 175 107 6 116 81 114 207 48 53
224 14 81
73 223 19 21 167 59 11 98 194 61 230 221 146 148 22 88 84 139 178 90
189 47 174
145 228 190 42 3 51 161 125 196 207 49 15 2 171 120 26 148 64 107 195
97 43 229
111 140 58 131 57 95 13 87 48 95 170 125 197 169 131 172 41 124 45 144
3 92 145
154 155 33 77 12 208 200 78 48 21 208 43 40 130 178 16 227 196 17 106
44 34 228
75 23 123 166 65 217 43 114 165 29 173 35 149 173 151 130 62 35 34 154
113 14
197 127 112 61 156 28 137 222 214 208 159 94 19 62 151 88 43 155 149
58 205 92
104 170 126 1 20 57 211 225 49 90 230 31 203 177 54 145 39 209 153 119
147 69
47 216 164 35 75 138 142 46 23 82 182 36 185 198 ^
483 1 155 95 50 71 73 123 225 76 28 144 227 126 130 228 114 74 148 192
118 172
34 87 192 216 193 206 54 36 65 230 31 49 146 44 228 214 180 62 92 47
131 48 58
162 161 3 142 14 3 107 141 89 207 212 130 193 8 160 15 198 159 223 63
117 14
173 192 181 87 59 89 107 106 216 60 79 201 1 227 189 213 20 116 222 38
150 205
29 134 23 21 141 124 80 126 30 122 200 109 219 104 201 64 142 149 215
168 6 12
125 7 72 133 196 181 185 15 28 9 183 65 32 137 196 46 206 187 163 151
32 37 90
132 8 25 204 137 61 67 145 89 72 69 95 85 9 220 23 154 35 210 115 197
71 108
229 139 209 190 103 176 97 100 5 157 10 173 94 160 148 92 12 25 69 158
202 222
171 44 58 64 205 64 7 170 154 70 83 29 97 67 10 110 11 135 146 177 122
2 47 133
133 33 160 13 113 184 225 211 7 37 27 49 48 23 56 14 38 213 55 222 136
171 144
164 211 105 226 199 184 125 105 93 72 98 25 34 52 86 69 60 212 30 87
94 195 21

6 145 19 28 172 204 4 167 214 45 83 31 14 40 65 95 220 221 48 121 77
186 5 57
137 26 194 231 189 78 41 55 124 115 232 177 22 129 68 168 204 74 211
42 143 174
182 210 215 70 218 73 158 98 223 161 138 186 140 101 30 59 52 149 217
20 214 10
210 117 49 46 106 81 16 18 103 78 76 161 38 15 226 136 36 100 103 178
182 75 3
9 185 150 7 135 48 156 183 100 138 51 96 40 165 201 134 121 42 80 91
157 208 67
180 27 219 133 231 54 99 213 190 33 197 36 2 82 6 47 231 154 205 127
155 195
156 50 206 52 184 90 232 179 172 169 66 20 81 116 124 17 43 5 120 52
51 196 146
79 101 128 80 224 51 61 102 18 3 205 162 28 109 56 212 165 66 155 60
177 157
173 194 33 62 57 174 217 83 129 203 212 32 172 69 193 106 116 122 165
3 7 35
175 9 50 171 179 37 125 5 220 158 198 96 75 38 115 99 50 1 94 62 127
153 45 181
52 121 175 207 177 24 46 108 164 ^
499 0 44 4 182 188 53 71 7 148 48 130 108 166 220 35 8 12 198 93 9 63
171 97
195 194 87 204 14 173 132 203 143 54 231 126 102 125 118 37 43 20 94
60 36 175
47 90 110 141 23 229 35 184 43 24 153 154 85 173 12 82 214 174 232 117
106 191
128 77 164 76 117 162 101 25 204 74 142 123 143 234 32 229 211 215 105
31 9 123
87 88 135 102 65 202 99 50 165 116 62 187 62 67 177 191 8 192 52 196
118 206
105 91 214 114 27 179 166 100 150 79 79 199 84 13 182 128 137 133 67
57 17 132
23 53 187 122 36 115 131 152 170 50 45 231 186 18 63 111 181 21 18 179
226 16
88 19 223 230 82 98 59 161 50 105 200 124 11 207 89 179 106 131 212
139 224 221
86 164 16 148 224 161 170 226 76 57 10 163 85 113 96 144 38 30 18 119
21 78 148
164 9 232 50 221 26 12 78 208 80 183 172 141 199 24 44 191 188 46 155
69 208
140 99 110 81 54 55 42 40 65 26 68 41 105 24 93 185 147 180 157 104
216 89 119
137 193 219 231 10 223 1 51 36 235 16 28 103 49 1 227 120 61 76 173 72
51 152
114 87 148 104 40 5 57 77 48 14 229 95 100 34 200 8 132 83 22 178 62
127 93 129
1 47 30 194 203 83 138 4 186 11 147 120 5 113 40 181 141 139 108 106
15 97 15
157 220 95 160 186 138 184 223 61 20 27 175 107 6 101 81 114 207 33 33
209 4 66

73 218 4 6 157 44 11 98 179 46 215 221 146 133 7 88 79 139 168 75 189
37 159
135 218 180 37 219 51 156 120 181 197 49 5 233 161 110 21 138 64 102
185 87 28
219 111 125 58 121 52 80 3 82 38 90 170 110 197 169 126 162 41 109 30
134 234
92 145 149 155 13 62 7 208 200 68 48 21 198 33 40 125 178 6 217 196 2
91 34 24
213 70 23 113 156 60 212 23 109 150 19 163 25 144 158 151 130 52 20 34
144 98
235 192 112 112 56 146 13 122 212 199 193 159 79 19 47 146 73 38 150
139 48 190
77 104 160 126 222 10 47 206 225 29 90 230 26 188 162 34 140 39 194
143 114 147
64 42 216 149 174 ^
473 1 75 131 121 18 2 68 161 29 185 203 145 95 40 71 63 113 220 66 8
144 212
116 115 228 109 59 138 177 108 172 19 82 187 211 188 191 34 36 55 215
11 39 136
24 223 199 170 52 82 42 131 28 53 162 161 226 142 14 231 97 141 79 192
207 120
183 236 145 233 198 149 208 48 107 237 173 192 166 77 39 84 102 96 206
40 64
186 224 227 174 213 10 111 212 23 150 195 9 124 13 6 136 124 65 121 25
117 195
109 204 218 99 201 44 132 149 215 163 234 230 110 225 62 133 181 176
180 10 18
9 183 50 22 122 191 41 196 182 163 141 27 32 80 127 8 10 204 132 56 57
130 89
57 223 54 90 85 227 215 18 154 25 210 115 187 66 108 224 134 209 180
103 176 87
90 223 147 10 163 89 150 143 87 235 20 69 148 228 202 217 171 39 43 59
190 54
235 165 149 70 83 29 92 67 233 105 229 130 141 172 112 2 27 118 123 33
160 8
103 169 210 201 230 17 22 29 33 8 51 9 38 208 55 222 126 166 129 154
211 95 226
194 174 120 90 78 67 88 25 24 37 76 59 45 212 30 72 84 195 16 1 140
237 28 157
199 222 167 209 35 68 16 232 25 65 80 220 211 43 111 67 181 228 57 137
11 184
216 184 78 26 45 119 110 232 177 22 129 68 168 189 74 196 37 128 174
172 205
205 60 218 58 158 93 218 151 123 171 125 86 10 59 52 134 202 20 214
228 195 102
44 46 106 71 11 236 93 73 71 156 33 15 211 136 31 100 88 163 182 70
221 232 170
145 2 120 38 156 178 95 133 36 96 20 155 191 134 116 22 70 76 152 203
52 175 7
209 128 226 49 94 198 190 13 192 21 235 72 229 37 221 139 205 112 140
185 151

30 206 47 179 75 217 164 162 154 66 15 66 101 109 7 23 5 115 42 31 186
131 69
96 113 75 224 41 51 102 3 3 200 157 13 99 56 202 165 56 150 50 167 157
158 179
23 57 37 169 217 78 119 203 197 17 167 64 193 101 116 122 150 236 7 30
165 4 35
161 174 27 125 205 153 148 ^
502 1 89 61 24 101 78 50 213 73 41 127 153 45 167 24 121 168 193 156 3
25 87
166 28 228 174 188 53 67 239 148 36 118 104 162 208 27 232 4 194 85 1
63 159 97
191 194 75 196 10 169 132 199 131 50 223 118 94 117 118 37 27 12 94 60
28 171
47 82 110 137 15 225 23 172 43 12 141 142 81 161 12 74 210 162 228 117
98 187
124 69 164 76 105 158 101 21 200 70 130 111 143 230 32 221 203 211 101
31 5 123
83 76 135 90 57 194 95 46 153 112 62 183 54 63 169 179 4 184 40 188
110 194 105
91 206 102 27 175 154 88 142 71 79 199 84 5 170 128 129 125 55 49 17
120 19 45
187 114 24 111 123 140 166 38 33 227 178 18 59 99 173 9 10 179 214 12
84 15 215
218 82 90 47 149 42 93 188 120 239 195 81 171 94 127 200 127 220 217
86 160 16
144 216 157 162 218 72 45 234 159 77 101 88 132 26 26 6 119 13 70 140
156 233
220 34 213 26 236 74 200 80 175 164 133 195 24 44 191 184 30 143 57
204 140 99
106 77 54 43 42 36 61 14 56 25 97 8 81 181 139 180 153 92 212 77 111
133 185
219 231 238 219 1 43 32 231 8 24 91 33 237 223 108 49 64 165 72 35 152
106 87
136 96 32 1 53 77 44 238 229 83 96 22 196 124 79 6 166 58 123 93 121
225 43 22
186 195 71 126 4 174 235 135 120 229 113 28 181 129 135 100 102 7 85
11 145 212
87 156 186 138 184 223 61 4 11 175 107 6 89 81 114 207 21 17 197 236
54 73 214
232 234 149 32 11 98 167 34 203 221 146 121 235 88 75 139 160 63 189
29 147 127
210 172 33 207 51 152 116 169 189 49 237 233 153 102 17 130 64 98 177
79 16 211
111 113 58 113 48 68 235 78 30 86 170 98 197 169 122 154 41 97 18 126
234 92
145 145 155 237 50 3 208 200 60 48 21 190 25 40 121 178 238 209 196
230 79 26
16 201 66 23 105 148 56 208 7 105 138 11 155 17 140 146 151 130 44 8
34 136 86
227 188 100 112 52 138 1 110 204 187 181 159 67 19 35 142 61 34 146
131 40 178
65 104 152 126 214 2 39 202 173 ^

486 1 5 90 230 20 170 144 10 134 39 176 131 108 147 58 36 216 131 35
75 127 109
2 232 60 149 25 185 195 137 95 32 71 55 105 216 58 234 144 200 108 103
228 105
47 130 165 100 172 7 78 183 207 184 179 18 36 47 203 237 31 128 8 219
187 162
44 74 38 131 12 49 162 161 218 142 14 227 89 141 71 180 203 112 175
232 133 221
198 141 196 36 99 229 173 192 154 69 23 80 98 88 198 24 52 174 216 227
162 213
2 107 204 11 150 187 235 116 5 236 132 124 53 117 21 113 191 109 192
206 95 201
28 124 149 215 159 230 218 98 213 54 133 169 172 176 6 10 9 183 38 14
110 187
37 188 178 163 133 23 28 72 123 8 240 204 128 52 49 118 89 45 215 42
86 85 215
211 14 154 17 210 115 179 62 108 220 130 209 172 103 176 79 82 211 139
10 155
85 142 139 83 227 16 69 140 224 202 213 171 35 31 55 178 46 231 161
145 70 83
29 88 67 225 101 217 126 137 168 104 2 11 106 115 33 160 4 95 157 198
193 222 1
18 13 21 238 47 5 38 204 55 222 118 162 117 146 211 87 226 190 166 116
78 66 63
80 25 16 25 68 51 33 212 30 60 76 195 12 239 136 225 28 145 195 210
167 205 27
56 4 220 13 65 68 220 203 39 103 59 177 220 57 137 241 176 204 180 78
14 37 115
106 232 177 22 129 68 168 177 74 184 33 116 174 164 201 197 52 218 46
158 89
214 143 111 159 113 74 236 59 52 122 190 20 214 216 183 90 40 46 106
63 7 224
85 69 67 152 29 15 199 136 27 100 76 151 182 66 209 224 158 141 240
108 30 156
174 91 129 24 96 4 147 183 134 112 6 62 64 148 199 40 171 233 201 124
222 45 90
186 190 239 188 9 235 64 221 29 213 127 205 100 128 177 147 14 206 43
175 63
205 152 154 142 66 11 54 89 97 241 7 5 111 34 15 178 119 61 92 101 71
224 33 43
102 233 3 196 153 1 91 56 194 165 48 146 42 159 157 146 167 15 53 21
165 217 74
111 203 185 5 163 60 193 97 116 122 138 236 7 26 157 23 153 170 184 ^
505 1 125 238 187 147 165 85 53 16 93 66 50 201 61 29 127 153 45 159 8
121 164
185 144 235 13 75 166 12 216 166 188 53 63 235 148 24 106 100 158 196
19 220
240 190 77 237 63 147 97 187 194 63 188 6 165 132 195 119 46 215 110
86 109 118
37 11 4 94 60 20 167 47 74 110 133 7 221 11 160 43 129 130 77 149 12
66 206 150

224 117 90 183 120 61 164 76 93 154 101 17 196 66 118 99 143 226 32
213 195 207
97 31 1 123 79 64 135 78 49 186 91 42 141 108 62 179 46 59 161 167 176
28 180
102 182 105 91 198 90 27 171 142 76 134 63 79 199 84 241 158 128 121
117 43 41
17 108 15 37 187 106 12 107 115 128 162 26 21 223 170 18 55 87 165 241
2 179
202 8 80 11 207 206 82 82 35 137 34 81 176 116 231 183 73 163 82 123
188 115
216 213 86 156 16 140 208 153 154 210 68 33 222 155 69 89 80 120 14 22
238 119
5 62 132 148 221 208 18 205 26 224 70 192 80 167 156 125 191 24 44 191
180 14
131 45 200 140 99 102 73 54 31 42 32 57 2 44 9 89 236 69 177 131 180
149 80 208
65 103 129 177 219 231 230 215 1 35 28 227 20 79 17 237 219 96 37 52
157 72 19
152 98 87 124 88 24 241 49 77 40 226 229 71 92 10 192 236 116 75 234
154 54 119
93 113 213 39 14 178 187 59 114 4 162 223 123 120 217 113 16 181 117
131 92 98
243 73 7 133 204 79 152 186 138 184 223 61 232 239 175 107 6 77 81 114
207 9 1
185 232 42 73 210 224 226 141 20 11 98 155 22 191 221 146 109 227 88
71 139 152
51 189 21 135 119 202 164 29 195 51 148 112 157 181 49 233 233 145 94
13 122 64
94 169 71 4 203 111 101 58 105 44 56 231 74 22 82 170 86 197 169 118
146 41 85
6 118 234 92 145 141 155 225 38 243 208 200 52 48 21 182 17 40 117 178
234 201
196 222 67 18 8 189 62 23 97 140 52 204 235 101 126 3 147 9 136 134
151 130 36
240 34 128 74 219 184 88 112 48 130 233 98 196 175 169 159 55 19 23
138 49 30
142 123 32 166 53 104 144 126 206 238 31 198 203 ^
483 1 235 90 230 16 158 132 240 130 39 164 123 104 147 54 32 216 119
35 75 123
97 232 224 52 137 21 185 187 129 95 24 71 47 97 212 50 222 144 188 100
91 228
101 35 122 153 92 172 241 74 179 203 180 167 2 36 39 191 225 23 120
238 215 175
154 36 66 34 131 242 45 162 161 210 142 14 223 81 141 63 168 199 104
167 228
121 209 198 133 184 24 91 221 173 192 142 61 7 76 94 80 190 8 40 162
208 227
150 213 240 103 196 245 150 179 223 108 243 228 128 124 41 113 17 109
187 109
180 194 91 201 12 116 149 215 155 226 206 86 201 46 133 157 168 172 2
2 9 183

26 6 98 183 33 180 174 163 125 19 24 64 119 8 232 204 124 48 41 106 89
33 207
30 82 85 203 207 10 154 9 210 115 171 58 108 216 126 209 164 103 176
71 74 199
131 10 147 81 134 135 79 219 12 69 132 220 202 209 171 31 19 51 166 38
227 157
141 70 83 29 84 67 217 97 205 122 133 164 96 2 241 94 107 33 160 87
145 186 185
214 231 14 243 9 230 43 1 38 200 55 222 110 158 105 138 211 79 226 186
158 112
66 54 59 72 25 8 13 60 43 21 212 30 48 68 195 8 239 132 213 28 133 191
198 167
201 19 44 238 208 1 65 56 220 195 35 95 51 173 212 57 137 233 168 192
176 78 2
29 111 102 232 177 22 129 68 168 165 74 172 29 104 174 156 197 189 44
218 34
158 85 210 135 99 147 101 62 224 59 52 110 178 20 214 204 171 78 36 46
106 55 3
212 77 65 63 148 25 15 187 136 23 100 64 139 182 62 197 216 146 137
240 96 22
156 170 87 125 12 96 234 139 175 134 108 236 54 52 144 195 28 167 221
193 120
218 41 86 174 190 227 184 243 235 56 213 21 205 115 205 88 116 169 143
244 206
39 171 51 193 140 146 130 66 7 42 77 85 237 237 5 107 26 245 170 107
53 88 89
67 224 25 35 102 225 3 192 149 235 83 56 186 165 40 142 34 151 157 134
155 7 49
5 161 217 70 103 203 173 239 159 56 193 93 116 122 126 236 7 22 149
242 178 ^
507 1 141 164 7 125 238 175 143 153 81 45 8 85 54 50 189 49 17 127 153
45 151
240 121 160 177 132 227 1 63 166 244 204 158 188 53 59 231 148 12 94
96 154 184
11 208 236 186 69 233 63 135 97 183 194 51 180 2 161 132 191 107 42
207 102 78
101 118 37 243 244 94 60 12 163 47 66 110 129 247 217 247 148 43 236
117 118 73
137 12 58 202 138 220 117 82 179 116 53 164 76 81 150 101 13 192 62
106 87 143
222 32 205 187 203 93 31 245 123 75 52 135 66 41 178 87 38 129 104 62
175 38 55
153 155 244 168 16 172 94 170 105 91 190 78 27 167 130 64 126 55 79
199 84 237
146 128 113 109 31 33 17 96 11 29 187 98 103 107 116 158 14 9 219 162
18 51 75
157 233 242 179 190 4 76 7 199 194 82 74 23 125 26 69 164 112 223 171
65 155 70
119 176 103 212 209 86 152 16 136 200 149 146 202 64 21 210 151 61 77
72 108 2
18 230 119 245 54 124 140 209 196 2 197 26 212 66 184 80 159 148 117
187 24 44

191 176 246 119 33 196 140 99 98 69 54 19 42 28 53 238 32 241 81 224
57 173 123
180 145 68 204 53 95 125 169 219 231 222 211 1 27 24 223 240 16 67 1
237 215 84
25 40 149 72 3 152 90 87 112 80 16 241 45 77 36 214 229 59 88 246 188
232 108
71 222 142 50 115 93 105 201 35 6 170 179 47 102 4 150 211 111 120 205
113 4
181 105 127 84 94 239 61 3 121 196 71 148 186 138 184 223 61 220 227
175 107 6
65 81 114 207 245 233 173 228 30 73 206 216 218 133 8 11 98 143 10 179
221 146
97 219 88 67 139 144 39 189 13 123 111 194 156 25 183 51 144 108 145
173 49 229
233 137 86 9 114 64 90 161 63 240 195 111 89 58 97 40 44 227 70 14 78
170 74
197 169 114 138 41 73 242 110 234 92 145 137 155 213 26 243 208 200 44
48 21
174 9 40 113 178 230 193 196 214 55 10 177 58 23 89 132 48 200 223 97
114 243
139 1 132 122 151 130 28 232 34 120 62 211 180 76 112 44 122 225 86
188 163 157
159 43 19 11 134 37 26 138 115 24 154 41 104 136 126 198 234 140 ^
490 1 192 225 223 90 230 12 146 120 228 126 39 152 115 100 147 50 28
216 107 35
75 119 85 220 216 44 125 17 185 179 121 95 16 71 39 89 208 42 210 144
176 92 79
228 97 23 114 141 84 172 233 70 175 199 176 155 236 36 31 179 213 15
112 226
211 163 146 28 58 30 131 230 41 162 161 202 142 14 219 73 141 55 156
195 96 159
224 109 197 198 125 172 12 83 213 173 192 130 53 241 72 90 72 182 242
28 150
200 227 138 213 236 99 188 237 150 171 211 100 239 220 124 124 29 109
13 105
183 109 168 182 87 201 246 108 149 215 151 222 194 74 189 38 133 145
164 168
248 244 9 183 14 248 86 179 29 172 170 163 117 15 20 56 115 8 224 204
120 44 33
94 89 21 199 18 78 85 191 203 6 154 1 210 115 163 54 108 212 122 209
156 103
176 63 66 187 123 10 139 77 126 131 75 211 8 69 124 216 202 205 171 27
7 47 154
30 223 153 137 70 83 29 80 67 209 93 193 118 129 160 88 2 229 82 99 33
160 246
79 133 174 177 206 219 10 231 247 222 39 247 38 196 55 222 102 154 93
130 211
71 226 182 150 108 54 42 55 64 25 1 52 35 9 212 30 36 60 195 4 239 128
201 28
121 187 186 167 197 11 32 230 196 239 65 44 220 187 31 87 43 169 204
57 137 225

160 180 172 78 240 21 107 98 232 177 22 129 68 168 153 74 160 25 92
174 148 193
181 36 218 22 158 81 206 127 87 135 89 50 212 59 52 98 166 20 214 192
159 66 32
46 106 47 249 200 69 61 59 144 21 15 175 136 19 100 52 127 182 58 185
208 134
133 240 84 14 156 166 83 121 96 222 131 167 134 104 224 46 40 140 191
16 163
209 185 116 214 37 82 162 190 215 180 235 235 48 205 13 197 103 205 76
104 161
139 232 206 35 167 39 181 128 138 118 66 3 30 65 73 233 225 5 103 18
233 162 95
45 84 77 63 224 17 27 102 217 3 188 145 227 75 56 178 165 32 138 26
143 157 122
143 249 45 239 157 217 66 95 203 161 231 155 52 193 89 116 122 114 236
7 18 141
242 249 137 162 3 125 238 69 ^
514 1 139 141 77 37 77 42 50 177 37 5 127 153 45 143 228 121 156 169
120 219
241 51 166 232 192 150 188 53 55 227 148 82 92 150 172 3 196 232 182
61 229 63
123 97 179 194 39 172 250 157 132 187 95 38 199 94 70 93 118 37 231
240 94 60 4
159 47 58 110 125 243 213 239 136 43 228 105 106 69 125 12 50 198 126
216 117
74 175 112 45 164 76 69 146 101 9 188 58 94 75 143 218 32 197 179 199
89 31 245
123 71 40 135 54 33 170 83 34 117 100 62 171 30 51 145 143 244 160 4
164 86 158
105 91 182 66 27 163 118 52 118 47 79 199 84 233 134 128 105 101 19 25
17 84 7
21 187 90 240 99 99 104 154 2 249 215 154 18 47 63 149 225 238 179 178
72 3 191
182 82 66 11 113 18 57 152 108 215 159 57 147 58 115 164 91 208 205 86
148 16
132 192 145 138 194 60 9 198 147 53 65 64 96 242 14 222 119 241 46 116
132 197
184 238 189 26 200 62 176 80 151 140 109 183 24 44 191 172 234 107 21
192 140
99 94 65 54 7 42 24 49 230 20 229 73 212 45 169 115 180 141 56 200 41
87 121
161 219 231 214 207 1 19 20 219 236 12 55 237 237 211 72 13 28 141 72
239 152
82 87 100 72 8 241 41 77 32 202 229 47 84 238 184 228 100 67 210 130
46 111 93
97 189 31 250 162 171 35 90 4 138 199 99 120 193 113 244 181 93 123 76
90 235
49 251 109 188 63 144 186 138 184 223 61 208 215 175 107 6 53 81 114
207 237
221 161 224 18 73 202 208 210 125 248 11 98 131 250 167 221 146 85 211
88 63

139 136 27 189 5 111 103 186 148 21 171 51 140 104 133 165 49 225 233
129 78 5
106 64 86 153 55 232 187 111 77 58 89 36 32 223 66 6 74 170 62 197 169
110 130
41 61 234 102 234 92 145 133 155 201 14 243 208 200 36 48 21 166 1 40
109 178
226 185 196 206 43 2 244 165 54 23 81 124 44 196 211 93 102 239 131
245 128 110
151 130 20 224 34 112 50 203 176 64 112 40 114 217 74 180 151 145 159
31 19 251
130 25 22 134 107 16 142 29 104 128 126 190 230 15 190 225 217 90 230
10 140
114 222 124 39 146 111 217 ^
510 0 147 46 24 216 95 35 75 115 73 208 208 36 113 13 185 171 113 95 8
71 31 81
204 34 198 144 164 84 67 228 93 11 106 129 76 172 225 66 171 195 172
143 224 36
23 167 201 7 104 214 207 151 138 20 50 26 131 218 37 162 161 194 142
14 215 65
141 47 144 191 88 151 220 97 185 198 117 160 75 205 173 192 118 45 229
68 86 64
174 230 16 138 192 227 126 213 232 95 180 229 150 163 199 92 235 212
120 124 17
105 9 101 179 109 156 170 83 201 234 100 149 215 147 218 182 62 177 30
133 133
160 164 248 240 9 183 2 244 74 175 25 164 166 163 109 11 16 48 111 8
216 204
116 40 25 82 89 9 191 6 74 85 179 199 2 154 247 210 115 155 50 108 208
118 209
148 103 176 55 58 175 115 10 131 73 118 127 71 203 4 69 116 212 202
201 171 23
249 43 142 22 219 149 133 70 83 29 76 67 201 89 181 114 125 156 80 2
217 70 91
33 160 246 71 121 162 169 198 207 6 219 239 214 35 247 38 192 55 222
94 150 81
122 211 63 226 178 142 104 42 30 51 56 25 246 243 44 27 251 212 30 24
52 195
239 124 189 28 109 183 174 167 193 3 20 222 184 231 65 32 220 179 27
79 35 165
196 57 137 217 152 168 168 78 232 13 103 94 232 177 22 129 68 168 141
74 148 21
80 174 140 189 173 28 218 10 158 77 202 119 75 123 77 38 200 59 52 86
154 20
214 180 147 54 28 46 106 39 249 188 61 57 55 140 17 15 163 136 15 100
40 115
182 54 173 200 122 129 240 72 6 156 162 79 117 242 96 210 123 159 134
100 212
38 28 136 187 4 159 197 177 112 210 33 78 150 190 203 176 227 235 40
197 5 189
91 205 64 92 153 135 220 206 31 163 27 169 116 130 106 66 253 18 53 61
229 213

5 99 10 221 154 83 37 80 65 59 224 9 19 102 209 3 184 141 219 67 56
170 165 24
134 18 135 157 110 131 245 41 227 153 217 62 87 203 149 223 151 48 193
85 116
122 102 236 7 14 133 242 241 129 158 249 125 238 157 137 135 75 33 250
73 36 50
171 31 253 127 153 45 139 222 121 154 165 114 215 237 45 166 226 186
146 188 53
53 225 148 248 118 ^
533 0 87 145 157 250 181 227 177 51 224 63 108 97 174 194 24 162 250
152 132
182 80 33 189 84 60 83 118 37 216 235 94 60 251 154 47 48 110 120 238
208 229
121 43 218 90 91 64 110 12 40 193 111 211 117 64 170 107 35 164 76 54
141 101 4
183 53 79 60 143 213 32 187 169 194 84 31 245 123 66 25 135 39 23 160
78 29 102
95 62 166 20 46 135 128 244 150 246 154 76 143 105 91 172 51 27 158
103 37 108
37 79 199 84 228 119 128 95 91 4 15 17 69 2 11 187 80 230 94 89 89 149
244 239
210 144 18 42 48 139 215 233 179 163 252 67 255 181 167 82 56 253 98 8
42 137
103 205 144 47 137 43 110 149 76 203 200 86 143 16 127 182 140 128 184
55 251
183 142 43 50 54 81 232 9 212 119 236 36 106 122 182 169 223 179 26
185 57 166
80 141 130 99 178 24 44 191 167 219 92 6 187 140 99 89 60 54 249 42 19
44 220 5
214 63 197 30 164 105 180 136 41 195 26 77 116 151 219 231 204 202 1 9
15 214
231 7 40 222 237 206 57 255 13 131 72 224 152 72 87 85 62 255 241 36
77 27 187
229 32 79 228 179 223 90 62 195 115 41 106 93 87 174 26 245 152 161 20
75 4 123
184 84 120 178 113 234 181 78 118 66 85 230 34 251 94 178 53 139 186
138 184
223 61 193 200 175 107 6 38 81 114 207 227 206 146 219 3 73 197 198
200 115 238
11 98 116 240 152 221 146 70 201 88 58 139 126 12 189 252 96 93 176
138 16 156
51 135 99 118 155 49 220 233 119 68 96 64 81 143 45 222 177 111 62 58
79 31 17
218 61 253 69 170 47 197 169 105 120 41 46 224 92 234 92 145 128 155
186 256
243 208 200 26 48 21 156 248 40 104 178 221 175 196 196 28 249 239 150
49 23 71
114 39 191 196 88 87 234 121 240 123 95 151 130 10 214 34 102 35 193
171 49 112
35 104 207 59 170 136 130 159 16 19 241 125 10 17 129 97 6 127 14 104
118 126

180 225 5 185 225 202 90 230 5 125 99 207 119 39 131 101 93 147 43 21
216 86 35
75 112 64 199 202 30 104 10 185 165 107 95 2 71 25 75 201 28 189 144
155 78 58
228 90 2 100 120 70 172 219 63 168 192 169 134 215 36 17 158 192 1 98
205 137 ^
522 1 139 130 12 42 22 131 206 33 162 161 186 142 14 211 57 141 39 132
187 80
143 216 85 173 198 109 148 246 67 197 173 192 106 37 217 64 82 56 166
218 4 126
184 227 114 213 228 91 172 221 150 155 187 84 231 204 116 124 5 101 5
97 175
109 144 158 79 201 222 92 149 215 143 214 170 50 165 22 133 121 156
160 248 236
9 183 248 240 62 171 21 156 162 163 101 7 12 40 107 8 208 204 112 36
17 70 89
255 183 252 70 85 167 195 256 154 243 210 115 147 46 108 204 114 209
140 103
176 47 50 163 107 10 123 69 110 123 67 195 69 108 208 202 197 171 19
241 39 130
14 215 145 129 70 83 29 72 67 193 85 169 110 121 152 72 2 205 58 83 33
160 246
63 109 150 161 190 195 2 207 231 206 31 247 38 188 55 222 86 146 69
114 211 55
226 174 134 100 30 18 47 48 25 242 235 36 19 243 212 30 12 44 195 254
239 120
177 28 97 179 162 167 189 253 8 214 172 223 65 20 220 171 23 71 27 161
188 57
137 209 144 156 164 78 224 5 99 90 232 177 22 129 68 168 129 74 136 17
68 174
132 185 165 20 218 256 158 73 198 111 63 111 65 26 188 59 52 74 142 20
214 168
135 42 24 46 106 31 249 176 53 53 51 136 13 15 151 136 11 100 28 103
182 50 161
192 110 125 240 60 256 156 158 75 113 234 96 198 115 151 134 96 200 30
16 132
183 250 155 185 169 108 206 29 74 138 190 191 172 219 235 32 189 255
181 79 205
52 80 145 131 208 206 27 159 15 157 104 122 94 66 253 6 41 49 225 201
5 95 2
209 146 71 29 76 53 55 224 1 11 102 201 3 180 137 211 59 56 162 165 16
130 10
127 157 98 119 241 37 215 149 217 58 79 203 137 215 147 44 193 81 116
122 90
236 7 10 125 242 233 121 154 245 125 238 145 133 123 71 25 246 65 24
50 159 19
245 127 153 45 131 210 121 150 157 102 207 229 33 166 214 174 138 188
53 49 221
148 240 64 86 144 154 249 178 226 176 49 223 63 105 97 173 194 21 160
250 151
132 181 77 32 187 82 58 81 118 37 213 234 94 60 250 153 47 46 110 119
237 207

227 118 43 216 87 88 63 107 12 38 192 108 210 117 62 169 106 33 164 76
51 211 ^
540 0 101 1 180 50 70 51 143 210 32 181 163 191 81 31 245 123 63 16
135 30 17
154 75 26 93 92 62 163 14 43 129 119 244 144 240 148 70 134 105 91 166
42 27
155 94 28 102 31 79 199 84 225 110 128 89 85 255 9 17 60 259 5 187 74
224 91 83
80 146 238 233 207 138 18 39 39 133 209 230 179 154 252 64 255 175 158
82 50
247 89 2 33 128 100 199 135 41 131 34 107 140 67 200 197 86 140 16 124
176 137
122 178 52 245 174 139 37 41 48 72 226 6 206 119 233 30 100 116 173
160 214 173
26 176 54 160 80 135 124 93 175 24 44 191 164 210 83 257 184 140 99 86
57 54
243 42 16 41 214 256 205 57 188 21 161 99 180 133 32 192 17 71 113 145
219 231
198 199 1 3 12 211 228 4 31 213 237 203 48 249 4 125 72 215 152 66 87
76 56 252
241 33 77 24 178 229 23 76 222 176 220 84 59 186 106 38 103 93 81 165
23 242
146 155 11 66 4 114 175 75 120 169 113 228 181 69 115 60 82 227 25 251
85 172
47 136 186 138 184 223 61 184 191 175 107 6 29 81 114 207 221 197 137
216 254
73 194 192 194 109 232 11 98 107 234 143 221 146 61 195 88 55 139 120
3 189 249
87 87 170 132 13 147 51 132 96 109 149 49 217 233 113 62 257 90 64 78
137 39
216 171 111 53 58 73 28 8 215 58 250 66 170 38 197 169 102 114 41 37
218 86 234
92 145 125 155 177 250 243 208 200 20 48 21 150 245 40 101 178 218 169
196 190
19 246 236 141 46 23 65 108 36 188 187 85 78 231 115 237 120 86 151
130 4 208
34 96 26 187 168 40 112 32 98 201 50 164 127 121 159 7 19 235 122 1 14
126 91
118 5 104 112 126 174 222 259 182 225 193 90 230 2 116 90 198 116 39
122 95 90
147 40 18 216 77 35 75 109 55 190 196 24 95 7 185 159 101 95 256 71 19
69 198
22 180 144 146 72 49 228 87 253 94 111 64 172 213 60 165 189 166 125
206 36 11
149 183 255 92 196 201 133 126 8 38 20 131 200 31 162 161 182 142 14
209 53 141
35 126 185 76 139 214 79 167 198 105 142 242 63 193 173 192 100 33 211
62 80 52
162 212 258 120 180 227 108 213 226 89 168 217 150 151 181 80 229 200
114 124
259 99 3 95 173 109 138 152 77 201 129 ^

555 0 84 149 215 139 210 158 38 153 14 133 109 152 156 248 232 9 183
240 236 50
167 17 148 158 163 93 3 8 32 103 8 200 204 108 32 9 58 89 247 175 244
66 85 155
191 256 154 239 210 115 139 42 108 200 110 209 132 103 176 39 42 151
99 10 115
65 102 119 63 187 258 69 100 204 202 193 171 15 233 35 118 6 211 141
125 70 83
29 68 67 185 81 157 106 117 148 64 2 193 46 75 33 160 246 55 97 138
153 182 183
260 195 223 198 27 247 38 184 55 222 78 142 57 106 211 47 226 170 126
96 18 6
43 40 25 238 227 28 11 235 212 30 36 195 254 239 116 165 28 85 175 150
167 185
249 258 206 160 215 65 8 220 163 19 63 19 157 180 57 137 201 136 144
160 78 216
259 95 86 232 177 22 129 68 168 117 74 124 13 56 174 124 181 157 12
218 248 158
69 194 103 51 99 53 14 176 59 52 62 130 20 214 156 123 30 20 46 106 23
249 164
45 49 47 132 9 15 139 136 7 100 16 91 182 46 149 184 98 121 240 48 252
156 154
71 109 226 96 186 107 143 134 92 188 22 4 128 179 242 151 173 161 104
202 25 70
126 190 179 168 211 235 24 181 251 173 67 205 40 68 137 127 196 206 23
155 3
145 92 114 82 66 253 256 29 37 221 189 5 91 256 197 138 59 21 72 41 51
224 255
3 102 193 3 176 133 203 51 56 154 165 8 126 2 119 157 86 107 237 33
203 145 217
54 71 203 125 207 143 40 193 77 116 122 78 236 7 6 117 242 225 113 150
241 125
238 133 129 111 67 17 242 57 12 50 147 7 237 127 153 45 123 198 121
146 149 90
199 221 21 166 202 162 130 188 53 45 217 148 232 52 82 140 142 245 166
222 172
41 219 63 93 97 169 194 9 152 250 147 132 177 65 28 179 74 50 73 118
37 201 230
94 60 246 149 47 38 110 115 233 203 219 106 43 208 75 76 59 95 12 30
188 96 206
117 54 165 102 25 164 76 39 136 101 261 178 48 64 45 143 208 32 177
159 189 79
31 245 123 61 10 135 24 13 150 73 24 87 90 62 161 10 41 125 113 244
140 236 144
66 128 105 91 162 36 27 153 88 22 98 27 79 199 84 223 104 128 85 81
251 5 17 54
259 1 187 70 220 89 79 74 144 234 229 205 134 18 37 33 129 205 228 179
148 252
62 255 171 152 82 46 243 83 260 27 122 98 195 129 37 127 28 105 134
207 ^
545 1 196 193 86 136 16 120 168 133 114 170 48 237 162 135 29 29 40 60
218 2

198 119 229 22 92 108 161 148 202 165 26 164 50 152 80 127 116 85 171
24 44 191
160 198 71 249 180 140 99 82 53 54 235 42 12 37 206 248 193 49 176 9
157 91 180
129 20 188 5 63 109 137 219 231 190 195 1 259 8 207 224 19 201 237 199
36 241
256 117 72 203 152 58 87 64 48 248 241 29 77 20 166 229 11 72 214 172
216 76 55
174 94 34 99 93 73 153 19 238 138 147 263 54 4 102 163 63 120 157 113
220 181
57 111 52 78 223 13 251 73 164 39 132 186 138 184 223 61 172 179 175
107 6 17
81 114 207 213 185 125 212 246 73 190 184 186 101 224 11 98 95 226 131
221 146
49 187 88 51 139 112 255 189 245 75 79 162 124 9 135 51 128 92 97 141
49 213
233 105 54 257 82 64 74 129 31 208 163 111 41 58 65 24 260 211 54 246
62 170 26
197 169 98 106 41 25 210 78 234 92 145 121 155 165 242 243 208 200 12
48 21 142
241 40 97 178 214 161 196 182 7 242 232 129 42 23 57 100 32 184 175 81
66 227
107 233 116 74 151 130 260 200 34 88 14 179 164 28 112 28 90 193 38
156 115 109
159 259 19 227 118 253 10 122 83 256 106 257 104 104 126 166 218 255
178 225
181 90 230 262 104 78 186 112 39 110 87 86 147 36 14 216 65 35 75 105
43 178
188 16 83 3 185 151 93 95 252 71 11 61 194 14 168 144 134 64 37 228 83
245 86
99 56 172 205 56 161 185 162 113 194 36 3 137 171 251 84 184 197 121
118 30 16
131 188 27 162 161 174 142 14 205 45 141 27 114 181 68 131 210 67 155
198 97
130 234 55 185 173 192 88 25 199 58 76 44 154 200 250 108 172 227 96
213 222 85
160 209 150 143 169 72 225 192 110 124 251 95 263 91 169 109 126 140
73 201 204
80 149 215 137 208 152 32 147 10 133 103 150 154 248 230 9 183 236 234
44 165
15 144 156 163 89 1 6 28 101 8 196 204 106 30 5 52 89 243 171 240 64
85 149 189
256 154 237 210 115 135 40 108 198 108 209 128 103 176 35 38 145 95 10
111 63
98 117 61 183 258 69 96 202 202 191 171 13 229 33 112 2 209 139 123 70
83 29 66
67 181 79 151 104 115 146 60 2 187 40 71 33 160 246 51 91 132 178 ^
563 1 174 171 260 183 215 190 23 247 38 180 55 222 70 138 45 98 211 39
226 166
118 92 6 260 39 32 25 234 219 20 3 227 212 30 254 28 195 254 239 112
153 28 73

171 138 167 181 245 250 198 148 207 65 262 220 155 15 55 11 153 172 57
137 193
128 132 156 78 208 255 91 82 232 177 22 129 68 168 105 74 112 9 44 174
116 177
149 4 218 240 158 65 190 95 39 87 41 2 164 59 52 50 118 20 214 144 111
18 16 46
106 15 249 152 37 45 43 128 5 15 127 136 3 100 4 79 182 42 137 176 86
117 240
36 248 156 150 67 105 218 96 174 99 135 134 88 176 14 258 124 175 234
147 161
153 100 198 21 66 114 190 167 164 203 235 16 173 247 165 55 205 28 56
129 123
184 206 19 151 257 133 80 106 70 66 253 248 17 25 217 177 5 87 252 185
130 47
13 68 29 47 224 251 261 102 185 3 172 129 195 43 56 146 165 122 260
111 157 74
95 233 29 191 141 217 50 63 203 113 199 139 36 193 73 116 122 66 236 7
2 109
242 217 105 146 237 125 238 121 125 99 63 9 238 49 50 135 261 229 127
153 45
115 186 121 142 141 78 191 213 9 166 190 150 122 188 53 41 213 148 224
40 78
136 130 241 154 218 168 33 215 63 81 97 165 194 263 144 250 143 132
173 53 24
171 66 42 65 118 37 189 226 94 60 242 145 47 30 110 111 229 199 211 94
43 200
63 64 55 83 12 22 184 84 202 117 46 161 98 17 164 76 27 132 101 261
174 44 52
33 143 204 32 169 151 185 75 31 245 123 57 264 135 12 5 142 69 20 75
86 62 157
2 37 117 101 244 132 228 136 58 116 105 91 154 24 27 149 76 10 90 19
79 199 84
219 92 128 77 73 243 263 17 42 259 259 187 62 212 85 71 62 140 226 221
201 126
18 33 21 121 197 224 179 136 252 58 255 163 140 82 38 235 71 256 15
110 94 187
117 29 119 16 101 122 49 194 191 86 134 16 118 164 131 110 166 46 233
156 133
25 23 36 54 214 194 119 227 18 88 104 155 142 196 161 26 158 48 148 80
123 112
81 169 24 44 191 158 192 65 245 178 140 99 80 51 54 231 42 10 35 202
244 187 45
170 3 155 87 180 127 14 186 265 59 107 133 219 231 186 193 1 257 6 205
222 264
13 195 237 197 30 237 252 113 72 197 152 54 87 58 44 246 241 27 77 18
160 229 5
70 210 170 214 72 53 168 88 32 97 93 69 261 ^
563 1 15 234 130 139 255 42 4 90 151 51 120 145 113 212 181 45 107 44
74 219 1
251 61 156 31 128 186 138 184 223 61 160 167 175 107 6 5 81 114 207
205 173 113

208 238 73 186 176 178 93 216 11 98 83 218 119 221 146 37 179 88 47
139 104 247
189 241 63 71 154 116 5 123 51 124 88 85 133 49 209 233 97 46 257 74
64 70 121
23 200 155 111 29 58 57 20 252 207 50 242 58 170 14 197 169 94 98 41
13 202 70
234 92 145 117 155 153 234 243 208 200 4 48 21 134 237 40 93 178 210
153 196
174 263 238 228 117 38 23 49 92 28 180 163 77 54 223 99 229 112 62 151
130 256
192 34 80 2 171 160 16 112 24 82 185 26 148 103 97 159 251 19 219 114
245 6 118
75 252 94 249 104 96 126 158 214 251 174 225 169 90 230 262 92 66 174
108 39 98
79 82 147 32 10 216 53 35 75 101 31 166 180 8 71 267 185 143 85 95 248
71 3 53
190 6 156 144 122 56 25 228 79 237 78 87 48 172 197 52 157 181 158 101
182 36
263 125 159 247 76 172 193 109 110 260 22 12 131 176 23 162 161 166
142 14 201
37 141 19 102 177 60 123 206 55 143 198 89 118 226 47 177 173 192 76
17 187 54
72 36 146 188 242 96 164 227 84 213 218 81 152 201 150 135 157 64 221
184 106
124 243 91 263 87 165 109 114 128 69 201 192 72 149 215 133 204 140 20
135 2
133 91 146 150 248 226 9 183 228 230 32 161 11 136 152 163 81 265 2 20
97 8 188
204 102 26 265 40 89 235 163 232 60 85 137 185 256 154 233 210 115 127
36 108
194 104 209 120 103 176 27 30 133 87 10 103 59 90 113 57 175 258 69 88
198 202
187 171 9 221 29 100 262 205 135 119 70 83 29 62 67 173 75 139 100 111
142 52 2
175 28 63 33 160 246 43 79 120 141 170 165 260 177 211 186 21 247 38
178 55 222
66 136 39 94 211 35 226 164 114 90 256 37 28 25 232 215 16 267 223 212
30 250
24 195 254 239 110 147 28 67 169 132 167 179 243 246 194 142 203 65
258 220 151
13 51 7 151 168 57 137 189 124 126 154 78 204 253 89 80 232 177 22 129
68 168
99 74 106 7 38 174 112 175 145 218 236 158 63 188 91 33 81 35 264 158
59 52 44
112 20 214 138 105 12 14 46 106 11 249 146 33 43 41 126 3 15 121 136 1
100 266
73 182 40 131 172 80 115 240 30 246 186 ^
573 0 146 63 101 210 96 162 91 127 134 84 164 6 250 120 171 226 143
149 145 96
194 17 62 102 190 155 160 195 235 8 165 243 157 43 205 16 44 121 119
172 206 15

147 249 121 68 98 58 66 253 240 5 13 213 165 5 83 248 173 122 35 5 64
17 43 224
247 257 102 177 3 168 125 187 35 56 138 165 262 118 256 103 157 62 83
229 25
179 137 217 46 55 203 101 191 135 32 193 69 116 122 54 236 7 268 101
242 209 97
142 233 125 238 109 121 87 59 1 234 41 258 50 123 253 221 127 153 45
107 174
121 138 133 66 183 205 267 166 178 138 114 188 53 37 209 148 216 28 74
132 118
237 142 214 164 25 211 63 69 97 161 194 255 136 250 139 132 169 41 20
163 58 34
57 118 37 177 222 94 60 238 141 47 22 110 107 225 195 203 82 43 192 51
52 51 71
12 14 180 72 198 117 38 157 94 9 164 76 15 128 101 261 170 40 40 21
143 200 32
161 143 181 71 31 245 123 53 256 135 267 134 65 16 63 82 62 153 264 33
109 89
244 124 220 128 50 104 105 91 146 12 27 145 64 268 82 11 79 199 84 215
80 128
69 65 235 259 17 30 259 255 187 54 204 81 63 50 136 218 213 197 118 18
29 9 113
189 220 179 124 252 54 255 155 128 82 30 227 59 252 3 98 90 179 105 21
111 4 97
110 37 190 187 86 130 16 114 156 127 102 158 42 225 144 129 17 11 28
42 206 266
186 119 223 10 80 96 143 130 184 153 26 146 44 140 80 115 104 73 165
24 44 191
154 180 53 237 174 140 99 76 47 54 223 42 6 31 194 236 175 37 158 261
151 79
180 123 2 182 257 51 103 125 219 231 178 189 1 253 2 201 218 264 1 183
237 193
18 229 244 105 72 185 152 46 87 46 36 242 241 23 77 14 148 229 263 66
202 166
210 64 49 156 76 28 93 93 61 135 13 232 126 135 251 36 4 84 145 45 120
139 113
208 181 39 105 40 72 217 265 251 55 152 27 126 186 138 184 223 61 154
161 175
107 6 269 81 114 207 201 167 107 206 234 73 184 172 174 89 212 11 98
77 214 113
221 146 31 175 88 45 139 100 243 189 239 57 67 150 112 3 117 51 122 86
79 129
49 207 233 93 42 257 70 64 68 117 19 196 151 111 23 58 53 18 248 205
48 240 56
170 8 197 169 92 94 41 7 198 66 234 92 145 115 155 147 230 243 208 200
48 21
130 235 40 91 178 208 149 196 170 259 236 226 111 36 23 45 88 26 178
157 207 ^
566 1 42 219 91 225 108 50 151 130 252 184 34 72 262 163 156 4 112 20
74 177 14
140 91 85 159 243 19 211 110 237 2 114 67 248 82 241 104 88 126 150
210 247 170

225 157 90 230 262 80 54 162 104 39 86 71 78 147 28 6 216 41 35 75 97
19 154
172 59 267 185 135 77 95 244 71 267 45 186 270 144 144 110 48 13 228
75 229 70
75 40 172 189 48 153 177 154 89 170 36 259 113 147 243 68 160 189 97
102 256 14
8 131 164 19 162 161 158 142 14 197 29 141 11 90 173 52 115 202 43 131
198 81
106 218 39 169 173 192 64 9 175 50 68 28 138 176 234 84 156 227 72 213
214 77
144 193 150 127 145 56 217 176 102 124 235 87 263 83 161 109 102 116
65 201 180
64 149 215 129 200 128 8 123 266 133 79 142 146 248 222 9 183 220 226
20 157 7
128 148 163 73 265 270 12 93 8 180 204 98 22 261 28 89 227 155 224 56
85 125
181 256 154 229 210 115 119 32 108 190 100 209 112 103 176 19 22 121
79 10 95
55 82 109 53 167 258 69 80 194 202 183 171 5 213 25 88 258 201 131 115
70 83 29
58 67 165 71 127 96 107 138 44 2 163 16 55 33 160 246 35 67 108 133
162 153 260
165 203 178 17 247 38 174 55 222 58 132 27 86 211 27 226 160 106 86
260 248 33
20 25 228 207 8 263 215 212 30 242 16 195 254 239 106 135 28 55 165
120 167 175
239 238 186 130 195 65 250 220 143 9 43 271 147 160 57 137 181 116 114
150 78
196 249 85 76 232 177 22 129 68 168 87 74 94 3 26 174 104 171 137 264
218 228
158 59 184 83 21 69 23 256 146 59 52 32 100 20 214 126 93 10 46 106 3
249 134
25 39 37 122 271 15 109 136 269 100 258 61 182 36 119 164 68 111 240
18 242 156
144 61 99 206 96 156 87 123 134 82 158 2 246 118 169 222 141 143 141
94 192 15
60 96 190 149 158 191 235 4 161 241 153 37 205 10 38 117 117 166 206
13 145 245
115 62 94 52 66 253 236 271 7 211 159 5 81 246 167 118 29 1 62 11 41
224 245
255 102 173 3 166 123 183 31 56 134 165 260 116 254 99 157 56 77 227
23 173 135
217 44 51 203 95 187 133 30 193 67 116 122 48 236 7 268 97 242 205 93
140 231
125 238 103 119 81 57 269 232 37 254 50 117 249 217 127 153 45 103 168
121 136
129 60 179 201 263 166 172 132 110 188 53 35 207 148 212 262 ^
589 0 70 128 106 233 130 210 160 17 207 63 57 97 157 194 247 128 250
135 132
165 29 16 155 50 26 49 118 37 165 218 94 60 234 137 47 14 110 103 221
191 195

70 43 184 39 40 47 59 12 6 176 60 194 117 30 153 90 1 164 76 3 124 101
261 166
36 28 9 143 196 32 153 135 177 67 31 245 123 49 248 135 262 263 126 61
12 51 78
62 149 260 29 101 77 244 116 212 120 42 92 105 91 138 27 141 52 260 74
3 79 199
84 211 68 128 61 57 227 255 17 18 259 251 187 46 196 77 55 38 132 210
205 193
110 18 25 271 105 181 216 179 112 252 50 255 147 116 82 22 219 47 248
265 86 86
171 93 13 103 266 93 98 25 186 183 86 126 16 110 148 123 94 150 38 217
132 125
9 273 20 30 198 266 178 119 219 2 72 88 131 118 172 145 26 134 40 132
80 107 96
65 161 24 44 191 150 168 41 229 170 140 99 72 43 54 215 42 2 27 186
228 163 29
146 253 147 71 180 119 264 178 249 43 99 117 219 231 170 185 1 249 272
197 214
264 263 171 237 189 6 221 236 97 72 173 152 38 87 34 28 238 241 19 77
10 136
229 255 62 194 162 206 56 45 144 64 24 89 93 53 123 9 228 118 127 243
24 4 72
133 33 120 127 113 200 181 27 101 32 68 213 257 251 43 144 19 122 186
138 184
223 61 142 149 175 107 6 261 81 114 207 193 155 95 202 226 73 180 164
166 81
204 11 98 65 206 101 221 146 19 167 88 41 139 92 235 189 235 45 59 142
104 273
105 51 118 82 67 121 49 203 233 85 34 257 62 64 64 109 11 188 143 111
11 58 45
14 240 201 44 236 52 170 270 197 169 88 86 41 269 190 58 234 92 145
111 155 135
222 243 208 200 266 48 21 122 231 40 87 178 204 141 196 162 251 232
222 99 32
23 37 80 22 174 145 71 36 217 87 223 106 44 151 130 250 180 34 68 258
159 154
272 112 18 70 173 8 136 85 79 159 239 19 207 108 233 112 63 246 76 237
104 84
126 146 208 245 168 225 151 90 230 262 74 48 156 102 39 80 67 76 147
26 4 216
35 35 75 95 13 148 168 270 53 267 185 131 73 95 242 71 265 41 184 268
138 144
104 44 7 228 73 225 66 69 36 172 185 46 151 175 152 83 164 36 257 107
141 241
64 154 187 91 98 254 10 6 131 158 17 162 161 154 142 14 195 25 141 7
84 171 48
111 200 37 125 198 77 100 214 35 165 173 192 58 5 169 48 66 24 134 170
230 78
152 227 66 213 212 75 140 189 150 123 139 52 215 172 124 ^
584 0 124 227 83 263 79 157 109 90 104 61 201 168 56 149 215 125 196
116 272

111 262 133 67 138 142 248 218 9 183 212 222 8 153 3 120 144 163 65
265 270 4
89 8 172 204 94 18 257 16 89 219 147 216 52 85 113 177 256 154 225 210
115 111
28 108 186 96 209 104 103 176 11 14 109 71 10 87 51 74 105 49 159 258
69 72 190
202 179 171 1 205 21 76 254 197 127 111 70 83 29 54 67 157 67 115 92
103 134 36
2 151 4 47 33 160 246 27 55 96 125 154 141 260 153 195 170 13 247 38
170 55 222
50 128 15 78 211 19 226 156 98 82 252 240 29 12 25 224 199 259 207 212
30 234 8
195 254 239 102 123 28 43 161 108 167 171 235 230 178 118 187 65 242
220 135 5
35 267 143 152 57 137 173 108 102 146 78 188 245 81 72 232 177 22 129
68 168 75
74 82 275 14 174 96 167 129 260 218 220 158 55 180 75 9 57 11 248 134
59 52 20
88 20 214 114 81 264 6 46 106 271 249 122 17 35 33 118 271 15 97 136
269 100
250 49 182 32 107 156 56 107 240 6 238 156 140 57 95 198 96 144 79 115
134 78
146 270 238 114 165 214 137 131 133 90 188 11 56 84 190 137 154 183
235 272 153
237 145 25 205 274 26 109 113 154 206 9 141 237 103 50 86 40 66 253
228 263 271
207 147 5 77 242 155 110 17 269 58 275 37 224 241 251 102 165 3 162
119 175 23
56 126 165 256 112 250 91 157 44 65 223 19 161 131 217 40 43 203 83
179 129 26
193 63 116 122 36 236 7 268 89 242 197 85 136 227 125 238 91 115 69 53
265 228
29 246 50 105 241 209 127 153 45 95 156 121 132 121 48 171 193 255 166
160 120
102 188 53 31 203 148 204 10 68 126 100 231 124 208 158 13 205 63 51
97 155 194
243 124 250 133 132 163 23 14 151 46 22 45 118 37 159 216 94 60 232
135 47 10
110 101 219 189 191 64 43 180 33 34 45 53 12 2 174 54 192 117 26 151
88 273 164
76 273 122 101 261 164 34 22 3 143 194 32 149 131 175 65 31 245 123 47
244 135
258 261 122 59 10 45 76 62 147 258 27 97 71 244 112 208 116 38 86 105
91 134
270 27 139 46 256 70 275 79 199 84 209 62 128 57 53 223 253 17 12 259
249 187
42 192 75 51 32 130 206 201 191 106 18 23 267 101 177 214 179 106 252
48 255
143 110 82 18 215 41 246 261 80 84 167 87 9 99 262 91 92 19 184 181 86
124 16
108 144 121 90 146 36 213 126 123 5 269 230 ^

587 0 18 190 266 170 119 215 272 64 80 119 106 160 137 26 122 36 124
80 99 88
57 157 24 44 191 146 156 29 221 166 140 99 68 39 54 207 42 276 23 178
220 151
21 134 245 143 63 180 115 256 174 241 35 95 109 219 231 162 181 1 245
272 193
210 264 255 159 237 185 272 213 228 89 72 161 152 30 87 22 20 234 241
15 77 6
124 229 247 58 186 158 202 48 41 132 52 20 85 93 45 111 5 224 110 119
235 12 4
60 121 21 120 115 113 192 181 15 97 24 64 209 249 251 31 136 11 118
186 138 184
223 61 130 137 175 107 6 253 81 114 207 185 143 83 198 218 73 176 156
158 73
196 11 98 53 198 89 221 146 7 159 88 37 139 84 227 189 231 33 51 134
96 273 93
51 114 78 55 113 49 199 233 77 26 257 54 64 60 101 3 180 135 111 277
58 37 10
232 197 40 232 48 170 262 197 169 84 78 41 261 182 50 234 92 145 107
155 123
214 243 208 200 262 48 21 114 227 40 83 178 200 133 196 154 243 228
218 87 28
23 29 72 18 170 133 67 24 213 79 219 102 32 151 130 246 172 34 60 250
151 150
264 112 14 62 165 274 128 73 67 159 231 19 199 104 225 274 108 55 242
64 229
104 76 126 138 204 241 164 225 139 90 230 262 62 36 144 98 39 68 59 72
147 22
216 23 35 75 91 1 136 160 266 41 267 185 123 65 95 238 71 261 33 180
264 126
144 92 36 273 228 69 217 58 57 28 172 177 42 147 171 148 71 152 36 253
95 129
237 56 142 183 79 90 250 2 2 131 146 13 162 161 146 142 14 191 17 141
277 72
167 40 103 196 25 113 198 69 88 206 27 157 173 192 46 275 157 44 62 16
126 158
222 66 144 227 54 213 208 71 132 181 150 115 127 44 211 164 96 124 223
81 263
77 155 109 84 98 59 201 162 52 149 215 123 194 110 268 105 260 133 61
136 140
248 216 9 183 208 220 2 151 1 116 142 163 61 265 270 87 8 168 204 92
16 255 10
89 215 143 212 50 85 107 175 256 154 223 210 115 107 26 108 184 94 209
100 103
176 7 10 103 67 10 83 49 70 103 47 155 258 69 68 188 202 177 171 277
201 19 70
252 195 125 109 70 83 29 52 67 153 65 109 90 101 132 32 2 145 276 43
33 160 246
23 49 90 121 150 135 260 147 191 166 11 247 38 168 55 222 46 126 9 74
211 15
226 154 94 80 248 236 27 8 25 222 195 274 257 203 212 30 230 4 195 254
239 100

117 28 37 159 102 167 169 233 226 174 112 183 65 238 220 131 3 31 270
^
609 1 140 146 57 137 167 102 93 143 78 182 242 78 69 232 177 22 129 68
168 66
74 73 275 5 174 90 164 123 257 218 214 158 52 177 69 48 2 242 125 59
52 11 79
20 214 105 72 258 3 46 106 268 249 113 11 32 30 115 271 15 88 136 269
100 244
40 182 29 98 150 47 104 240 276 235 156 137 54 92 192 96 135 73 109
134 75 137
267 232 111 162 208 134 122 127 87 185 8 53 75 190 128 151 177 235 269
147 234
139 16 205 268 17 103 110 145 206 6 138 231 94 41 80 31 66 253 222 257
265 204
138 5 74 239 146 104 8 266 55 269 34 224 238 248 102 159 3 159 116 169
17 56
120 165 253 109 247 85 157 35 56 220 16 152 128 217 37 37 203 74 173
126 23 193
60 116 122 27 236 7 268 83 242 191 79 133 224 125 238 82 112 60 50 262
225 23
240 50 96 235 203 127 153 45 89 147 121 129 115 39 165 187 249 166 151
111 96
188 53 28 200 148 198 1 65 123 91 228 115 205 155 7 202 63 42 97 152
194 237
118 250 130 132 160 14 11 145 40 16 39 118 37 150 213 94 60 229 132 47
4 110 98
216 186 185 55 43 174 24 25 42 44 12 275 171 45 189 117 20 148 85 270
164 76
267 119 101 261 161 31 13 273 143 191 32 143 125 172 62 31 245 123 44
238 135
252 258 116 56 7 36 73 62 144 255 24 91 62 244 106 202 110 32 77 105
91 128 264
27 136 37 250 64 272 79 199 84 206 53 128 51 47 217 250 17 3 259 246
187 36 186
72 45 23 127 200 195 188 100 18 20 261 95 171 211 179 97 252 45 255
137 101 82
12 209 32 243 255 71 81 161 78 3 93 256 88 83 10 181 178 86 121 16 105
138 118
84 140 33 207 117 120 278 263 10 15 188 266 168 119 214 271 62 78 116
103 157
135 26 119 35 122 80 97 86 55 156 24 44 191 145 153 26 219 165 140 99
67 38 54
205 42 276 22 176 218 148 19 131 243 142 61 180 114 254 173 239 33 94
107 219
231 160 180 1 244 272 192 209 264 253 156 237 184 270 211 226 87 72
158 152 28
87 19 18 233 241 14 77 5 121 229 245 57 184 157 201 46 40 129 49 19 84
93 43
108 4 223 108 117 233 9 4 57 118 18 120 112 113 190 181 12 96 22 63
208 247 251
28 134 9 117 186 138 184 223 61 127 134 175 107 6 251 81 114 207 183
140 80 197

216 73 175 154 156 71 194 11 98 50 196 86 221 146 4 157 88 36 139 82
225 189
230 30 49 132 94 273 90 51 113 77 52 111 49 198 233 75 24 257 52 64 59
99 1 178
133 111 275 58 35 9 230 196 39 231 47 170 216 ^
589 0 197 169 81 72 41 255 176 44 234 92 145 104 155 114 208 243 208
200 259 48
21 108 224 40 80 178 197 127 196 148 237 225 215 78 25 23 23 66 15 167
124 64
15 210 73 216 99 23 151 130 243 166 34 54 244 145 147 258 112 11 56
159 268 122
64 58 159 225 19 193 101 219 274 105 49 239 55 223 104 70 126 132 201
238 161
225 130 90 230 262 53 27 135 95 39 59 53 69 147 19 278 216 14 35 75 88
273 127
154 263 32 267 185 117 59 95 235 71 258 27 177 261 117 144 83 30 267
228 66 211
52 48 22 172 171 39 144 168 145 62 143 36 250 86 120 234 50 133 180 70
84 247
277 280 131 137 10 162 161 140 142 14 188 11 141 274 63 164 34 97 193
16 104
198 63 79 200 21 151 173 192 37 272 148 41 59 10 120 149 216 57 138
227 45 213
205 68 126 175 150 109 118 38 208 158 93 124 217 78 263 74 152 109 75
89 56 201
153 46 149 215 120 191 101 262 96 257 133 52 133 137 248 213 9 183 202
217 274
148 279 110 139 163 55 265 270 275 84 8 162 204 89 13 252 1 89 209 137
206 47
85 98 172 256 154 220 210 115 101 23 108 181 91 209 94 103 176 1 4 94
61 10 77
46 64 100 44 149 258 69 62 185 202 174 171 277 195 16 61 249 192 122
106 70 83
29 49 67 147 62 100 87 98 129 26 2 136 270 37 33 160 246 17 40 81 115
144 126
260 138 185 160 8 247 38 165 55 222 40 123 68 211 9 226 151 88 77 242
230 24 2
25 219 189 271 254 197 212 30 224 279 195 254 239 97 108 28 28 156 93
167 166
230 220 168 103 177 65 232 220 125 25 262 138 142 57 137 163 98 87 141
78 178
240 76 67 232 177 22 129 68 168 60 74 67 275 280 174 86 162 119 255
218 210 158
50 175 65 275 42 277 238 119 59 52 5 73 20 214 99 66 254 1 46 106 266
249 107 7
30 28 113 271 15 82 136 269 100 240 34 182 27 92 146 41 102 240 272
233 156 135
52 90 188 96 129 69 105 134 73 131 265 228 109 160 204 132 116 123 85
183 6 51
69 190 122 149 173 235 267 143 232 135 10 205 264 11 99 108 139 206 4
136 227

88 35 76 25 66 253 218 253 261 202 132 5 72 237 140 100 2 264 53 265
32 224 236
246 102 155 3 157 114 165 13 56 116 165 251 107 245 81 157 29 50 218
14 146 126
217 35 33 203 68 169 124 21 193 58 116 122 21 236 7 268 79 242 187 75
131 222
125 238 76 110 54 48 260 223 19 236 50 90 231 199 127 153 45 270 ^
623 1 135 121 125 107 27 157 179 241 166 139 99 88 188 53 24 196 148
190 272 61
119 79 224 103 201 151 282 198 63 30 97 148 194 229 110 250 126 132
156 2 7 137
32 8 31 118 37 138 209 94 60 225 128 47 279 110 94 212 182 177 43 43
166 12 13
38 32 12 271 167 33 185 117 12 144 81 266 164 76 259 115 101 261 157
27 1 265
143 187 32 135 117 168 58 31 245 123 40 230 135 244 254 108 52 3 24 69
62 140
251 20 83 50 244 98 194 102 24 65 105 91 120 256 27 132 25 242 56 268
79 199 84
202 41 128 43 39 209 246 17 274 259 242 187 28 178 68 37 11 123 192
187 184 92
18 16 253 87 163 207 179 85 252 41 255 129 89 82 4 201 20 239 247 59
77 153 66
278 85 248 84 71 281 177 174 86 117 16 101 130 114 76 132 29 199 105
116 274
255 2 3 180 266 160 119 210 267 54 70 104 91 145 127 26 107 31 114 80
89 78 47
152 24 44 191 141 141 14 211 161 140 99 63 34 54 197 42 276 18 168 210
136 11
119 235 138 53 180 110 246 169 231 25 90 99 219 231 152 176 1 240 272
188 205
264 245 144 237 180 262 203 218 79 72 146 152 20 87 7 10 229 241 10 77
1 109
229 237 53 176 153 197 38 36 117 37 15 80 93 35 96 219 100 109 225 280
4 45 106
6 120 100 113 182 181 92 14 59 204 239 251 16 126 1 113 186 138 184
223 61 115
122 175 107 6 243 81 114 207 175 128 68 193 208 73 171 146 148 63 186
11 98 38
188 74 221 146 275 149 88 32 139 74 217 189 226 18 41 124 86 273 78 51
109 73
40 103 49 194 233 67 16 257 44 64 55 91 276 170 125 111 267 58 27 5
222 192 35
227 43 170 252 197 169 79 68 41 251 172 40 234 92 145 102 155 108 204
243 208
200 257 48 21 104 222 40 78 178 195 123 196 144 233 223 213 72 23 23
19 62 13
165 118 62 9 208 69 214 97 17 151 130 241 162 34 50 240 141 145 254
112 9 52
155 264 118 58 52 159 221 19 189 99 215 274 103 45 237 49 219 104 66
126 128

199 236 159 225 124 90 230 262 47 21 129 93 39 53 49 67 147 17 278 216
8 35 75
86 269 121 150 261 26 267 185 113 55 95 233 71 256 23 175 259 111 144
77 26 263
228 64 207 48 42 18 172 167 37 142 166 143 56 137 36 248 80 114 232 46
127 178
64 80 245 275 280 131 131 8 162 161 136 142 14 186 7 141 272 57 162 30
93 191
10 98 198 59 73 196 17 147 173 192 31 270 142 39 57 6 116 143 212 51
134 227 39
213 203 66 122 171 150 105 112 34 206 154 91 124 213 76 263 72 150 109
69 83 54
201 147 42 149 202 ^
602 1 116 187 89 254 84 253 133 40 129 133 248 209 9 183 194 213 266
144 279
102 135 163 47 265 270 271 80 8 154 204 85 9 248 274 89 201 129 198 43
85 86
168 256 154 216 210 115 93 19 108 177 87 209 86 103 176 278 281 82 53
10 69 42
56 96 40 141 258 69 54 181 202 170 171 277 187 12 49 245 188 118 102
70 83 29
45 67 139 58 88 83 94 125 18 2 124 262 29 33 160 246 9 28 69 107 136
114 260
126 177 152 4 247 38 161 55 222 32 119 273 60 211 1 226 147 80 73 234
222 20
279 25 215 181 267 250 189 212 30 216 275 195 254 239 93 96 28 16 152
81 167
162 226 212 160 91 169 65 224 220 117 281 17 258 134 134 57 137 155 90
75 137
78 170 236 72 63 232 177 22 129 68 168 48 74 55 275 272 174 78 158 111
251 218
202 158 46 171 57 267 30 269 230 107 59 52 278 61 20 214 87 54 246 282
46 106
262 249 95 284 26 24 109 271 15 70 136 269 100 232 22 182 23 80 138 29
98 240
264 229 156 131 48 86 180 96 117 61 97 134 69 119 261 220 105 156 196
128 104
115 81 179 2 47 57 190 110 145 165 235 263 135 228 127 283 205 256 284
91 104
127 206 132 219 76 23 68 13 66 253 210 245 253 198 120 5 68 233 128 92
275 260
49 257 28 224 232 242 102 147 3 153 110 157 5 56 108 165 247 103 241
73 157 17
38 214 10 134 122 217 31 25 203 56 161 120 17 193 54 116 122 9 236 7
268 71 242
179 67 127 218 125 238 64 106 42 44 256 219 11 228 50 78 223 191 127
153 45 77
129 121 123 103 21 153 175 237 166 133 93 84 188 53 22 194 148 186 268
59 117
73 222 97 199 149 280 196 63 24 97 146 194 225 106 250 124 132 154 281
5 133 28

4 27 118 37 132 207 94 60 223 126 47 277 110 92 210 180 173 37 43 162
6 7 36 26
12 269 165 27 183 117 8 142 79 264 164 76 255 113 101 261 155 25 280
261 143
185 32 131 113 166 56 31 245 123 38 226 135 240 252 104 50 1 18 67 62
138 249
18 79 44 244 94 190 98 20 59 105 91 116 252 27 130 19 238 52 266 79
199 84 200
35 128 39 35 205 244 17 270 259 240 187 24 174 66 33 5 121 188 183 182
88 18 14
249 83 159 205 179 79 252 39 255 125 83 82 197 14 237 243 53 75 149 60
276 81
244 82 65 277 175 172 86 115 16 99 126 112 72 128 27 195 99 114 272
251 283 282
176 266 156 119 208 265 50 66 98 85 139 123 26 101 29 110 80 85 74 43
150 24 44
191 139 135 8 207 159 140 99 61 248 ^
609 1 54 189 42 276 14 160 202 124 3 107 227 134 45 180 106 238 165
223 17 86
91 219 231 144 172 1 236 272 184 201 264 237 132 237 176 254 195 210
71 72 134
152 12 87 282 2 225 241 6 77 284 97 229 229 49 168 149 193 30 32 105
25 11 76
93 27 84 283 215 92 101 217 272 4 33 94 281 120 88 113 174 181 275 88
6 55 200
231 251 4 118 280 109 186 138 184 223 61 103 110 175 107 6 235 81 114
207 167
116 56 189 200 73 167 138 140 55 178 11 98 26 180 62 221 146 267 141
88 28 139
66 209 189 222 6 33 116 78 273 66 51 105 69 28 95 49 190 233 59 8 257
36 64 51
83 272 162 117 111 259 58 19 1 214 188 31 223 39 170 244 197 169 75 60
41 243
164 32 234 92 145 98 155 96 196 243 208 200 253 48 21 96 218 40 74 178
191 115
196 136 225 219 209 60 19 23 11 54 9 161 106 58 284 204 61 210 93 5
151 130 237
154 34 42 232 133 141 246 112 5 44 147 256 110 46 40 159 213 19 181 95
207 274
99 37 233 37 211 104 58 126 120 195 232 155 225 112 90 230 262 35 9
117 89 39
41 41 63 147 13 278 216 283 35 75 82 261 109 142 257 14 267 185 105 47
95 229
71 252 15 171 255 99 144 65 18 255 228 60 199 40 30 10 172 159 33 138
162 139
44 125 36 244 68 102 228 38 115 174 52 72 241 271 280 131 119 4 162
161 128 142
14 182 286 141 268 45 158 22 85 187 285 86 198 51 61 188 9 139 173 192
19 266
130 35 53 285 108 131 204 39 126 227 27 213 199 62 114 163 150 97 100
26 202

146 87 124 205 72 263 68 146 109 57 71 50 201 135 34 149 215 114 185
83 250 78
251 133 34 127 131 248 207 9 183 190 211 262 142 279 98 133 163 43 265
270 269
78 8 150 204 83 7 246 270 89 197 125 194 41 85 80 166 256 154 214 210
115 89 17
108 175 85 209 82 103 176 276 279 76 49 10 65 40 52 94 38 137 258 69
50 179 202
168 171 277 183 10 43 243 186 116 100 70 83 29 43 67 135 56 82 81 92
123 14 2
118 258 25 33 160 246 5 22 63 103 132 108 260 120 173 148 2 247 38 159
55 222
28 117 269 56 211 284 226 145 76 71 230 218 18 277 25 213 177 265 248
185 212
30 212 273 195 254 239 91 90 28 10 150 75 167 160 224 208 156 85 165
65 220 220
113 281 13 256 132 130 57 137 151 86 69 135 78 166 234 70 61 232 177
22 129 68
168 42 74 49 275 268 174 74 156 107 249 218 198 158 44 169 53 263 24
265 226
101 59 52 274 55 20 214 81 48 242 282 46 106 260 145 ^
614 1 83 280 22 20 105 271 15 58 136 269 100 224 10 182 19 68 130 17
94 240 256
225 156 127 44 82 172 96 105 53 89 134 65 107 257 212 101 152 188 124
92 107 77
175 287 43 45 190 98 141 157 235 259 127 224 119 275 205 248 276 83
100 115 206
285 128 211 64 11 60 1 66 253 202 237 245 194 108 5 64 229 116 84 267
256 45
249 24 224 228 238 102 139 3 149 106 149 286 56 100 165 243 99 237 65
157 5 26
210 6 122 118 217 27 17 203 44 153 116 13 193 50 116 122 286 236 7 268
63 242
171 59 123 214 125 238 52 102 30 40 252 215 3 220 50 66 215 183 127
153 45 69
117 121 119 95 9 145 167 229 166 121 81 76 188 53 18 190 148 178 260
55 113 61
218 85 195 145 276 192 63 12 97 142 194 217 98 250 120 132 150 273 1
125 20 285
19 118 37 120 203 94 60 219 122 47 273 110 88 206 176 165 25 43 154
283 284 32
14 12 265 161 15 179 117 138 75 260 164 76 247 109 101 261 151 21 272
253 143
181 32 123 105 162 52 31 245 123 34 218 135 232 248 96 46 286 6 63 62
134 245
14 71 32 244 86 182 90 12 47 105 91 108 244 27 126 7 230 44 262 79 199
84 196
23 128 31 27 197 240 17 262 259 236 187 16 166 62 25 282 117 180 175
178 80 18
10 241 75 151 201 179 67 252 35 255 117 71 82 281 189 2 233 235 41 71
141 48

272 73 236 78 53 269 171 168 86 111 16 95 118 108 64 120 23 187 87 110
268 243
279 274 168 266 148 119 204 261 42 58 86 73 127 115 26 89 25 102 80 77
66 35
146 24 44 191 135 123 285 199 155 140 99 57 28 54 185 42 276 12 156
198 118 288
101 223 132 41 180 104 234 163 219 13 84 87 219 231 140 170 1 234 272
182 199
264 233 126 237 174 250 191 206 67 72 128 152 8 87 278 287 223 241 4
77 284 91
229 225 47 164 147 191 26 30 99 19 9 74 93 23 78 283 213 88 97 213 268
4 27 88
277 120 82 113 170 181 271 86 2 53 198 227 251 287 114 278 107 186 138
184 223
61 97 104 175 107 6 231 81 114 207 163 110 50 187 196 73 165 134 136
51 174 11
98 20 176 56 221 146 263 137 88 26 139 62 205 189 220 29 112 74 273 60
51 103
67 22 91 49 188 233 55 4 257 32 64 49 79 270 158 113 111 255 58 15 288
210 186
29 221 37 170 240 197 169 73 56 41 239 160 28 234 92 145 96 155 90 192
243 208
200 251 48 21 92 216 40 72 178 189 111 196 132 221 217 207 54 17 23 7
50 7 159
100 56 280 202 57 208 91 288 151 130 235 150 34 38 228 129 139 242 112
226 ^
619 0 38 141 250 104 37 31 159 207 19 175 92 201 274 96 31 230 28 205
104 52
126 114 192 229 152 225 103 90 230 262 26 108 86 39 32 35 60 147 10
278 216 277
35 75 79 255 100 136 254 5 267 185 99 41 95 226 71 249 9 168 252 90
144 56 12
249 228 57 193 34 21 4 172 153 30 135 159 136 35 116 36 241 59 93 225
32 106
171 43 66 238 268 280 131 110 1 162 161 122 142 14 179 283 141 265 36
155 16 79
184 279 77 198 45 52 182 3 133 173 192 10 263 121 32 50 282 102 122
198 30 120
227 18 213 196 59 108 157 150 91 91 20 199 140 84 124 199 69 263 65
143 109 48
62 47 201 126 28 149 215 111 182 74 244 69 248 133 25 124 128 248 204
9 183 184
208 256 139 279 92 130 163 37 265 270 266 75 8 144 204 80 4 243 264 89
191 119
188 38 85 71 163 256 154 211 210 115 83 14 108 172 82 209 76 103 176
273 276 67
43 10 59 37 46 91 35 131 258 69 44 176 202 165 171 277 177 7 34 240
183 113 97
70 83 29 40 67 129 53 73 78 89 120 8 2 109 252 19 33 160 246 289 13 54
97 126
99 260 111 167 142 289 247 38 156 55 222 22 114 263 50 211 281 226 142
70 68

224 212 15 274 25 210 171 262 245 179 212 30 206 270 195 254 239 88 81
28 1 147
66 167 157 221 202 150 76 159 65 214 220 107 281 7 253 129 124 57 137
145 80 60
132 78 160 231 67 58 232 177 22 129 68 168 33 74 40 275 262 174 68 153
101 246
218 192 158 41 166 47 257 15 259 220 92 59 52 268 46 20 214 72 39 236
282 46
106 257 249 80 279 21 19 104 271 15 55 136 269 100 222 7 182 18 65 128
14 93
240 254 224 156 126 43 81 170 96 102 51 87 134 64 104 256 210 100 151
186 123
89 105 76 174 287 42 42 190 95 140 155 235 258 125 223 117 273 205 246
274 81
99 112 206 285 127 209 61 8 58 288 66 253 200 235 243 193 105 5 63 228
113 82
265 255 44 247 23 224 227 237 102 137 3 148 105 147 285 56 98 165 242
98 236 63
157 2 23 209 5 119 117 217 26 15 203 41 151 115 12 193 49 116 122 284
236 7 268
61 242 169 57 122 213 125 238 49 101 27 39 251 214 1 218 50 63 213 181
127 153
45 67 114 121 118 93 6 143 165 227 166 118 78 74 188 53 17 189 148 176
258 54
112 58 217 82 194 144 275 191 63 9 97 141 194 215 96 250 119 132 149
271 123 18
284 17 118 37 117 202 94 60 218 121 47 272 110 87 205 175 163 22 43
152 281 282
31 11 12 264 160 12 178 117 288 137 74 259 164 76 245 108 101 261 150
20 270
251 143 180 264 ^
639 0 117 99 159 49 31 245 123 31 212 135 226 245 90 43 286 289 60 62
131 242
11 65 23 244 80 176 84 6 38 105 91 102 238 27 123 290 224 38 259 79
199 84 193
14 128 25 21 191 237 17 256 259 233 187 10 160 59 19 276 114 174 169
175 74 18
7 235 69 145 198 179 58 252 32 255 111 62 82 278 183 285 230 229 32 68
135 39
269 67 230 75 44 263 168 165 86 108 16 92 112 105 58 114 20 181 78 107
265 237
276 268 162 266 142 119 201 258 36 52 77 64 118 109 26 80 22 96 80 71
60 29 143
24 44 191 132 114 279 193 152 140 99 54 25 54 179 42 276 9 150 192 109
285 92
217 129 35 180 101 228 160 213 7 81 81 219 231 134 167 1 231 272 179
196 264
227 117 237 171 244 185 200 61 72 119 152 2 87 272 284 220 241 1 77
284 82 229
219 44 158 144 188 20 27 90 10 6 71 93 17 69 283 210 82 91 207 262 4
18 79 271

120 73 113 164 181 265 83 288 50 195 221 251 281 108 275 104 186 138
184 223 61
88 95 175 107 6 225 81 114 207 157 101 41 184 190 73 162 128 130 45
168 11 98
11 170 47 221 146 257 131 88 23 139 56 199 189 217 283 23 106 68 273
51 51 100
64 13 85 49 185 233 49 290 257 26 64 46 73 267 152 107 111 249 58 9
288 204 183
26 218 34 170 234 197 169 70 50 41 233 154 22 234 92 145 93 155 81 186
243 208
200 248 48 21 86 213 40 69 178 186 105 196 126 215 214 204 45 14 23 1
44 4 156
91 53 274 199 51 205 88 282 151 130 232 144 34 32 222 123 136 236 112
34 137
246 100 31 25 159 203 19 171 90 197 274 94 27 228 22 201 104 48 126
110 190 227
150 225 97 90 230 262 20 286 102 84 39 26 31 58 147 8 278 216 273 35
75 77 251
94 132 252 291 267 185 95 37 95 224 71 247 5 166 250 84 144 50 8 245
228 55 189
30 15 172 149 28 133 157 134 29 110 36 239 53 87 223 28 100 169 37 62
236 266
280 131 104 291 162 161 118 142 14 177 281 141 263 30 153 12 75 182
275 71 198
41 46 178 291 129 173 192 4 261 115 30 48 280 98 116 194 24 116 227 12
213 194
57 104 153 150 87 85 16 197 136 82 124 195 67 263 63 141 109 42 56 45
201 120
24 149 215 109 180 68 240 63 246 133 19 122 126 248 202 9 183 180 206
252 137
279 88 128 163 33 265 270 264 73 8 140 204 78 2 241 260 89 187 115 184
36 85 65
161 256 154 209 210 115 79 12 108 170 80 209 72 103 176 271 274 61 39
10 55 35
42 89 33 127 258 69 40 174 202 163 171 277 173 5 28 238 181 111 95 70
83 29 38
67 125 51 67 76 87 118 4 2 103 248 15 33 160 246 287 7 48 93 283 ^
611 0 87 260 99 159 134 289 247 38 152 55 222 14 110 255 42 211 277
226 138 62
64 216 204 11 270 25 206 163 258 241 171 212 30 198 266 195 254 239 84
69 28
283 143 54 167 153 217 194 142 64 151 65 206 220 99 281 293 249 125
116 57 137
137 72 48 128 78 152 227 63 54 232 177 22 129 68 168 21 74 28 275 254
174 60
149 93 242 218 184 158 37 162 39 249 3 251 212 80 59 52 260 34 20 214
60 27 228
282 46 106 253 249 68 275 17 15 100 271 15 43 136 269 100 214 289 182
14 53 120
2 89 240 246 220 156 122 39 77 162 96 90 43 79 134 60 92 252 202 96
147 178 119

77 97 72 170 287 38 30 190 83 136 147 235 254 117 219 109 265 205 238
266 73 95
100 206 285 123 201 49 290 50 280 66 253 192 227 235 189 93 5 59 224
101 74 257
251 40 239 19 224 223 233 102 129 3 144 101 139 281 56 90 165 238 94
232 55 157
284 11 205 1 107 113 217 22 7 203 29 143 111 8 193 45 116 122 276 236
7 268 53
242 161 49 118 209 125 238 37 97 15 35 247 210 287 210 50 51 205 173
127 153 45
59 102 121 114 85 288 135 157 219 166 106 66 66 188 53 13 185 148 168
250 50
108 46 213 70 190 140 271 187 63 291 97 137 194 207 88 250 115 132 145
263 290
115 10 280 9 118 37 105 198 94 60 214 117 47 268 110 83 201 171 155 10
43 144
273 274 27 293 12 260 156 174 117 284 133 70 255 164 76 237 104 101
261 146 16
262 243 143 176 32 113 95 157 47 31 245 123 29 208 135 222 243 86 41
286 285 58
62 129 240 9 61 17 244 76 172 80 2 32 105 91 98 234 27 121 286 220 34
257 79
199 84 191 8 128 21 17 187 235 17 252 259 231 187 6 156 57 15 272 112
170 165
173 70 18 5 231 65 141 196 179 52 252 30 255 107 56 82 276 179 281 228
225 26
66 131 33 267 63 226 73 38 259 166 163 86 106 16 90 108 103 54 110 18
177 72
105 263 233 274 264 158 266 138 119 199 256 32 48 71 58 112 105 26 74
20 92 80
67 56 25 141 24 44 191 130 108 275 189 150 140 99 52 23 54 175 42 276
7 146 188
103 283 86 213 127 31 180 99 224 158 209 3 79 77 219 231 130 165 1 229
272 177
194 264 223 111 237 169 240 181 196 57 72 113 152 292 87 268 282 218
241 293 77
284 76 229 215 42 154 142 186 16 25 84 4 4 69 93 13 63 283 208 78 87
203 258 4
12 73 267 120 67 113 160 181 261 81 286 48 193 217 251 277 104 273 102
186 138
184 223 61 82 89 175 107 6 221 81 114 207 153 95 35 182 186 73 275 ^
634 0 120 122 37 160 11 98 295 162 35 221 146 249 123 88 19 139 48 191
189 213
275 15 98 60 273 39 51 96 60 1 77 49 181 233 41 286 257 18 64 42 65
263 144 99
111 241 58 1 288 196 179 22 214 30 170 226 197 169 66 42 41 225 146 14
234 92
145 89 155 69 178 243 208 200 244 48 21 78 209 40 65 178 182 97 196
118 207 210
200 33 10 23 289 36 152 79 49 266 195 43 201 84 274 151 130 228 136 34
24 214

115 132 228 112 292 26 129 238 92 19 13 159 195 19 163 86 189 274 90
19 224 10
193 104 40 126 102 186 223 146 225 85 90 230 262 8 278 90 80 39 14 23
54 147 4
278 216 265 35 75 73 243 82 124 248 283 267 185 87 29 95 220 71 243
293 162 246
72 144 38 237 228 51 181 22 3 288 172 141 24 129 153 130 17 98 36 235
41 75 219
20 88 165 25 54 232 262 280 131 92 291 162 161 110 142 14 173 277 141
259 18
149 4 67 178 267 59 198 33 34 170 287 121 173 192 288 257 103 26 44
276 90 104
186 12 108 227 213 190 53 96 145 150 79 73 8 193 128 78 124 187 63 263
59 137
109 30 44 41 201 108 16 149 215 105 176 56 232 51 242 133 7 118 122
248 198 9
183 172 202 244 133 279 80 124 163 25 265 270 260 69 8 132 204 74 294
237 252
89 179 107 176 32 85 53 157 256 154 205 210 115 71 8 108 166 76 209 64
103 176
267 270 49 31 10 47 31 34 85 29 119 258 69 32 170 202 159 171 277 165
1 16 234
177 107 91 70 83 29 34 67 117 47 55 72 83 114 292 2 91 240 7 33 160
246 283 291
36 85 114 81 260 93 155 130 289 247 38 150 55 222 10 108 251 38 211
275 226 136
58 62 212 200 9 268 25 204 159 256 239 167 212 30 194 264 195 254 239
82 63 28
279 141 48 167 151 215 190 138 58 147 65 202 220 95 281 291 247 123
112 57 137
133 68 42 126 78 148 225 61 52 232 177 22 129 68 168 15 74 22 275 250
174 56
147 89 240 218 180 158 35 160 35 245 293 247 208 74 59 52 256 28 20
214 54 21
224 282 46 106 251 249 62 273 15 13 98 271 15 37 136 269 100 210 285
182 12 47
116 292 87 240 242 218 156 120 37 75 158 96 84 39 75 134 58 86 250 198
94 145
174 117 71 93 70 168 287 36 24 190 77 134 143 235 252 113 217 105 261
205 234
262 69 93 94 206 285 121 197 43 286 46 276 66 253 188 223 231 187 87 5
57 222
95 70 253 249 38 235 17 224 221 231 102 125 3 142 99 135 279 56 86 165
236 92
230 51 157 280 5 203 295 101 111 217 20 3 203 23 139 109 6 193 43 116
122 272
236 7 268 49 242 157 45 116 207 125 238 31 95 9 33 245 220 ^
638 0 284 204 50 42 199 167 127 153 45 53 93 121 111 79 282 129 151
213 166 97
57 60 188 53 10 182 148 162 244 47 105 37 210 61 187 137 268 184 63
285 97 134

194 201 82 250 112 132 142 257 290 109 4 277 3 118 37 96 195 94 60 211
114 47
265 110 80 198 168 149 1 43 138 267 268 24 287 12 257 153 288 171 117
281 130
67 252 164 76 231 101 101 261 143 13 256 237 143 173 32 107 89 154 44
31 245
123 26 202 135 216 240 80 38 286 279 55 62 126 237 6 55 8 244 70 166
74 293 23
105 91 92 228 27 118 280 214 28 254 79 199 84 188 296 128 15 11 181
232 17 246
259 228 187 150 54 9 266 109 164 159 170 64 18 2 225 59 135 193 179 43
252 27
255 101 47 82 273 173 275 225 219 17 63 125 24 264 57 220 70 29 253
163 160 86
103 16 87 102 100 48 104 15 171 63 102 260 227 271 258 152 266 132 119
196 253
26 42 62 49 103 99 26 65 17 86 80 61 50 19 138 24 44 191 127 99 269
183 147 140
99 49 20 54 169 42 276 4 140 182 94 280 77 207 124 25 180 96 218 155
203 294 76
71 219 231 124 162 1 226 272 174 191 264 217 102 237 166 234 175 190
51 72 104
152 289 87 262 279 215 241 293 77 284 67 229 209 39 148 139 183 10 22
75 292 1
66 93 7 54 283 205 72 81 197 252 4 3 64 261 120 58 113 154 181 255 78
283 45
190 211 251 271 98 270 99 186 138 184 223 61 73 80 175 107 6 215 81
114 207 147
86 26 179 180 73 157 118 120 35 158 11 98 293 160 32 221 146 247 121
88 18 139
46 189 189 212 273 13 96 58 273 36 51 95 59 295 75 49 180 233 39 285
257 16 64
41 63 262 142 97 111 239 58 296 288 194 178 21 213 29 170 224 197 169
65 40 41
223 144 12 234 92 145 88 155 66 176 243 208 200 243 48 21 76 208 40 64
178 181
95 196 116 205 209 199 30 9 23 288 34 296 151 76 48 264 194 41 200 83
272 151
130 227 134 34 22 212 113 131 226 112 292 24 127 236 90 16 10 159 193
19 161 85
187 274 89 17 223 7 191 104 38 126 100 185 222 145 225 82 90 230 262 5
276 87
79 39 11 21 53 147 3 278 216 263 35 75 72 241 79 122 247 281 267 185
85 27 95
219 71 242 292 161 245 69 144 35 295 235 228 50 179 20 287 172 139 23
128 152
129 14 95 36 234 38 72 218 18 85 164 22 52 231 261 280 131 89 291 162
161 108
142 14 172 276 141 258 15 148 2 65 177 265 56 198 31 31 168 286 119
173 192 286
256 100 25 43 275 88 101 184 9 106 227 294 213 189 52 94 143 150 77 70
6 192

126 77 124 185 62 263 58 136 109 27 41 40 201 105 14 149 215 104 175
53 230 219
^
635 1 239 133 297 115 119 248 195 9 183 166 199 238 130 279 74 121 163
19 265
270 257 66 8 126 204 71 294 234 246 89 173 101 170 29 85 44 154 256
154 202 210
115 65 5 108 163 73 209 58 103 176 264 267 40 25 10 41 28 28 82 26 113
258 69
26 167 202 156 171 277 159 297 7 231 174 104 88 70 83 29 31 67 111 44
46 69 80
111 289 2 82 234 1 33 160 246 280 285 27 79 108 72 260 84 149 124 289
247 38
147 55 222 4 105 245 32 211 272 226 133 52 59 206 194 6 265 25 201 153
253 236
161 212 30 188 261 195 254 239 79 54 28 273 138 39 167 148 212 184 132
49 141
65 196 220 89 281 288 244 120 106 57 137 127 62 33 123 78 142 222 58
49 232 177
22 129 68 168 6 74 13 275 244 174 50 144 83 237 218 174 158 32 157 29
239 287
241 202 65 59 52 250 19 20 214 45 12 218 282 46 106 248 249 53 270 12
10 95 271
15 28 136 269 100 204 279 182 9 38 110 286 84 240 236 215 156 117 34
72 152 96
75 33 69 134 55 77 247 192 91 142 168 114 62 87 67 165 287 33 15 190
68 131 137
235 249 107 214 99 255 205 228 256 63 90 85 206 285 118 191 34 280 40
270 66
253 182 217 225 184 78 5 54 219 86 64 247 246 35 229 14 224 218 228
102 119 3
139 96 129 276 56 80 165 233 89 227 45 157 274 295 200 295 92 108 217
17 296
203 14 133 106 3 193 40 116 122 266 236 7 268 43 242 151 39 113 204
125 238 22
92 30 242 205 282 200 50 36 195 163 127 153 45 49 87 121 109 75 278
125 147 209
166 91 51 56 188 53 8 180 148 158 240 45 103 31 208 55 185 135 266 182
63 281
97 132 194 197 78 250 110 132 140 253 290 105 275 298 118 37 90 193 94
60 209
112 47 263 110 78 196 166 145 294 43 134 263 264 22 283 12 255 151 284
169 117
279 128 65 250 164 76 227 99 101 261 141 11 252 233 143 171 32 103 85
152 42 31
245 123 24 198 135 212 238 76 36 286 275 53 62 124 235 4 51 2 244 66
162 70 291
17 105 91 88 224 27 116 276 210 24 252 79 199 84 186 292 128 11 7 177
230 17
242 259 226 187 295 146 52 5 262 107 160 155 168 60 18 221 55 131 191
179 37

252 25 255 97 41 82 271 169 271 223 215 11 61 121 18 262 53 216 68 23
249 161
158 86 101 16 85 98 98 44 100 13 167 57 100 258 223 269 254 148 266
128 119 194
251 22 38 56 43 97 95 26 59 15 82 80 57 46 15 136 24 44 191 125 93 265
179 145
140 99 47 18 54 165 42 276 2 136 178 88 278 71 203 122 21 180 94 214
153 199
292 74 67 219 231 120 160 1 224 272 172 189 264 213 96 237 164 230 171
186 153 ^
638 1 72 92 152 285 87 254 275 211 241 293 77 284 55 229 201 35 140
135 179 2
18 63 284 298 62 93 300 42 283 201 64 73 189 244 4 292 52 253 120 46
113 146
181 247 74 279 41 186 203 251 263 90 266 95 186 138 184 223 61 61 68
175 107 6
207 81 114 207 139 74 14 175 172 73 153 110 112 27 150 11 98 285 152
20 221 146
239 113 88 14 139 38 181 189 208 265 5 88 50 273 24 51 91 55 287 67 49
176 233
31 281 257 8 64 37 55 258 134 89 111 231 58 292 288 186 174 17 209 25
170 216
197 169 61 32 41 215 136 4 234 92 145 84 155 54 168 243 208 200 239 48
21 68
204 40 60 178 177 87 196 108 197 205 195 18 5 23 284 26 296 147 64 44
256 190
33 196 79 264 151 130 223 126 34 14 204 105 127 218 112 292 16 119 228
82 4 299
159 185 19 153 81 179 274 85 9 219 296 183 104 30 126 92 181 218 141
225 70 90
230 262 294 268 75 75 39 300 13 49 147 300 278 216 255 35 75 68 233 67
114 243
273 267 185 77 19 95 215 71 238 288 157 241 57 144 23 291 227 228 46
171 12 289
283 172 131 19 124 148 125 2 83 36 230 26 60 214 10 73 160 10 44 227
257 280
131 77 291 162 161 100 142 14 168 272 141 254 3 144 295 57 173 257 44
198 23 19
160 282 111 173 192 278 252 88 21 39 271 80 89 176 298 98 227 286 213
185 48 86
135 150 69 58 299 188 118 73 124 177 58 263 54 132 109 15 29 36 201 93
6 149
215 100 171 41 222 36 237 133 293 113 117 248 193 9 183 162 197 234
128 279 70
119 163 15 265 270 255 64 8 122 204 69 294 232 242 89 169 97 166 27 85
38 152
256 154 200 210 115 61 3 108 161 71 209 54 103 176 262 265 34 21 10 37
26 24 80
24 109 258 69 22 165 202 154 171 277 155 297 1 229 172 102 86 70 83 29
29 67
107 42 40 67 78 109 287 2 76 230 298 33 160 246 278 281 21 75 104 66
260 78 145

120 289 247 38 145 55 222 103 241 28 211 270 226 131 48 57 202 190 4
263 25 199
149 251 234 157 212 30 184 259 195 254 239 77 48 28 269 136 33 167 146
210 180
128 43 137 65 192 220 85 281 286 242 118 102 57 137 123 58 27 121 78
138 220 56
47 232 177 22 129 68 168 74 7 275 240 174 46 142 79 235 218 170 158 30
155 25
235 283 237 198 59 59 52 246 13 20 214 39 6 214 282 46 106 246 249 47
268 10 8
93 271 15 22 136 269 100 200 275 182 7 32 106 282 82 240 232 213 156
115 32 70
148 96 69 29 65 134 53 71 245 188 89 140 164 112 56 83 65 163 287 31 9
190 62
129 133 235 247 103 212 95 251 205 224 252 59 88 79 206 285 116 187 28
276 263 ^
640 0 264 66 253 176 211 219 181 69 5 51 216 77 58 241 243 32 223 11
224 215
225 102 113 3 136 93 123 273 56 74 165 230 86 224 39 157 268 289 197
295 83 105
217 14 293 203 5 127 103 193 37 116 122 260 236 7 268 37 242 145 33
110 201 125
238 13 89 293 27 239 202 279 194 50 27 189 157 127 153 45 43 78 121
106 69 272
119 141 203 166 82 42 50 188 53 5 177 148 152 234 42 100 22 205 46 182
132 263
179 63 275 97 129 194 191 72 250 107 132 137 247 290 99 296 272 295
118 37 81
190 94 60 206 109 47 260 110 75 193 163 139 288 43 128 257 258 19 277
12 252
148 278 166 117 276 125 62 247 164 76 221 96 101 261 138 8 246 227 143
168 32
97 79 149 39 31 245 123 21 192 135 206 235 70 33 286 269 50 62 121 232
1 45 295
244 60 156 64 288 8 105 91 82 218 27 113 270 204 18 249 79 199 84 183
286 128 5
1 171 227 17 236 259 223 187 292 140 49 301 256 104 154 149 165 54 18
299 215
49 125 188 179 28 252 22 255 91 32 82 268 163 265 220 209 2 58 115 9
259 47 210
65 14 243 158 155 86 98 16 82 92 95 38 94 10 161 48 97 255 217 266 248
142 266
122 119 191 248 16 32 47 34 88 89 26 50 12 76 80 51 40 9 133 24 44 191
122 84
259 173 142 140 99 44 15 54 159 42 276 301 130 172 79 275 62 197 119
15 180 91
208 150 193 289 71 61 219 231 114 157 1 221 272 169 186 264 207 87 237
161 224
165 180 41 72 89 152 284 87 252 274 210 241 293 77 284 52 229 199 34
138 134
178 17 60 282 298 61 93 299 39 283 200 62 71 187 242 4 290 49 251 120
43 113

144 181 245 73 278 40 185 201 251 261 88 265 94 186 138 184 223 61 58
65 175
107 6 205 81 114 207 137 71 11 174 170 73 152 108 110 25 148 11 98 283
150 17
221 146 237 111 88 13 139 36 179 189 207 263 3 86 48 273 21 51 90 54
285 65 49
175 233 29 280 257 6 64 36 53 257 132 87 111 229 58 291 288 184 173 16
208 24
170 214 197 169 60 30 41 213 134 2 234 92 145 83 155 51 166 243 208
200 238 48
21 66 203 40 59 178 176 85 196 106 195 204 194 15 4 23 283 24 296 146
61 43 254
189 31 195 78 262 151 130 222 124 34 12 202 103 126 216 112 292 14 117
226 80 1
297 159 183 19 151 80 177 274 84 7 218 294 181 104 28 126 90 180 217
140 225 67
90 230 262 292 266 72 74 39 298 11 48 147 300 278 216 253 35 75 67 231
64 112
242 271 267 185 75 17 95 214 71 237 287 156 240 54 144 20 290 225 228
45 169 10
287 282 172 129 18 123 147 124 301 80 36 229 23 57 213 8 70 159 7 42
226 256
280 131 213 ^
640 0 291 162 161 94 142 14 165 269 141 251 298 141 292 51 170 251 35
198 17 10
154 279 105 173 192 272 249 79 18 36 268 74 80 170 292 92 227 280 213
182 45 80
129 150 63 49 296 185 112 70 124 171 55 263 51 129 109 6 20 33 201 84
149 215
97 168 32 216 27 234 133 287 110 114 248 190 9 183 156 194 228 125 279
64 116
163 9 265 270 252 61 8 116 204 66 294 229 236 89 163 91 160 24 85 29
149 256
154 197 210 115 55 108 158 68 209 48 103 176 259 262 25 15 10 31 23 18
77 21
103 258 69 16 162 202 151 171 277 149 297 296 226 169 99 83 70 83 29
26 67 101
39 31 64 75 106 284 2 67 224 295 33 160 246 275 275 12 69 98 57 260 69
139 114
289 247 38 142 55 222 298 100 235 22 211 267 226 128 42 54 196 184 1
260 25 196
143 248 231 151 212 30 178 256 195 254 239 74 39 28 263 133 24 167 143
207 174
122 34 131 65 186 220 79 281 283 239 115 96 57 137 117 52 18 118 78
132 217 53
44 232 177 22 129 68 168 295 74 302 275 234 174 40 139 73 232 218 164
158 27
152 19 229 277 231 192 50 59 52 240 4 20 214 30 301 208 282 46 106 243
249 38
265 7 5 90 271 15 13 136 269 100 194 269 182 4 23 100 276 79 240 226
210 156

112 29 67 142 96 60 23 59 134 50 62 242 182 86 137 158 109 47 77 62
160 287 28
190 53 126 127 235 244 97 209 89 245 205 218 246 53 85 70 206 285 113
181 19
270 30 260 66 253 172 207 215 179 63 5 49 214 71 54 237 241 30 219 9
224 213
223 102 109 3 134 91 119 271 56 70 165 228 84 222 35 157 264 285 195
295 77 103
217 12 291 203 303 123 101 302 193 35 116 122 256 236 7 268 33 242 141
29 108
199 125 238 7 87 289 25 237 200 277 190 50 21 185 153 127 153 45 39 72
121 104
65 268 115 137 199 166 76 36 46 188 53 3 175 148 148 230 40 98 16 203
40 180
130 261 177 63 271 97 127 194 187 68 250 105 132 135 243 290 95 294
270 293 118
37 75 188 94 60 204 107 47 258 110 73 191 161 135 284 43 124 253 254
17 273 12
250 146 274 164 117 274 123 60 245 164 76 217 94 101 261 136 6 242 223
143 166
32 93 75 147 37 31 245 123 19 188 135 202 233 66 31 286 265 48 62 119
230 303
41 291 244 56 152 60 286 2 105 91 78 214 27 111 266 200 14 247 79 199
84 181
282 128 1 301 167 225 17 232 259 221 187 290 136 47 299 252 102 150
145 163 50
18 299 211 45 121 186 179 22 252 20 255 87 26 82 266 159 261 218 205
300 56 111
3 257 43 206 63 8 239 156 153 86 96 16 80 88 93 34 90 8 157 42 95 253
213 264
244 138 266 118 119 189 246 79 ^
650 0 24 35 22 76 81 26 38 8 68 80 43 32 1 129 24 44 191 118 72 251
165 138 140
99 40 11 54 151 42 276 301 122 164 67 271 50 189 115 7 180 87 200 146
185 285
67 53 219 231 106 153 1 217 272 165 182 264 199 75 237 157 216 157 172
33 72 77
152 280 87 244 270 206 241 293 77 284 40 229 191 30 130 130 174 298 13
48 274
298 57 93 295 27 283 196 54 63 179 234 4 282 37 243 120 31 113 136 181
237 69
274 36 181 193 251 253 80 261 90 186 138 184 223 61 46 53 175 107 6
197 81 114
207 129 59 305 170 162 73 148 100 102 17 140 11 98 275 142 5 221 146
229 103 88
9 139 28 171 189 203 255 301 78 40 273 9 51 86 50 277 57 49 171 233 21
276 257
304 64 32 45 253 124 79 111 221 58 287 288 176 169 12 204 20 170 206
197 169 56
22 41 205 126 300 234 92 145 79 155 39 158 243 208 200 234 48 21 58
199 40 55

178 172 77 196 98 187 200 190 3 23 279 16 296 142 49 39 246 185 23 191
74 254
151 130 218 116 34 4 194 95 122 208 112 292 6 109 218 72 295 289 159
175 19 143
76 169 274 80 305 214 286 173 104 20 126 82 176 213 136 225 55 90 230
262 284
258 60 70 39 290 3 44 147 300 278 216 245 35 75 63 223 52 104 238 263
267 185
67 9 95 210 71 233 283 152 236 42 144 8 286 217 228 41 161 2 279 278
172 121 14
119 143 120 293 68 36 225 11 45 209 58 155 301 34 222 252 280 131 62
291 162
161 90 142 14 163 267 141 249 294 139 290 47 168 247 29 198 13 4 150
277 101
173 192 268 247 73 16 34 266 70 74 166 288 88 227 276 213 180 43 76
125 150 59
43 294 183 108 68 124 167 53 263 49 127 109 14 31 201 78 302 149 215
95 166 26
212 21 232 133 283 108 112 248 188 9 183 152 192 224 123 279 60 114
163 5 265
270 250 59 8 112 204 64 294 227 232 89 159 87 156 22 85 23 147 256 154
195 210
115 51 304 108 156 66 209 44 103 176 257 260 19 11 10 27 21 14 75 19
99 258 69
12 160 202 149 171 277 145 297 292 224 167 97 81 70 83 29 24 67 97 37
25 62 73
104 282 2 61 220 293 33 160 246 273 271 6 65 94 51 260 63 135 110 289
247 38
140 55 222 296 98 231 18 211 265 226 126 38 52 192 180 305 258 25 194
139 246
229 147 212 30 174 254 195 254 239 72 33 28 259 131 18 167 141 205 170
118 28
127 65 182 220 75 281 281 237 113 92 57 137 113 48 12 116 78 128 215
51 42 232
177 22 129 68 168 291 74 298 275 230 174 36 137 69 230 218 160 158 25
150 15
225 273 227 188 44 59 52 236 304 20 214 24 297 204 282 46 106 241 249
32 263 5
3 88 271 15 7 136 269 100 190 265 182 2 17 96 283 ^
649 0 76 240 220 207 156 109 26 64 136 96 51 17 53 134 47 53 239 176
83 134 152
106 38 71 59 157 287 25 298 190 44 123 121 235 241 91 206 83 239 205
212 240 47
82 61 206 285 110 175 10 264 24 254 66 253 166 201 209 176 54 5 46 211
62 48
231 238 27 213 6 224 210 220 102 103 3 131 88 113 268 56 64 165 225 81
219 29
157 258 279 192 295 68 100 217 9 288 203 297 117 98 302 193 32 116 122
250 236
7 268 27 242 135 23 105 196 125 238 305 84 283 22 234 197 274 184 50
12 179 147

127 153 45 33 63 121 101 59 262 109 131 193 166 67 27 40 188 53 172
148 142 224
37 95 7 200 31 177 127 258 174 63 265 97 124 194 181 62 250 102 132
132 237 290
89 291 267 290 118 37 66 185 94 60 201 104 47 255 110 70 188 158 129
278 43 118
247 248 14 267 12 247 143 268 161 117 271 120 57 242 164 76 211 91 101
261 133
3 236 217 143 163 32 87 69 144 34 31 245 123 16 182 135 196 230 60 28
286 259
45 62 116 227 303 35 285 244 50 146 54 283 300 105 91 72 208 27 108
260 194 8
244 79 199 84 178 276 128 302 298 161 222 17 226 259 218 187 287 130
44 296 246
99 144 139 160 44 18 299 205 39 115 183 179 13 252 17 255 81 17 82 263
153 255
215 199 294 53 105 301 254 37 200 60 306 233 153 150 86 93 16 77 82 90
28 84 5
151 33 92 250 207 261 238 132 266 112 119 186 243 6 22 32 19 73 79 26
35 7 66
80 41 30 306 128 24 44 191 117 69 249 163 137 140 99 39 10 54 149 42
276 301
120 162 64 270 47 187 114 5 180 86 198 145 183 284 66 51 219 231 104
152 1 216
272 164 181 264 197 72 237 156 214 155 170 31 72 74 152 279 87 242 269
205 241
293 77 284 37 229 189 29 128 129 173 297 12 45 272 298 56 93 294 24
283 195 52
61 177 232 4 280 34 241 120 28 113 134 181 235 68 273 35 180 191 251
251 78 260
89 186 138 184 223 61 43 50 175 107 6 195 81 114 207 127 56 303 169
160 73 147
98 100 15 138 11 98 273 140 2 221 146 227 101 88 8 139 26 169 189 202
253 300
76 38 273 6 51 85 49 275 55 49 170 233 19 275 257 303 64 31 43 252 122
77 111
219 58 286 288 174 168 11 203 19 170 204 197 169 55 20 41 203 124 299
234 92
145 78 155 36 156 243 208 200 233 48 21 56 198 40 54 178 171 75 196 96
185 199
189 306 23 278 14 296 141 46 38 244 184 21 190 73 252 151 130 217 114
34 2 192
93 121 206 112 292 4 107 216 70 293 287 159 173 19 141 75 167 274 79
304 213
284 171 104 18 126 80 175 212 135 225 52 90 230 262 282 256 57 69 39
288 1 43
147 300 278 216 243 35 75 62 221 49 102 237 261 267 140 ^
649 1 61 3 95 207 71 230 280 149 233 33 144 308 283 211 228 38 155 305
273 275
172 115 11 116 140 117 287 59 36 222 2 36 206 303 49 152 295 28 219
249 280 131

53 291 162 161 84 142 14 160 264 141 246 288 136 287 41 165 241 20 198
7 304
144 274 95 173 192 262 244 64 13 31 263 64 65 160 282 82 227 270 213
177 40 70
119 150 53 34 291 180 102 65 124 161 50 263 46 124 109 300 5 28 201 69
299 149
215 92 163 17 206 12 229 133 277 105 109 248 185 9 183 146 189 218 120
279 54
111 163 308 265 270 247 56 8 106 204 61 294 224 226 89 153 81 150 19
85 14 144
256 154 192 210 115 45 304 108 153 63 209 38 103 176 254 257 10 5 10
21 18 8 72
16 93 258 69 6 157 202 146 171 277 139 297 286 221 164 94 78 70 83 29
21 67 91
34 16 59 70 101 279 2 52 214 290 33 160 246 270 265 306 59 88 42 260
54 129 104
289 247 38 137 55 222 293 95 225 12 211 262 226 123 32 49 186 174 305
255 25
191 133 243 226 141 212 30 168 251 195 254 239 69 24 28 253 128 9 167
138 202
164 112 19 121 65 176 220 69 281 278 234 110 86 57 137 107 42 3 113 78
122 212
48 39 232 177 22 129 68 168 285 74 292 275 224 174 30 134 63 227 218
154 158 22
147 9 219 267 221 182 35 59 52 230 298 20 214 15 291 198 282 46 106
238 249 23
260 2 85 271 15 307 136 269 100 184 259 182 308 8 90 266 74 240 216
205 156 107
24 62 132 96 45 13 49 134 45 47 237 172 81 132 148 104 32 67 57 155
287 23 294
190 38 121 117 235 239 87 204 79 235 205 208 236 43 80 55 206 285 108
171 4 260
20 250 66 253 162 197 205 174 48 5 44 209 56 44 227 236 25 209 4 224
208 218
102 99 3 129 86 109 266 56 60 165 223 79 217 25 157 254 275 190 295 62
98 217 7
286 203 293 113 96 302 193 30 116 122 246 236 7 268 23 242 131 19 103
194 125
238 301 82 279 20 232 195 272 180 50 6 175 143 127 153 45 29 57 121 99
55 258
105 127 189 166 61 21 36 188 53 307 170 148 138 220 35 93 1 198 25 175
125 256
172 63 261 97 122 194 177 58 250 100 132 130 233 290 85 289 265 288
118 37 60
183 94 60 199 102 47 253 110 68 186 156 125 274 43 114 243 244 12 263
12 245
141 264 159 117 269 118 55 240 164 76 207 89 101 261 131 1 232 213 143
161 32
83 65 142 32 31 245 123 14 178 135 192 228 56 26 286 255 43 62 114 225
303 31
281 244 46 142 50 281 296 105 91 68 204 27 106 256 190 4 242 79 199 84
176 272

128 300 296 157 220 17 222 259 216 187 285 126 42 294 242 97 140 135
158 40 18
299 201 35 111 181 179 7 252 15 255 77 11 82 261 149 251 69 ^
655 0 191 286 49 97 293 250 29 192 56 298 225 149 146 86 89 16 73 74
86 20 76 1
143 21 88 246 199 257 230 124 266 104 119 182 239 309 14 20 7 61 71 26
23 3 58
80 33 22 302 124 24 44 191 113 57 241 155 133 140 99 35 6 54 141 42
276 301 112
154 52 266 35 179 110 308 180 82 190 141 175 280 62 43 219 231 96 148
1 212 272
160 177 264 189 60 237 152 206 147 162 23 72 62 152 275 87 234 265 201
241 293
77 284 25 229 181 25 120 125 169 293 8 33 264 298 52 93 290 12 283 191
44 53
169 224 4 272 22 233 120 16 113 126 181 227 64 269 31 176 183 251 243
70 256 85
186 138 184 223 61 31 38 175 107 6 187 81 114 207 119 44 295 165 152
73 143 90
92 7 130 11 98 265 132 301 221 146 219 93 88 4 139 18 161 189 198 245
296 68 30
273 305 51 81 45 267 47 49 166 233 11 271 257 299 64 27 35 248 114 69
111 211
58 282 288 166 164 7 199 15 170 196 197 169 51 12 41 195 116 295 234
92 145 74
155 24 148 243 208 200 229 48 21 48 194 40 50 178 167 67 196 88 177
195 185 299
306 23 274 6 296 137 34 34 236 180 13 186 69 244 151 130 213 106 34
305 184 85
117 198 112 292 307 99 208 62 285 279 159 165 19 133 71 159 274 75 300
209 276
163 104 10 126 72 171 208 131 225 40 90 230 262 274 248 45 65 39 280
304 39 147
300 278 216 235 35 75 58 213 37 94 233 253 267 185 57 310 95 205 71
228 278 147
231 27 144 304 281 207 228 36 151 303 269 273 172 111 9 114 138 115
283 53 36
220 307 30 204 301 43 150 291 24 217 247 280 131 47 291 162 161 80 142
14 158
262 141 244 284 134 285 37 163 237 14 198 3 300 140 272 91 173 192 258
242 58
11 29 261 60 59 156 278 78 227 266 213 175 38 66 115 150 49 28 289 178
98 63
124 157 48 263 44 122 109 296 310 26 201 63 297 149 215 90 161 11 202
6 227 133
273 103 107 248 183 9 183 142 187 214 118 279 50 109 163 306 265 270
245 54 8
102 204 59 294 222 222 89 149 77 146 17 85 8 142 256 154 190 210 115
41 304 108
151 61 209 34 103 176 252 255 4 1 10 17 16 4 70 14 89 258 69 2 155 202
144 171

277 135 297 282 219 162 92 76 70 83 29 19 67 87 32 10 57 68 99 277 2
46 210 288
33 160 246 268 261 302 55 84 36 260 48 125 100 289 247 38 135 55 222
291 93 221
8 211 260 226 121 28 47 182 170 305 253 25 189 129 241 224 137 212 30
164 249
195 254 239 67 18 28 249 126 3 167 136 200 160 108 13 117 65 172 220
65 281 276
232 108 82 57 137 103 38 308 111 78 118 210 46 37 232 177 22 129 68
168 281 74
288 275 220 174 26 132 59 225 218 150 158 20 145 5 215 263 100 ^
661 1 176 26 59 52 224 292 20 214 6 285 192 282 46 106 235 249 14 257
311 309
82 271 15 301 136 269 100 178 253 182 308 311 84 260 71 240 210 202
156 104 21
59 126 96 36 7 43 134 42 38 234 166 78 129 142 101 23 61 54 152 287 20
288 190
29 118 111 235 236 81 201 73 229 205 202 230 37 77 46 206 285 105 165
307 254
14 244 66 253 156 191 199 171 39 5 41 206 47 38 221 233 22 203 1 224
205 215
102 93 3 126 83 103 263 56 54 165 220 76 214 19 157 248 269 187 295 53
95 217 4
283 203 287 107 93 302 193 27 116 122 240 236 7 268 17 242 125 13 100
191 125
238 295 79 273 17 229 192 269 174 50 309 169 137 127 153 45 23 48 121
96 49 252
99 121 183 166 52 12 30 188 53 307 167 148 132 214 32 90 304 195 16
172 122 253
169 63 255 97 119 194 171 52 250 97 132 127 227 290 79 286 262 285 118
37 51
180 94 60 196 99 47 250 110 65 183 153 119 268 43 108 237 238 9 257 12
242 138
258 156 117 266 115 52 237 164 76 201 86 101 261 128 310 226 207 143
158 32 77
59 139 29 31 245 123 11 172 135 186 225 50 23 286 249 40 62 111 222
303 25 275
244 40 136 44 278 290 105 91 62 198 27 103 250 184 310 239 79 199 84
173 266
128 297 293 151 217 17 216 259 213 187 282 120 39 291 236 94 134 129
155 34 18
299 195 29 105 178 179 310 252 12 255 71 2 82 258 143 245 210 189 284
48 95 291
249 27 190 55 296 223 148 145 86 88 16 72 72 85 18 74 141 18 87 245
197 256 228
122 266 102 119 181 238 308 12 17 4 58 69 26 20 2 56 80 31 20 301 123
24 44 191
112 54 239 153 132 140 99 34 5 54 139 42 276 301 110 152 49 265 32 177
109 307
180 81 188 140 173 279 61 41 219 231 94 147 1 211 272 159 176 264 187
57 237

151 204 145 160 21 72 59 152 274 87 232 264 200 241 293 77 284 22 229
179 24
118 124 168 292 7 30 262 298 51 93 289 9 283 190 42 51 167 222 4 270
19 231 120
13 113 124 181 225 63 268 30 175 181 251 241 68 255 84 186 138 184 223
61 28 35
175 107 6 185 81 114 207 117 41 293 164 150 73 142 88 90 5 128 11 98
263 130
299 221 146 217 91 88 3 139 16 159 189 197 243 295 66 28 273 303 51 80
44 265
45 49 165 233 9 270 257 298 64 26 33 247 112 67 111 209 58 281 288 164
163 6
198 14 170 194 197 169 50 10 41 193 114 294 234 92 145 73 155 21 146
243 208
200 228 48 21 46 193 40 49 178 166 65 196 86 175 194 184 297 306 23
273 4 296
136 31 33 234 179 11 185 68 242 151 130 212 104 34 304 182 83 116 196
112 292
306 97 206 60 283 277 159 163 19 131 70 157 274 74 299 208 274 161 104
8 126 70
170 207 130 225 37 90 230 194 ^
643 0 268 242 36 62 39 274 301 36 147 300 278 216 229 35 75 55 207 28
88 230
247 267 185 51 307 95 202 71 225 275 144 228 18 144 298 278 201 228 33
145 300
263 270 172 105 6 111 135 112 277 44 36 217 301 21 201 298 34 147 285
18 214
244 280 131 38 291 162 161 74 142 14 155 259 141 241 278 131 282 31
160 231 5
198 311 294 134 269 85 173 192 252 239 49 8 26 258 54 50 150 272 72
227 260 213
172 35 60 109 150 43 19 286 175 92 60 124 151 45 263 41 119 109 290
304 23 201
54 294 149 215 87 158 2 196 311 224 133 267 100 104 248 180 9 183 136
184 208
115 279 44 106 163 303 265 270 242 51 8 96 204 56 294 219 216 89 143
71 140 14
85 313 139 256 154 187 210 115 35 304 108 148 58 209 28 103 176 249
252 309 309
10 11 13 312 67 11 83 258 69 310 152 202 141 171 277 129 297 276 216
159 89 73
70 83 29 16 67 81 29 1 54 65 96 274 2 37 204 285 33 160 246 265 255
296 49 78
27 260 39 119 94 289 247 38 132 55 222 288 90 215 2 211 257 226 118 22
44 176
164 305 250 25 186 123 238 221 131 212 30 158 246 195 254 239 64 9 28
243 123
308 167 133 197 154 102 4 111 65 166 220 59 281 273 229 105 76 57 137
97 32 302
108 78 112 207 43 34 232 177 22 129 68 168 275 74 282 275 214 174 20
129 53 222

218 144 158 17 142 313 209 257 211 172 20 59 52 220 288 20 214 281 188
282 46
106 233 249 8 255 311 309 80 271 15 297 136 269 100 174 249 182 308
307 80 256
69 240 206 200 156 102 19 57 122 96 30 3 39 134 40 32 232 162 76 127
138 99 17
57 52 150 287 18 284 190 23 116 107 235 234 77 199 69 225 205 198 226
33 75 40
206 285 103 161 303 250 10 240 66 253 152 187 195 169 33 5 39 204 41
34 217 231
20 199 313 224 203 213 102 89 3 124 81 99 261 56 50 165 218 74 212 15
157 244
265 185 295 47 93 217 2 281 203 283 103 91 302 193 25 116 122 236 236
7 268 13
242 121 9 98 189 125 238 291 77 269 15 227 190 267 170 50 305 165 133
127 153
45 19 42 121 94 45 248 95 117 179 166 46 6 26 188 53 307 165 148 128
210 30 88
300 193 10 170 120 251 167 63 251 97 117 194 167 48 250 95 132 125 223
290 75
284 260 283 118 37 45 178 94 60 194 97 47 248 110 63 181 151 115 264
43 104 233
234 7 253 12 240 136 254 154 117 264 113 50 235 164 76 197 84 101 261
126 310
222 203 143 156 32 73 55 137 27 31 245 123 9 168 135 182 223 46 21 286
245 38
62 109 220 303 21 271 244 36 132 40 276 286 105 91 58 194 27 101 246
180 308
237 79 199 84 171 262 128 295 291 147 215 17 212 259 219 ^
652 1 187 278 112 35 287 228 90 126 121 151 26 18 299 187 21 97 174
179 302 252
8 255 63 306 82 254 135 237 206 181 276 44 87 283 245 19 182 51 288
215 144 141
86 84 16 68 64 81 10 66 312 133 6 83 241 189 252 220 114 266 94 119
177 234 304
4 5 308 46 61 26 8 314 48 80 23 12 297 119 24 44 191 108 42 231 145
128 140 99
30 1 54 131 42 276 301 102 144 37 261 20 169 105 303 180 77 180 136
165 275 57
33 219 231 86 143 1 207 272 155 172 264 179 45 237 147 196 137 152 13
72 47 152
270 87 224 260 196 241 293 77 284 10 229 171 20 110 120 164 288 3 18
254 298 47
93 285 313 283 186 34 43 159 214 4 262 7 223 120 1 113 116 181 217 59
264 26
171 173 251 233 60 251 80 186 138 184 223 61 16 23 175 107 6 177 81
114 207 109
29 285 160 142 73 138 80 82 313 120 11 98 255 122 291 221 146 209 83
88 315 139
8 151 189 193 235 291 58 20 273 295 51 76 40 257 37 49 161 233 1 266
257 294 64

22 25 243 104 59 111 201 58 277 288 156 159 2 194 10 170 186 197 169
46 2 41
185 106 290 234 92 145 69 155 9 138 243 208 200 224 48 21 38 189 40 45
178 162
57 196 78 167 190 180 289 306 23 269 312 296 132 19 29 226 175 3 181
64 234 151
130 208 96 34 300 174 75 112 188 112 292 302 89 198 52 275 269 159 155
19 123
66 149 274 70 295 204 266 153 104 126 62 166 203 126 225 25 90 230 262
264 238
30 60 39 270 299 34 147 300 278 216 225 35 75 53 203 22 84 228 243 267
185 47
305 95 200 71 223 273 142 226 12 144 294 276 197 228 31 141 298 259
268 172 101
4 109 133 110 273 38 36 215 297 15 199 296 28 145 281 14 212 242 280
131 32 291
162 161 70 142 14 153 257 141 239 274 129 280 27 158 227 315 198 309
290 130
267 81 173 192 248 237 43 6 24 256 50 44 146 268 68 227 256 213 170 33
56 105
150 39 13 284 173 88 58 124 147 43 263 39 117 109 286 300 21 201 48
292 149 215
85 156 312 192 307 222 133 263 98 102 248 178 9 183 132 182 204 113
279 40 104
163 301 265 270 240 49 8 92 204 54 294 217 212 89 139 67 136 12 85 309
137 256
154 185 210 115 31 304 108 146 56 209 24 103 176 247 250 305 307 10 7
11 310 65
9 79 258 69 308 150 202 139 171 277 125 297 272 214 157 87 71 70 83 29
14 67 77
27 311 52 63 94 272 2 31 200 283 33 160 246 263 251 292 45 74 21 260
33 115 90
289 247 38 130 55 222 286 88 211 314 211 255 226 116 18 42 172 160 305
248 25
184 119 236 219 127 212 30 154 244 195 254 239 62 3 28 239 121 304 167
131 195
150 98 314 107 65 162 220 55 281 271 227 103 72 57 137 93 28 308 ^
670 0 105 78 106 204 40 31 232 177 22 129 68 168 269 74 276 275 208
174 14 126
47 219 218 138 158 14 139 310 203 251 205 166 11 59 52 214 282 20 214
308 275
182 282 46 106 230 249 316 252 311 309 77 271 15 291 136 269 100 168
243 182
308 301 74 250 66 240 200 197 156 99 16 54 116 96 21 314 33 134 37 23
229 156
73 124 132 96 8 51 49 147 287 15 278 190 14 113 101 235 231 71 196 63
219 205
192 220 27 72 31 206 285 100 155 297 244 4 234 66 253 146 181 189 166
24 5 36
201 32 28 211 228 17 193 313 224 200 210 102 83 3 121 78 93 258 56 44
165 215

71 209 9 157 238 259 182 295 38 90 217 316 278 203 277 97 88 302 193
22 116 122
230 236 7 268 7 242 115 3 95 186 125 238 285 74 263 12 224 187 264 164
50 299
159 127 127 153 45 13 33 121 91 39 242 89 111 173 166 37 314 20 188 53
307 162
148 122 204 27 85 294 190 1 167 117 248 164 63 245 97 114 194 161 42
250 92 132
122 217 290 69 281 257 280 118 37 36 175 94 60 191 94 47 245 110 60
178 148 109
258 43 98 227 228 4 247 12 237 133 248 151 117 261 110 47 232 164 76
191 81 101
261 123 310 216 197 143 153 32 67 49 134 24 31 245 123 6 162 135 176
220 40 18
286 239 35 62 106 217 303 15 265 244 30 126 34 273 280 105 91 52 188
27 98 240
174 305 234 79 199 84 168 256 128 292 288 141 212 17 206 259 208 187
277 110 34
286 226 89 124 119 150 24 18 299 185 19 95 173 179 300 252 7 255 61
304 82 253
133 235 205 179 274 43 85 281 244 17 180 50 286 213 143 140 86 83 16
67 62 80 8
64 312 131 3 82 240 187 251 218 112 266 92 119 176 233 303 2 2 306 43
59 26 5
314 46 80 21 10 296 118 24 44 191 107 39 229 143 127 140 99 29 54 129
42 276
301 100 142 34 260 17 167 104 302 180 76 178 135 163 274 56 31 219 231
84 142 1
206 272 154 171 264 177 42 237 146 194 135 150 11 72 44 152 269 87 222
259 195
241 293 77 284 7 229 169 19 108 119 163 287 2 15 252 298 46 93 284 311
283 185
32 41 157 212 4 260 4 221 120 315 113 114 181 215 58 263 25 170 171
251 231 58
250 79 186 138 184 223 61 13 20 175 107 6 175 81 114 207 107 26 283
159 140 73
137 78 80 312 118 11 98 253 120 289 221 146 207 81 88 315 139 6 149
189 192 233
290 56 18 273 293 51 75 39 255 35 49 160 233 316 265 257 293 64 21 23
242 102
57 111 199 58 276 288 154 158 1 193 9 170 184 197 169 45 41 183 104
289 234 92
145 68 155 6 136 243 208 200 223 48 21 36 188 40 44 178 161 55 196 76
165 189
179 287 306 23 268 311 296 131 16 28 224 174 1 180 63 232 151 130 207
94 34 299
172 73 111 186 112 292 301 87 196 50 273 267 159 153 235 ^
658 1 117 63 143 274 67 292 201 260 147 104 313 126 56 163 200 123 225
16 90
230 262 258 232 21 57 39 264 296 31 147 300 278 216 219 35 75 50 197
13 78 225

237 267 185 41 302 95 197 71 220 270 139 223 3 144 288 273 191 228 28
135 295
253 265 172 95 1 106 130 107 267 29 36 212 291 6 196 293 19 142 275 8
209 239
280 131 23 291 162 161 64 142 14 150 254 141 236 268 126 277 21 155
221 309 198
306 284 124 264 75 173 192 242 234 34 3 21 253 44 35 140 262 62 227
250 213 167
30 50 99 150 33 4 281 170 82 55 124 141 40 263 36 114 109 280 294 18
201 39 289
149 215 82 153 306 186 301 219 133 257 95 99 248 175 9 183 126 179 198
110 279
34 101 163 298 265 270 237 46 8 86 204 51 294 214 206 89 133 61 130 9
85 303
134 256 154 182 210 115 25 304 108 143 53 209 18 103 176 244 247 299
304 10 1 8
307 62 6 73 258 69 305 147 202 136 171 277 119 297 266 211 154 84 68
70 83 29
11 67 71 24 305 49 60 91 269 2 22 194 280 33 160 246 260 245 286 39 68
12 260
24 109 84 289 247 38 127 55 222 283 85 205 311 211 252 226 113 12 39
166 154
305 245 25 181 113 233 216 121 212 30 148 241 195 254 239 59 313 28
233 118 298
167 128 192 144 92 308 101 65 156 220 49 281 268 224 100 66 57 137 87
22 292
103 78 102 202 38 29 232 177 22 129 68 168 265 74 272 275 204 174 10
124 43 217
218 134 158 12 137 308 199 247 201 162 5 59 52 210 278 20 214 304 271
178 282
46 106 228 249 312 250 311 309 75 271 15 287 136 269 100 164 239 182
308 297 70
246 64 240 196 195 156 97 14 52 112 96 15 312 29 134 35 17 227 152 71
122 128
94 2 47 47 145 287 13 274 190 8 111 97 235 229 67 194 59 215 205 188
216 23 70
25 206 285 98 151 293 240 230 66 253 142 177 185 164 18 5 34 199 26 24
207 226
15 189 313 224 198 208 102 79 3 119 76 89 256 56 40 165 213 69 207 5
157 234
255 180 295 32 88 217 316 276 203 273 93 86 302 193 20 116 122 226 236
7 268 3
242 111 318 93 184 125 238 281 72 259 10 222 185 262 160 50 295 155
123 127 153
45 9 27 121 89 35 238 85 107 169 166 31 310 16 188 53 307 160 148 118
200 25 83
290 188 314 165 115 246 162 63 241 97 112 194 157 38 250 90 132 120
213 290 65
279 255 278 118 37 30 173 94 60 189 92 47 243 110 58 176 146 105 254
43 94 223
224 2 243 12 235 131 244 149 117 259 108 45 230 164 76 187 79 101 261
121 310

212 193 143 151 32 63 45 132 22 31 245 123 4 158 135 172 218 36 16 286
235 33
62 104 215 303 11 261 244 26 122 30 271 276 105 91 48 184 27 96 236
170 303 232
79 199 84 166 252 128 290 283 ^
649 1 133 208 17 198 259 204 187 273 102 30 282 218 85 116 111 146 16
18 299
177 11 87 169 179 292 252 3 255 53 296 82 249 125 227 201 171 266 39
77 273 240
9 172 46 278 205 139 136 86 79 16 63 54 76 56 312 123 312 78 236 179
247 210
104 266 84 119 172 229 299 315 311 298 31 51 26 314 314 38 80 13 2 292
114 24
44 191 103 27 221 135 123 140 99 25 317 54 121 42 276 301 92 134 22
256 5 159
100 298 180 72 170 131 155 270 52 23 219 231 76 138 1 202 272 150 167
264 169
30 237 142 186 127 142 3 72 32 152 265 87 214 255 191 241 293 77 284
316 229
161 15 100 115 159 283 319 3 244 298 42 93 280 303 283 181 24 33 149
204 4 252
313 213 120 307 113 106 181 207 54 259 21 166 163 251 223 50 246 75
186 138 184
223 61 1 8 175 107 6 167 81 114 207 99 14 275 155 132 73 133 70 72 308
110 11
98 245 112 281 221 146 199 73 88 315 139 319 141 189 188 225 286 48 10
273 285
51 71 35 247 27 49 156 233 312 261 257 289 64 17 15 238 94 49 111 191
58 272
288 146 154 318 189 5 170 176 197 169 41 313 41 175 96 285 234 92 145
64 155
315 128 243 208 200 219 48 21 28 184 40 40 178 157 47 196 68 157 185
175 279
306 23 264 307 296 127 4 24 216 170 314 176 59 224 151 130 203 86 34
295 164 65
107 178 112 292 297 79 188 42 265 259 159 145 19 113 61 139 274 65 290
199 256
143 104 311 126 52 161 198 121 225 10 90 230 262 254 228 15 55 39 260
294 29
147 300 278 216 215 35 75 48 193 7 74 223 233 267 185 37 300 95 195 71
218 268
137 221 318 144 284 271 187 228 26 131 293 249 263 172 91 320 104 128
105 263
23 36 210 287 194 291 13 140 271 4 207 237 280 131 17 291 162 161 60
142 14 148
252 141 234 264 124 275 17 153 217 305 198 304 280 120 262 71 173 192
238 232
28 1 19 251 40 29 136 258 58 227 246 213 165 28 46 95 150 29 319 279
168 78 53
124 137 38 263 34 112 109 276 290 16 201 33 287 149 215 80 151 302 182
297 217

133 253 93 97 248 173 9 183 122 177 194 108 279 30 99 163 296 265 270
235 44 8
82 204 49 294 212 202 89 129 57 126 7 85 299 132 256 154 180 210 115
21 304 108
141 51 209 14 103 176 242 245 295 302 10 318 6 305 60 4 69 258 69 303
145 202
134 171 277 115 297 262 209 152 82 66 70 83 29 9 67 67 22 301 47 58 89
267 2 16
190 278 33 160 246 258 241 282 35 64 6 260 18 105 80 289 247 38 125 55
222 281
83 201 309 211 250 226 111 8 37 162 150 305 243 25 179 109 231 214 117
212 30
144 239 195 254 239 57 309 28 229 116 294 167 126 190 140 88 304 97 65
152 220
45 281 178 ^
<D

H>SHS Type 3 Strings<H

D>

40 1 18 20 11 15 11 11 9 6 2 15 5 17 12 1 9 1 3 5 11 7 2 2 6 18 2 9 11
8 16 11
16 15 18 18 3 20 2 17 17 16 ^
<D

File = boolean-has-no-value.message-raw

l _____ ÂðãÀLJ°UŠçæx_Ñ5" _çK}_ã<K•„_Eø):»__|fi7»ó_çl&t;H^D`v`
«

$\hat{E}P_{\hat{n}} < \hat{E}$

^aHr_□_žĐtÄE_

íZŸÓ°_de

File = bus-test.bat.cmake

:: bus-test wrapper

@echo off

:: session bus address

set DBUS_STARTER_BUS=tcp:host=localhost,port=1234

if NOT "%1" == "" (

 SET DATADIR=%1

) else (

 SET DATADIR=test\data

)

bin\bus-test.exe test\data

File = bus.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */

/* bus.c message bus context object

*

* Copyright (C) 2003, 2004 Red Hat, Inc.

*

* Licensed under the Academic Free License version 2.1

*

* This program is free software; you can redistribute it and/or
modify

* it under the terms of the GNU General Public License as published
by

* the Free Software Foundation; either version 2 of the License, or

* (at your option) any later version.

*

* This program is distributed in the hope that it will be useful,

* but WITHOUT ANY WARRANTY; without even the implied warranty of

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

* GNU General Public License for more details.

*

* You should have received a copy of the GNU General Public License

* along with this program; if not, write to the Free Software

* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA

02110-1301 USA

*

*/

#include <config.h>

#include "bus.h"

```

#include <stdio.h>

#include "activation.h"
#include "connection.h"
#include "services.h"
#include "utils.h"
#include "policy.h"
#include "config-parser.h"
#include "signals.h"
#include "selinux.h"
#include "dir-watch.h"
#include <dbus/dbus-list.h>
#include <dbus/dbus-hash.h>
#include <dbus/dbus-credentials.h>
#include <dbus/dbus-internals.h>

#ifdef DBUS_CYGWIN
#include <signal.h>
#endif

struct BusContext
{
    int refcount;
    DBusGUID uuid;
    char *config_file;
    char *type;
    char *servicehelper;
    char *address;
    char *pidfile;
    char *user;
    char *log_prefix;
    DBusLoop *loop;
    DBusList *servers;
    BusConnections *connections;
    BusActivation *activation;
    BusRegistry *registry;
    BusPolicy *policy;
    BusMatchmaker *matchmaker;
    BusLimits limits;
    unsigned int fork : 1;
    unsigned int syslog : 1;
    unsigned int keep_umask : 1;
    unsigned int allow_anonymous : 1;
    unsigned int systemd_activation : 1;
};

static dbus_int32_t server_data_slot = -1;

typedef struct
{
    BusContext *context;

```

```

} BusServerData;

#define BUS_SERVER_DATA(server) (dbus_server_get_data ((server),
server_data_slot))

static BusContext*
server_get_context (DBusServer *server)
{
    BusContext *context;
    BusServerData *bd;

    if (!dbus_server_allocate_data_slot (&server_data_slot))
        return NULL;

    bd = BUS_SERVER_DATA (server);
    if (bd == NULL)
    {
        dbus_server_free_data_slot (&server_data_slot);
        return NULL;
    }

    context = bd->context;

    dbus_server_free_data_slot (&server_data_slot);

    return context;
}

static dbus_bool_t
add_server_watch (DBusWatch *watch,
                 void *data)
{
    DBusServer *server = data;
    BusContext *context;

    context = server_get_context (server);

    return _dbus_loop_add_watch (context->loop, watch);
}

static void
remove_server_watch (DBusWatch *watch,
                   void *data)
{
    DBusServer *server = data;
    BusContext *context;

    context = server_get_context (server);

    _dbus_loop_remove_watch (context->loop, watch);
}

```

```

static void
toggle_server_watch (DBusWatch *watch,
                    void *data)
{
    DBusServer *server = data;
    BusContext *context;

    context = server_get_context (server);

    _dbus_loop_toggle_watch (context->loop, watch);
}

static dbus_bool_t
add_server_timeout (DBusTimeout *timeout,
                  void *data)
{
    DBusServer *server = data;
    BusContext *context;

    context = server_get_context (server);

    return _dbus_loop_add_timeout (context->loop, timeout);
}

static void
remove_server_timeout (DBusTimeout *timeout,
                    void *data)
{
    DBusServer *server = data;
    BusContext *context;

    context = server_get_context (server);

    _dbus_loop_remove_timeout (context->loop, timeout);
}

static void
new_connection_callback (DBusServer *server,
                       DBusConnection *new_connection,
                       void *data)
{
    BusContext *context = data;

    if (!bus_connections_setup_connection (context->connections,
                                          new_connection))
        {
            _dbus_verbose ("No memory to setup new connection\n");

            /* if we don't do this, it will get unref'd without
             * being disconnected... kind of strange really
             * that we have to do this, people won't get it right
             * in general.
            */
        }
}

```



```

        */
        dbus_connection_close (new_connection);
    }

    dbus_connection_set_max_received_size (new_connection,
                                           context->
>limits.max_incoming_bytes);

    dbus_connection_set_max_message_size (new_connection,
                                           context->
>limits.max_message_size);

    dbus_connection_set_max_received_unix_fds (new_connection,
                                               context->
>limits.max_incoming_unix_fds);

    dbus_connection_set_max_message_unix_fds (new_connection,
                                              context->
>limits.max_message_unix_fds);

    dbus_connection_set_allow_anonymous (new_connection,
                                         context->allow_anonymous);

    /* on OOM, we won't have ref'd the connection so it will die. */
}

static void
free_server_data (void *data)
{
    BusServerData *bd = data;

    dbus_free (bd);
}

static dbus_bool_t
setup_server (BusContext *context,
             DBusServer *server,
             char **auth_mechanisms,
             DBusError *error)
{
    BusServerData *bd;

    bd = dbus_new0 (BusServerData, 1);
    if (bd == NULL || !dbus_server_set_data (server,
                                             server_data_slot,
                                             bd, free_server_data))
    {
        dbus_free (bd);
        BUS_SET_OOM (error);
        return FALSE;
    }
}

```

```

    bd->context = context;

    if (!dbus_server_set_auth_mechanisms (server, (const char**)
auth_mechanisms))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    dbus_server_set_new_connection_function (server,
                                           new_connection_callback,
                                           context, NULL);

    if (!dbus_server_set_watch_functions (server,
                                         add_server_watch,
                                         remove_server_watch,
                                         toggle_server_watch,
                                         server,
                                         NULL))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!dbus_server_set_timeout_functions (server,
                                           add_server_timeout,
                                           remove_server_timeout,
                                           NULL,
                                           server, NULL))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}

/* This code only gets executed the first time the
 * config files are parsed. It is not executed
 * when config files are reloaded.
 */
static dbus_bool_t
process_config_first_time_only (BusContext *context,
                               BusConfigParser *parser,
                               const DBusString *address,
                               BusContextFlags flags,
                               DBusError *error)
{
    DBusString log_prefix;
    DBusList *link;
    DBusList **addresses;
    const char *user, *pidfile;

```

```

char **auth_mechanisms;
DBusList **auth_mechanisms_list;
int len;
dbus_bool_t retval;

_DBUS_ASSERT_ERROR_IS_CLEAR (error);

retval = FALSE;
auth_mechanisms = NULL;
pidfile = NULL;

_dbus_init_system_log ();

if (flags & BUS_CONTEXT_FLAG_SYSTEMD_ACTIVATION)
    context->systemd_activation = TRUE;
else
    context->systemd_activation = FALSE;

/* Check for an existing pid file. Of course this is a race;
 * we'd have to use fcntl() locks on the pid file to
 * avoid that. But we want to check for the pid file
 * before overwriting any existing sockets, etc.
 */

if (flags & BUS_CONTEXT_FLAG_WRITE_PID_FILE)
    pidfile = bus_config_parser_get_pidfile (parser);

if (pidfile != NULL)
{
    DBusString u;
    DBusStat stbuf;

    _dbus_string_init_const (&u, pidfile);

    if (_dbus_stat (&u, &stbuf, NULL))
    {
#ifdef DBUS_CYGWIN
        DBusString p;
        long /* int */ pid;

        _dbus_string_init (&p);
        _dbus_file_get_contents(&p, &u, NULL);
        _dbus_string_parse_int(&p, 0, &pid, NULL);
        _dbus_string_free(&p);

        if ((kill((int)pid, 0)) {
            dbus_set_error(NULL, DBUS_ERROR_FILE_EXISTS,
                "pid %ld not running, removing stale pid
file\n",
                pid);
            _dbus_delete_file(&u, NULL);
        } else {

```

```

#endif
        dbus_set_error (error, DBUS_ERROR_FAILED,
                        "The pid file \"%s\" exists, if the
message bus is not running, remove this file",
                        pidfile);
        goto failed;
#ifdef DBUS_CYGWIN
    }
#endif
}
}

/* keep around the pid filename so we can delete it later */
context->pidfile = _dbus_strdup (pidfile);

/* note that type may be NULL */
context->type = _dbus_strdup (bus_config_parser_get_type (parser));
if (bus_config_parser_get_type (parser) != NULL && context->type ==
NULL)
    goto oom;

user = bus_config_parser_get_user (parser);
if (user != NULL)
{
    context->user = _dbus_strdup (user);
    if (context->user == NULL)
        goto oom;
}

/* Set up the prefix for syslog messages */
if (!_dbus_string_init (&log_prefix))
    goto oom;
if (context->type && !strcmp (context->type, "system"))
{
    if (!_dbus_string_append (&log_prefix, "[system] "))
        goto oom;
}
else if (context->type && !strcmp (context->type, "session"))
{
    DBusCredentials *credentials;

    credentials = _dbus_credentials_new_from_current_process ();
    if (!credentials)
        goto oom;
    if (!_dbus_string_append (&log_prefix, "[session] "))
    {
        _dbus_credentials_unref (credentials);
        goto oom;
    }
    if (!_dbus_credentials_to_string_append (credentials,
&log_prefix))
    {

```

```

        _dbus_credentials_unref (credentials);
        goto oom;
    }
    if (!_dbus_string_append (&log_prefix, "] ")
        {
            _dbus_credentials_unref (credentials);
            goto oom;
        }
        _dbus_credentials_unref (credentials);
    }
    if (!_dbus_string_steal_data (&log_prefix, &context->log_prefix))
        goto oom;
    _dbus_string_free (&log_prefix);

/* Build an array of auth mechanisms */

auth_mechanisms_list = bus_config_parser_get_mechanisms (parser);
len = _dbus_list_get_length (auth_mechanisms_list);

if (len > 0)
    {
        int i;

        auth_mechanisms = dbus_new0 (char*, len + 1);
        if (auth_mechanisms == NULL)
            goto oom;

        i = 0;
        link = _dbus_list_get_first_link (auth_mechanisms_list);
        while (link != NULL)
            {
                auth_mechanisms[i] = _dbus_strdup (link->data);
                if (auth_mechanisms[i] == NULL)
                    goto oom;
                link = _dbus_list_get_next_link (auth_mechanisms_list,
link);
                i += 1;
            }
    }
else
    {
        auth_mechanisms = NULL;
    }

/* Listen on our addresses */

if (address)
    {
        DBusServer *server;

        server = dbus_server_listen
(_dbus_string_get_const_data(address), error);

```

```

    if (server == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }
    else if (!setup_server (context, server, auth_mechanisms,
error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }

    if (!_dbus_list_append (&context->servers, server))
        goto oom;
}
else
{
    addresses = bus_config_parser_get_addresses (parser);

    link = _dbus_list_get_first_link (addresses);
    while (link != NULL)
    {
        DBusServer *server;

        server = dbus_server_listen (link->data, error);
        if (server == NULL)
        {
            _DBUS_ASSERT_ERROR_IS_SET (error);
            goto failed;
        }
        else if (!setup_server (context, server, auth_mechanisms,
error))
        {
            _DBUS_ASSERT_ERROR_IS_SET (error);
            goto failed;
        }

        if (!_dbus_list_append (&context->servers, server))
            goto oom;

        link = _dbus_list_get_next_link (addresses, link);
    }
}

context->fork = bus_config_parser_get_fork (parser);
context->syslog = bus_config_parser_get_syslog (parser);
context->keep_umask = bus_config_parser_get_keep_umask (parser);
context->allow_anonymous = bus_config_parser_get_allow_anonymous
(parser);

_DBUS_ASSERT_ERROR_IS_CLEAR (error);
retval = TRUE;

```

```

failed:
    dbus_free_string_array (auth_mechanisms);
    return retval;

oom:
    BUS_SET_OOM (error);
    dbus_free_string_array (auth_mechanisms);
    return FALSE;
}

/* This code gets executed every time the config files
 * are parsed: both during BusContext construction
 * and on reloads. This function is slightly screwy
 * since it can do a "half reload" in out-of-memory
 * situations. Realistically, unlikely to ever matter.
 */
static dbus_bool_t
process_config_every_time (BusContext      *context,
                           BusConfigParser *parser,
                           dbus_bool_t     is_reload,
                           DBusError      *error)
{
    DBusString full_address;
    DBusList *link;
    DBusList **dirs;
    char *addr;
    const char *servicehelper;
    char *s;

    dbus_bool_t retval;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    addr = NULL;
    retval = FALSE;

    if (!_dbus_string_init (&full_address))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    /* get our limits and timeout lengths */
    bus_config_parser_get_limits (parser, &context->limits);

    if (context->policy)
        bus_policy_unref (context->policy);
    context->policy = bus_config_parser_steal_policy (parser);
    _dbus_assert (context->policy != NULL);

    /* We have to build the address backward, so that

```

```

* <listen> later in the config file have priority
*/
link = _dbus_list_get_last_link (&context->servers);
while (link != NULL)
{
    addr = dbus_server_get_address (link->data);
    if (addr == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    if (_dbus_string_get_length (&full_address) > 0)
    {
        if (!_dbus_string_append (&full_address, ";"))
        {
            BUS_SET_OOM (error);
            goto failed;
        }
    }

    if (!_dbus_string_append (&full_address, addr))
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    dbus_free (addr);
    addr = NULL;

    link = _dbus_list_get_prev_link (&context->servers, link);
}

if (is_reload)
    dbus_free (context->address);

if (!_dbus_string_copy_data (&full_address, &context->address))
{
    BUS_SET_OOM (error);
    goto failed;
}

/* get the service directories */
dirs = bus_config_parser_get_service_dirs (parser);

/* and the service helper */
servicehelper = bus_config_parser_get_servicehelper (parser);

s = _dbus_strdup(servicehelper);
if (s == NULL && servicehelper != NULL)
{
    BUS_SET_OOM (error);
}

```



```

        goto failed;
    }
else
    {
        dbus_free(context->servicehelper);
        context->servicehelper = s;
    }

/* Create activation subsystem */
if (context->activation)
    {
        if (!bus_activation_reload (context->activation, &full_address,
dirs, error))
            goto failed;
    }
else
    {
        context->activation = bus_activation_new (context,
&full_address, dirs, error);
    }

if (context->activation == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }

_DBUS_ASSERT_ERROR_IS_CLEAR (error);
retval = TRUE;

failed:
    _dbus_string_free (&full_address);

if (addr)
    dbus_free (addr);

return retval;
}

static dbus_bool_t
list_concat_new (DBusList **a,
                DBusList **b,
                DBusList **result)
{
    DBusList *link;

    *result = NULL;

    for (link = _dbus_list_get_first_link (a); link; link =
_dbus_list_get_next_link (a, link))
        {
            if (!_dbus_list_append (result, link->data))

```

```

        goto oom;
    }
    for (link = _dbus_list_get_first_link (b); link; link =
        _dbus_list_get_next_link (b, link))
    {
        if (!_dbus_list_append (result, link->data))
            goto oom;
    }

    return TRUE;
oom:
    _dbus_list_clear (result);
    return FALSE;
}

static void
raise_file_descriptor_limit (BusContext      *context)
{
    /* I just picked this out of thin air; we need some extra
     * descriptors for things like any internal pipes we create,
     * inotify, connections to SELinux, etc.
     */
    unsigned int arbitrary_extra_fds = 32;
    unsigned int limit;

    limit = context->limits.max_completed_connections +
        context->limits.max_incomplete_connections
        + arbitrary_extra_fds;

    _dbus_request_file_descriptor_limit (limit);
}

static dbus_bool_t
process_config_postinit (BusContext      *context,
                        BusConfigParser *parser,
                        DBusError       *error)
{
    DBusHashTable *service_context_table;
    DBusList *watched_dirs = NULL;

    raise_file_descriptor_limit (context);

    service_context_table =
bus_config_parser_steal_service_context_table (parser);
    if (!_bus_registry_set_service_context_table (context->registry,
                                                service_context_table))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }
}

```

```

_dbus_hash_table_unref (service_context_table);

/* We need to monitor both the configuration directories and
directories
* containing .service files.
*/
if (!list_concat_new (bus_config_parser_get_conf_dirs (parser),
                    bus_config_parser_get_service_dirs (parser),
                    &watched_dirs))
{
    BUS_SET_OOM (error);
    return FALSE;
}

bus_set_watched_dirs (context, &watched_dirs);

_dbus_list_clear (&watched_dirs);

return TRUE;
}

BusContext*
bus_context_new (const DBusString *config_file,
                BusContextFlags  flags,
                DBusPipe         *print_addr_pipe,
                DBusPipe         *print_pid_pipe,
                const DBusString *address,
                DBusError        *error)
{
    BusContext *context;
    BusConfigParser *parser;

    _dbus_assert ((flags & BUS_CONTEXT_FLAG_FORK_NEVER) == 0 ||
                 (flags & BUS_CONTEXT_FLAG_FORK_ALWAYS) == 0);

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    context = NULL;
    parser = NULL;

    if (!dbus_server_allocate_data_slot (&server_data_slot))
    {
        BUS_SET_OOM (error);
        return NULL;
    }

    context = dbus_new0 (BusContext, 1);
    if (context == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }
}

```

```

context->refcount = 1;

_dbus_generate_uuid (&context->uuid);

if (!_dbus_string_copy_data (config_file, &context->config_file))
{
    BUS_SET_OOM (error);
    goto failed;
}

context->loop = _dbus_loop_new ();
if (context->loop == NULL)
{
    BUS_SET_OOM (error);
    goto failed;
}

context->registry = bus_registry_new (context);
if (context->registry == NULL)
{
    BUS_SET_OOM (error);
    goto failed;
}

parser = bus_config_load (config_file, TRUE, NULL, error);
if (parser == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    goto failed;
}

if (!process_config_first_time_only (context, parser, address,
flags, error))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    goto failed;
}

if (!process_config_every_time (context, parser, FALSE, error))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    goto failed;
}

/* we need another ref of the server data slot for the context
 * to own
 */
if (!_dbus_server_allocate_data_slot (&server_data_slot))
    _dbus_assert_not_reached ("second ref of server data slot
failed");

/* Note that we don't know whether the print_addr_pipe is
 * one of the sockets we're using to listen on, or some

```

```

* other random thing. But I think the answer is "don't do
* that then"
*/
if (print_addr_pipe != NULL && _dbus_pipe_is_valid
(print_addr_pipe))
{
    DBusString addr;
    const char *a = bus_context_get_address (context);
    int bytes;

    _dbus_assert (a != NULL);
    if (!_dbus_string_init (&addr))
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    if (!_dbus_string_append (&addr, a) ||
        !_dbus_string_append (&addr, "\n"))
    {
        _dbus_string_free (&addr);
        BUS_SET_OOM (error);
        goto failed;
    }

    bytes = _dbus_string_get_length (&addr);
    if (_dbus_pipe_write (print_addr_pipe, &addr, 0, bytes, error)
!= bytes)
    {
        /* pipe write returns an error on failure but not short
write */
        if (error != NULL && !dbus_error_is_set (error))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "Printing message bus address: did not
write all bytes\n");
        }
        _dbus_string_free (&addr);
        goto failed;
    }

    if (!_dbus_pipe_is_stdout_or_stderr (print_addr_pipe))
        _dbus_pipe_close (print_addr_pipe, NULL);

    _dbus_string_free (&addr);
}

context->connections = bus_connections_new (context);
if (context->connections == NULL)
{
    BUS_SET_OOM (error);
    goto failed;
}

```

```

    }

    context->matchmaker = bus_matchmaker_new ();
    if (context->matchmaker == NULL)
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    /* check user before we fork */
    if (context->user != NULL)
    {
        if (!_dbus_verify_daemon_user (context->user))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "Could not get UID and GID for username
\\\"%s\\\" ",
                context->user);
            goto failed;
        }
    }

    /* Now become a daemon if appropriate and write out pid file in any
case */
    {
        DBusString u;

        if (context->pidfile)
            _dbus_string_init_const (&u, context->pidfile);

        if (((flags & BUS_CONTEXT_FLAG_FORK_NEVER) == 0 && context->fork)
||
            (flags & BUS_CONTEXT_FLAG_FORK_ALWAYS))
        {
            _dbus_verbose ("Forking and becoming daemon\n");

            if (!_dbus_become_daemon (context->pidfile ? &u : NULL,
                print_pid_pipe,
                error,
                context->keep_umask))
            {
                _DBUS_ASSERT_ERROR_IS_SET (error);
                goto failed;
            }
        }
    }
    else
    {
        _dbus_verbose ("Fork not requested\n");

        /* Need to write PID file and to PID pipe for ourselves,
        * not for the child process. This is a no-op if the pidfile
        * is NULL and print_pid_pipe is NULL.

```

```

        */
        if (!_dbus_write_pid_to_file_and_pipe (context->pidfile ? &u :
NULL,
                                           print_pid_pipe,
                                           _dbus_getpid (),
                                           error))
        {
            _DBUS_ASSERT_ERROR_IS_SET (error);
            goto failed;
        }
    }
}

if (print_pid_pipe && _dbus_pipe_is_valid (print_pid_pipe) &&
    !_dbus_pipe_is_stdout_or_stderr (print_pid_pipe))
    _dbus_pipe_close (print_pid_pipe, NULL);

if (!bus_selinux_full_init ())
{
    bus_context_log (context, DBUS_SYSTEM_LOG_FATAL, "SELinux
enabled but AVC initialization failed; check system log\n");
}

if (!process_config_postinit (context, parser, error))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    goto failed;
}

if (parser != NULL)
{
    bus_config_parser_unref (parser);
    parser = NULL;
}

/* Here we change our credentials if required,
 * as soon as we've set up our sockets and pidfile
 */
if (context->user != NULL)
{
    if (!_dbus_change_to_daemon_user (context->user, error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }
}

#ifdef HAVE_SELINUX
    /* FIXME - why not just put this in full_init() below? */
    bus_selinux_audit_init ();
#endif
}

```

```

dbus_server_free_data_slot (&server_data_slot);

return context;

failed:
if (parser != NULL)
    bus_config_parser_unref (parser);
if (context != NULL)
    bus_context_unref (context);

if (server_data_slot >= 0)
    dbus_server_free_data_slot (&server_data_slot);

return NULL;
}

dbus_bool_t
bus_context_get_id (BusContext      *context,
                   DBusString      *uuid)
{
    return _dbus_uuid_encode (&context->uuid, uuid);
}

dbus_bool_t
bus_context_reload_config (BusContext *context,
                           DBusError  *error)
{
    BusConfigParser *parser;
    DBusString config_file;
    dbus_bool_t ret;

    /* Flush the user database cache */
    _dbus_flush_caches ();

    ret = FALSE;
    _dbus_string_init_const (&config_file, context->config_file);
    parser = bus_config_load (&config_file, TRUE, NULL, error);
    if (parser == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }

    if (!process_config_every_time (context, parser, TRUE, error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }
    if (!process_config_postinit (context, parser, error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed;
    }
}

```



```

    }
    ret = TRUE;

    bus_context_log (context, DBUS_SYSTEM_LOG_INFO, "Reloaded
configuration");
failed:
    if (!ret)
        bus_context_log (context, DBUS_SYSTEM_LOG_INFO, "Unable to reload
configuration: %s", error->message);
    if (parser != NULL)
        bus_config_parser_unref (parser);
    return ret;
}

static void
shutdown_server (BusContext *context,
                 DBusServer *server)
{
    if (server == NULL ||
        !dbus_server_get_is_connected (server))
        return;

    if (!dbus_server_set_watch_functions (server,
                                         NULL, NULL, NULL,
                                         context,
                                         NULL))
        _dbus_assert_not_reached ("setting watch functions to NULL
failed");

    if (!dbus_server_set_timeout_functions (server,
                                           NULL, NULL, NULL,
                                           context,
                                           NULL))
        _dbus_assert_not_reached ("setting timeout functions to NULL
failed");

    dbus_server_disconnect (server);
}

void
bus_context_shutdown (BusContext *context)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&context->servers);
    while (link != NULL)
    {
        shutdown_server (context, link->data);

        link = _dbus_list_get_next_link (&context->servers, link);
    }
}

```

```

BusContext *
bus_context_ref (BusContext *context)
{
    _dbus_assert (context->refcount > 0);
    context->refcount += 1;

    return context;
}

void
bus_context_unref (BusContext *context)
{
    _dbus_assert (context->refcount > 0);
    context->refcount -= 1;

    if (context->refcount == 0)
    {
        DBusList *link;

        _dbus_verbose ("Finalizing bus context %p\n", context);

        bus_context_shutdown (context);

        if (context->connections)
        {
            bus_connections_unref (context->connections);
            context->connections = NULL;
        }

        if (context->registry)
        {
            bus_registry_unref (context->registry);
            context->registry = NULL;
        }

        if (context->activation)
        {
            bus_activation_unref (context->activation);
            context->activation = NULL;
        }

        link = _dbus_list_get_first_link (&context->servers);
        while (link != NULL)
        {
            dbus_server_unref (link->data);

            link = _dbus_list_get_next_link (&context->servers, link);
        }
        _dbus_list_clear (&context->servers);

        if (context->policy)

```

```

    {
        bus_policy_unref (context->policy);
        context->policy = NULL;
    }

    if (context->loop)
    {
        _dbus_loop_unref (context->loop);
        context->loop = NULL;
    }

    if (context->matchmaker)
    {
        bus_matchmaker_unref (context->matchmaker);
        context->matchmaker = NULL;
    }

    dbus_free (context->config_file);
    dbus_free (context->log_prefix);
    dbus_free (context->type);
    dbus_free (context->address);
    dbus_free (context->user);
    dbus_free (context->servicehelper);

    if (context->pidfile)
    {
        DBusString u;
        _dbus_string_init_const (&u, context->pidfile);

        /* Deliberately ignore errors here, since there's not much
         * we can do about it, and we're exiting anyways.
         */
        _dbus_delete_file (&u, NULL);

        dbus_free (context->pidfile);
    }
    dbus_free (context);

    dbus_server_free_data_slot (&server_data_slot);
}

/* type may be NULL */
const char*
bus_context_get_type (BusContext *context)
{
    return context->type;
}

const char*
bus_context_get_address (BusContext *context)
{

```

```

    return context->address;
}

const char*
bus_context_get_servicehelper (BusContext *context)
{
    return context->servicehelper;
}

dbus_bool_t
bus_context_get_systemd_activation (BusContext *context)
{
    return context->systemd_activation;
}

BusRegistry*
bus_context_get_registry (BusContext *context)
{
    return context->registry;
}

BusConnections*
bus_context_get_connections (BusContext *context)
{
    return context->connections;
}

BusActivation*
bus_context_get_activation (BusContext *context)
{
    return context->activation;
}

BusMatchmaker*
bus_context_get_matchmaker (BusContext *context)
{
    return context->matchmaker;
}

DBusLoop*
bus_context_get_loop (BusContext *context)
{
    return context->loop;
}

dbus_bool_t
bus_context_allow_unix_user (BusContext *context,
                             unsigned long uid)
{
    return bus_policy_allow_unix_user (context->policy,
                                       uid);
}

```

```

/* For now this is never actually called because the default
 * DbusConnection behavior of 'same user that owns the bus can
connect'
 * is all it would do.
 */
dbus_bool_t
bus_context_allow_windows_user (BusContext      *context,
                               const char      *windows_sid)
{
    return bus_policy_allow_windows_user (context->policy,
                                         windows_sid);
}

BusPolicy *
bus_context_get_policy (BusContext *context)
{
    return context->policy;
}

BusClientPolicy*
bus_context_create_client_policy (BusContext      *context,
                                  DbusConnection *connection,
                                  DbusError       *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    return bus_policy_create_client_policy (context->policy, connection,
                                           error);
}

int
bus_context_get_activation_timeout (BusContext *context)
{
    return context->limits.activation_timeout;
}

int
bus_context_get_auth_timeout (BusContext *context)
{
    return context->limits.auth_timeout;
}

int
bus_context_get_max_completed_connections (BusContext *context)
{
    return context->limits.max_completed_connections;
}

int
bus_context_get_max_incomplete_connections (BusContext *context)
{

```

```

    return context->limits.max_incomplete_connections;
}

int
bus_context_get_max_connections_per_user (BusContext *context)
{
    return context->limits.max_connections_per_user;
}

int
bus_context_get_max_pending_activations (BusContext *context)
{
    return context->limits.max_pending_activations;
}

int
bus_context_get_max_services_per_connection (BusContext *context)
{
    return context->limits.max_services_per_connection;
}

int
bus_context_get_max_match_rules_per_connection (BusContext *context)
{
    return context->limits.max_match_rules_per_connection;
}

int
bus_context_get_max_replies_per_connection (BusContext *context)
{
    return context->limits.max_replies_per_connection;
}

int
bus_context_get_reply_timeout (BusContext *context)
{
    return context->limits.reply_timeout;
}

void
bus_context_log (BusContext *context, DBusSystemLogSeverity severity,
const char *msg, ...) _DBUS_GNUC_PRINTF (3, 4);

void
bus_context_log (BusContext *context, DBusSystemLogSeverity severity,
const char *msg, ...)
{
    va_list args;

    if (!context->syslog)
    {
        /* we're not syslogging; just output to stderr */

```

```

        va_start (args, msg);
        vfprintf (stderr, msg, args);
        fprintf (stderr, "\n");
        va_end (args);
        return;
    }

    va_start (args, msg);

    if (context->log_prefix)
    {
        DBusString full_msg;

        if (!_dbus_string_init (&full_msg))
            goto out;
        if (!_dbus_string_append (&full_msg, context->log_prefix))
            goto oom_out;
        if (!_dbus_string_append_printf_valist (&full_msg, msg, args))
            goto oom_out;

        _dbus_system_log (severity, "%s", _dbus_string_get_const_data
(&full_msg));
        oom_out:
        _dbus_string_free (&full_msg);
    }
    else
        _dbus_system_logv (severity, msg, args);

out:
    va_end (args);
}

static inline const char *
nonnull (const char *maybe_null,
        const char *if_null)
{
    return (maybe_null ? maybe_null : if_null);
}

/*
 * Log something about a message, usually that it was rejected.
 */
static void
complain_about_message (DBusContext      *context,
                       const char      *error_name,
                       const char      *complaint,
                       int               matched_rules,
                       DBusMessage      *message,
                       DBusConnection  *sender,
                       DBusConnection  *proposed_recipient,
                       dbus_bool_t      requested_reply,
                       dbus_bool_t      log,

```

```

                                DBusError      *error)
{
    DBusError stack_error = DBUS_ERROR_INIT;
    const char *sender_name;
    const char *sender_logininfo;
    const char *proposed_recipient_logininfo;

    if (error == NULL && !log)
        return;

    if (sender != NULL)
    {
        sender_name = bus_connection_get_name (sender);
        sender_logininfo = bus_connection_get_logininfo (sender);
    }
    else
    {
        sender_name = "(unset)";
        sender_logininfo = "(bus)";
    }

    if (proposed_recipient != NULL)
        proposed_recipient_logininfo = bus_connection_get_logininfo
(proposed_recipient);
    else
        proposed_recipient_logininfo = "bus";

    dbus_set_error (&stack_error, error_name,
        "%s, %d matched rules; type=\"%s\", sender=\"%s\" (%s) "
        "interface=\"%s\" member=\"%s\" error name=\"%s\" "
        "requested_reply=\"%d\" destination=\"%s\" (%s)",
        complaint,
        matched_rules,
        dbus_message_type_to_string (dbus_message_get_type (message)),
        sender_name,
        sender_logininfo,
        nonnull (dbus_message_get_interface (message), "(unset)"),
        nonnull (dbus_message_get_member (message), "(unset)"),
        nonnull (dbus_message_get_error_name (message), "(unset)"),
        requested_reply,
        nonnull (dbus_message_get_destination (message),
DBUS_SERVICE_DBUS),
        proposed_recipient_logininfo);

    /* If we hit OOM while setting the error, this will syslog "out of
memory"
* which is itself an indication that something is seriously wrong
*/
    if (log)
        bus_context_log (context, DBUS_SYSTEM_LOG_SECURITY, "%s",
            stack_error.message);
}

```



```

    dbus_move_error (&stack_error, error);
}

/*
 * addressed_recipient is the recipient specified in the message.
 *
 * proposed_recipient is the recipient we're considering sending
 * to right this second, and may be an eavesdropper.
 *
 * sender is the sender of the message.
 *
 * NULL for proposed_recipient or sender definitely means the bus
driver.
 *
 * NULL for addressed_recipient may mean the bus driver, or may mean
 * no destination was specified in the message (e.g. a signal).
 */
dbus_bool_t
bus_context_check_security_policy (BusContext      *context,
                                   BusTransaction *transaction,
                                   DBusConnection *sender,
                                   DBusConnection
*addressed_recipient,
                                   DBusConnection *proposed_recipient,
                                   DBusMessage    *message,
                                   DBusError      *error)
{
    const char *dest;
    BusClientPolicy *sender_policy;
    BusClientPolicy *recipient_policy;
    dbus_int32_t toggles;
    dbus_bool_t log;
    int type;
    dbus_bool_t requested_reply;

    type = dbus_message_get_type (message);
    dest = dbus_message_get_destination (message);

    /* dispatch.c was supposed to ensure these invariants */
    _dbus_assert (dest != NULL ||
                  type == DBUS_MESSAGE_TYPE_SIGNAL ||
                  (sender == NULL && !bus_connection_is_active
(proposed_recipient)));
    _dbus_assert (type == DBUS_MESSAGE_TYPE_SIGNAL ||
                  addressed_recipient != NULL ||
                  strcmp (dest, DBUS_SERVICE_DBUS) == 0);

    switch (type)
    {
        {
            case DBUS_MESSAGE_TYPE_METHOD_CALL:
            case DBUS_MESSAGE_TYPE_SIGNAL:
            case DBUS_MESSAGE_TYPE_METHOD_RETURN:

```

```

    case DBUS_MESSAGE_TYPE_ERROR:
        break;

    default:
        _dbus_verbose ("security check disallowing message of unknown
type %d\n",
                    type);

        dbus_set_error (error, DBUS_ERROR_ACCESS_DENIED,
                        "Message bus will not accept messages of unknown
type\n");

        return FALSE;
    }

    requested_reply = FALSE;

    if (sender != NULL)
    {
        /* First verify the SELinux access controls.  If allowed then
        * go on with the standard checks.
        */
        if (!bus_selinux_allows_send (sender, proposed_recipient,
dbus_message_type_to_string
(dbus_message_get_type (message)),
dbus_message_get_interface (message),
dbus_message_get_member (message),
dbus_message_get_error_name (message),
dest ? dest : DBUS_SERVICE_DBUS, error))
        {
            if (error != NULL && !dbus_error_is_set (error))
            {
                /* don't syslog this, just set the error: avc_has_perm
should
                * have already written to either the audit log or
syslog */
                complain_about_message (context,
DBUS_ERROR_ACCESS_DENIED,
                "An SELinux policy prevents this sender from sending
this "
                "message to this recipient",
                0, message, sender, proposed_recipient, FALSE,
FALSE, error);
                _dbus_verbose ("SELinux security check denying send to
service\n");
            }

            return FALSE;
        }

        if (bus_connection_is_active (sender))
        {

```

```

sender_policy = bus_connection_get_policy (sender);
_dbus_assert (sender_policy != NULL);

/* Fill in requested_reply variable with TRUE if this is a
 * reply and the reply was pending.
 */
if (dbus_message_get_reply_serial (message) != 0)
{
    if (proposed_recipient != NULL /* not to the bus driver
*/ &&
        addressed_recipient == proposed_recipient /* not
eavesdropping */)
    {
        DBusError error2;

        dbus_error_init (&error2);
        requested_reply = bus_connections_check_reply
(bus_connection_get_connections (sender),
transaction,
sender, addressed_recipient, message,
&error2);

        if (dbus_error_is_set (&error2))
        {
            dbus_move_error (&error2, error);
            return FALSE;
        }
    }
}
else
{
    /* Policy for inactive connections is that they can only
send
    * the hello message to the bus driver
    */
    if (proposed_recipient == NULL &&
        dbus_message_is_method_call (message,
DBUS_INTERFACE_DBUS,
"Hello"))
    {
        _dbus_verbose ("security check allowing %s message\n",
"Hello");
        return TRUE;
    }
    else
    {
        _dbus_verbose ("security check disallowing non-%s
message\n",
"Hello");

```

```

        dbus_set_error (error, DBUS_ERROR_ACCESS_DENIED,
                        "Client tried to send a message other
than %s without being registered",
                        "Hello");

        return FALSE;
    }
}
else
{
    sender_policy = NULL;

    /* If the sender is the bus driver, we assume any reply was a
    * requested reply as bus driver won't send bogus ones
    */
    if (addressed_recipient == proposed_recipient /* not
eavesdropping */ &&
        dbus_message_get_reply_serial (message) != 0)
        requested_reply = TRUE;
}

_dbus_assert ((sender != NULL && sender_policy != NULL) ||
              (sender == NULL && sender_policy == NULL));

if (proposed_recipient != NULL)
{
    /* only the bus driver can send to an inactive recipient (as it
    * owns no services, so other apps can't address it). Inactive
    * recipients can receive any message.
    */
    if (bus_connection_is_active (proposed_recipient))
    {
        recipient_policy = bus_connection_get_policy
(proposed_recipient);
        _dbus_assert (recipient_policy != NULL);
    }
    else if (sender == NULL)
    {
        _dbus_verbose ("security check using NULL recipient policy
for message from bus\n");
        recipient_policy = NULL;
    }
    else
    {
        _dbus_assert_not_reached ("a message was somehow sent to an
inactive recipient from a source other than the message bus\n");
        recipient_policy = NULL;
    }
}
else

```

```

    recipient_policy = NULL;

    _dbus_assert ((proposed_recipient != NULL && recipient_policy !=
NULL) ||
                (proposed_recipient != NULL && sender == NULL &&
recipient_policy == NULL) ||
                (proposed_recipient == NULL && recipient_policy ==
NULL));

    log = FALSE;
    if (sender_policy &&
        !bus_client_policy_check_can_send (sender_policy,
                                           context->registry,
                                           requested_reply,
                                           proposed_recipient,
                                           message, &toggles, &log))
    {
        complain_about_message (context, DBUS_ERROR_ACCESS_DENIED,
                                "Rejected send message", toggles,
                                message, sender, proposed_recipient, requested_reply,
                                (addressed_recipient == proposed_recipient), error);
        _dbus_verbose ("security policy disallowing message due to
sender policy\n");
        return FALSE;
    }

    if (log)
    {
        /* We want to drop this message, and are only not doing so for
backwards
        * compatibility. */
        complain_about_message (context, DBUS_ERROR_ACCESS_DENIED,
                                "Would reject message", toggles,
                                message, sender, proposed_recipient, requested_reply,
                                TRUE, NULL);
    }

    if (recipient_policy &&
        !bus_client_policy_check_can_receive (recipient_policy,
                                              context->registry,
                                              requested_reply,
                                              sender,
                                              addressed_recipient,
proposed_recipient,
                                              message, &toggles))
    {
        complain_about_message (context, DBUS_ERROR_ACCESS_DENIED,
                                "Rejected receive message", toggles,
                                message, sender, proposed_recipient, requested_reply,
                                (addressed_recipient == proposed_recipient), NULL);
        _dbus_verbose ("security policy disallowing message due to
recipient policy\n");
    }

```

```

    return FALSE;
}

/* See if limits on size have been exceeded */
if (proposed_recipient &&
    ((dbus_connection_get_outgoing_size (proposed_recipient) >
context->limits.max_outgoing_bytes) ||
    (dbus_connection_get_outgoing_unix_fds (proposed_recipient) >
context->limits.max_outgoing_unix_fds)))
{
    complain_about_message (context, DBUS_ERROR_LIMITS_EXCEEDED,
        "Rejected: destination has a full message queue",
        0, message, sender, proposed_recipient, requested_reply,
TRUE,
        error);
    _dbus_verbose ("security policy disallowing message due to full
message queue\n");
    return FALSE;
}

/* Record that we will allow a reply here in the future (don't
 * bother if the recipient is the bus or this is an eavesdropping
 * connection). Only the addressed recipient may reply.
 */
if (type == DBUS_MESSAGE_TYPE_METHOD_CALL &&
    sender &&
    addressed_recipient &&
    addressed_recipient == proposed_recipient && /* not
eavesdropping */
    !bus_connections_expect_reply (bus_connection_get_connections
(sender),
                                transaction,
                                sender, addressed_recipient,
                                message, error))
{
    _dbus_verbose ("Failed to record reply expectation or problem
with the message expecting a reply\n");
    return FALSE;
}

_dbus_verbose ("security policy allowing message\n");
return TRUE;
}

```

File = bus.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* bus.h message bus context object
 *
 * Copyright (C) 2003 Red Hat, Inc.

```

```
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#ifndef BUS_BUS_H
#define BUS_BUS_H
```

```
#include <dbus/dbus.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-mainloop.h>
#include <dbus/dbus-pipe.h>
#include <dbus/dbus-sysdeps.h>
```

```
typedef struct BusActivation      BusActivation;
typedef struct BusConnections     BusConnections;
typedef struct BusContext        BusContext;
typedef struct BusPolicy         BusPolicy;
typedef struct BusClientPolicy   BusClientPolicy;
typedef struct BusPolicyRule     BusPolicyRule;
typedef struct BusRegistry       BusRegistry;
typedef struct BusSELinuxID      BusSELinuxID;
typedef struct BusService        BusService;
typedef struct BusOwner          BusOwner;
typedef struct BusTransaction    BusTransaction;
typedef struct BusMatchmaker     BusMatchmaker;
typedef struct BusMatchRule      BusMatchRule;
```

```
typedef struct
{
    long max_incoming_bytes;      /**< How many incoming message
bytes for a single connection */
    long max_incoming_unix_fds;  /**< How many incoming message
unix fds for a single connection */
```

```

    long max_outgoing_bytes;          /**< How many outgoing bytes can
be queued for a single connection */
    long max_outgoing_unix_fds;      /**< How many outgoing unix fds
can be queued for a single connection */
    long max_message_size;           /**< Max size of a single message
in bytes */
    long max_message_unix_fds;       /**< Max number of unix fds of a
single message*/
    int activation_timeout;           /**< How long to wait for an
activation to time out */
    int auth_timeout;                /**< How long to wait for an
authentication to time out */
    int max_completed_connections;    /**< Max number of authorized
connections */
    int max_incomplete_connections;  /**< Max number of incomplete
connections */
    int max_connections_per_user;     /**< Max number of connections
auth'd as same user */
    int max_pending_activations;      /**< Max number of pending
activations for the entire bus */
    int max_services_per_connection;  /**< Max number of owned services
for a single connection */
    int max_match_rules_per_connection; /**< Max number of match rules
for a single connection */
    int max_replies_per_connection;   /**< Max number of replies that
can be pending for each connection */
    int reply_timeout;                /**< How long to wait before
timing out a reply */
} BusLimits;

```

```

typedef enum
{
    BUS_CONTEXT_FLAG_NONE = 0,
    BUS_CONTEXT_FLAG_FORK_ALWAYS = (1 << 1),
    BUS_CONTEXT_FLAG_FORK_NEVER = (1 << 2),
    BUS_CONTEXT_FLAG_WRITE_PID_FILE = (1 << 3),
    BUS_CONTEXT_FLAG_SYSTEMD_ACTIVATION = (1 << 4)
} BusContextFlags;

```

```

DBusContext*      bus_context_new
(const DBusString *config_file,

DBusContextFlags  flags,

DBusPipe          *print_addr_pipe,

DBusPipe          *print_pid_pipe,

const DBusString *address,

DBusError         *error);

```



```

dbus_bool_t      bus_context_reload_config
(BusContext     *context,

                                                         DBusError     *error);

void            bus_context_shutdown
(BusContext     *context);
BusContext*    bus_context_ref
(BusContext     *context);
void           bus_context_unref
(BusContext     *context);
dbus_bool_t    bus_context_get_id
(BusContext     *context,

DBusString     *uuid);
const char*    bus_context_get_type
(BusContext     *context);
const char*    bus_context_get_address
(BusContext     *context);
const char*    bus_context_get_servicehelper
(BusContext     *context);
dbus_bool_t    bus_context_get_systemd_activation
(BusContext     *context);
BusRegistry*   bus_context_get_registry
(BusContext     *context);
BusConnections* bus_context_get_connections
(BusContext     *context);
BusActivation* bus_context_get_activation
(BusContext     *context);
BusMatchmaker* bus_context_get_matchmaker
(BusContext     *context);
DBusLoop*     bus_context_get_loop
(BusContext     *context);
dbus_bool_t    bus_context_allow_unix_user
(BusContext     *context,

unsigned long   uid);
dbus_bool_t    bus_context_allow_windows_user
(BusContext     *context,

const char     *windows_sid);
BusPolicy*     bus_context_get_policy
(BusContext     *context);

BusClientPolicy* bus_context_create_client_policy
(BusContext     *context,

DBusConnection *connection,

DBusError     *error);
int           bus_context_get_activation_timeout
(BusContext     *context);
int           bus_context_get_auth_timeout
(BusContext     *context);

```

```

int          bus_context_get_max_completed_connections
  (BusContext *context);
int          bus_context_get_max_incomplete_connections
  (BusContext *context);
int          bus_context_get_max_connections_per_user
  (BusContext *context);
int          bus_context_get_max_pending_activations
  (BusContext *context);
int          bus_context_get_max_services_per_connection
  (BusContext *context);
int          bus_context_get_max_match_rules_per_connection
  (BusContext *context);
int          bus_context_get_max_replies_per_connection
  (BusContext *context);
int          bus_context_get_reply_timeout
  (BusContext *context);
void         bus_context_log
  (BusContext *context,

```

```

DBusSystemLogSeverity severity,

```

```

const char  *msg,

```

```

...);

```

```

dbus_bool_t bus_context_check_security_policy
  (BusContext *context,

```

```

BusTransaction *transaction,

```

```

DBusConnection *sender,

```

```

DBusConnection *addressed_recipient,

```

```

DBusConnection *proposed_recipient,

```

```

DBusMessage *message,

```

```

DBusError *error);

```

```

#endif /* BUS_BUS_H */

```

```

File = busconfig.dtd

```

```

<!ELEMENT busconfig (user |
                    type |
                    fork |
                    keep_umask |
                    listen |
                    pidfile |
                    includedir |

```

```

        servicedir |
        servicehelper |
        auth |
        include |
        policy |
        limit |
        selinux)*>

<!ELEMENT user (#PCDATA)>
<!ELEMENT listen (#PCDATA)>
<!ELEMENT includedir (#PCDATA)>
<!ELEMENT servicedir (#PCDATA)>
<!ELEMENT servicehelper (#PCDATA)>
<!ELEMENT auth (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT pidfile (#PCDATA)>
<!ELEMENT fork EMPTY>
<!ELEMENT keep_umask EMPTY>

<!ELEMENT include (#PCDATA)>
<!ATTLIST include
    ignore_missing (yes|no) "no"
    if_selinux_enabled (yes|no) "no"
    selinux_root_relative (yes|no) "no">

<!ELEMENT policy (allow|deny)*>
<!ATTLIST policy
    context (default|mandatory) #IMPLIED
    user CDATA #IMPLIED
    group CDATA #IMPLIED
    at_console (yes|no) #IMPLIED>

<!ELEMENT allow EMPTY>
<!ATTLIST allow
    user CDATA #IMPLIED
    send CDATA #IMPLIED
    receive CDATA #IMPLIED
    own CDATA #IMPLIED
    send_to CDATA #IMPLIED
    receive_from CDATA #IMPLIED>

<!ELEMENT deny EMPTY>
<!ATTLIST deny
    user CDATA #IMPLIED
    send CDATA #IMPLIED
    receive CDATA #IMPLIED
    own CDATA #IMPLIED
    send_to CDATA #IMPLIED
    receive_from CDATA #IMPLIED>

<!ELEMENT limit (#PCDATA)>
<!ATTLIST limit name CDATA #REQUIRED>

```

```
<!ELEMENT selinux (associate)*>
<!ELEMENT associate EMPTY>
<!ATTLIST associate
    own CDATA #REQUIRED
    context CDATA #REQUIRED>
```

File = byte-hashes.shal

```
# Configuration information for "SHA-1 Test"
# SHA tests are configured for BYTE oriented implementations
H>SHS Type 1 Hashes<H
D>
DA39A3EE5E6B4B0D3255BFEEF95601890AFD80709 ^
3CDF2936DA2FC556BFA533AB1EB59CE710AC80E5 ^
19C1E2048FA7393CFBF2D310AD8209EC11D996E5 ^
CA775D8C80FAA6F87FA62BECA6CA6089D63B56E5 ^
71AC973D0E4B50AE9E5043FF4D615381120A25A0 ^
A6B5B9F854CFB76701C3BDDBF374B3094EA49CBA ^
D87A0EE74E4B9AD72E6847C87BDEEB3D07844380 ^
1976B8DD509FE66BF09C9A8D33534D4EF4F63BFD ^
5A78F439B6DB845BB8A558E4CEB106CD7B7FF783 ^
F871BCE62436C1E280357416695EE2EF9B83695C ^
62B243D1B780E1D31CF1BA2DE3F01C72AEEA0E47 ^
1698994A273404848E56E7FDA4457B5900DE1342 ^
056F4CDC02791DA7ED1EB2303314F7667518DEEF ^
9FE2DA967BD8441EEA1C32DF68DDAA9DC1FC8E4B ^
73A31777B4ACE9384EFA8BBEAD45C51A71ABA6DD ^
3F9D7C4E2384EDDABFF5DD8A31E23DE3D03F42AC ^
4814908F72B93FFD011135BEE347DE9A08DA838F ^
0978374B67A412A3102C5AA0B10E1A6596FC68EB ^
44AD6CB618BD935460D46D3F921D87B99AB91C1E ^
02DC989AF265B09CF8485640842128DCF95E9F39 ^
67507B8D497B35D6E99FC01976D73F54AECA75CF ^
1EAE0373C1317CB60C36A42A867B716039D441F5 ^
9C3834589E5BFFAC9F50950E0199B3EC2620BEC8 ^
209F7ABC7F3B878EE46CDF3A1FBB9C21C3474F32 ^
05FC054B00D97753A9B3E2DA8FBBA3EE808CEF22 ^
0C4980EA3A46C757DFBFC5BAA38AC6C8E72DDCE7 ^
96A460D2972D276928B69864445BEA353BDCFFD2 ^
F3EF04D8FA8C6FA9850F394A4554C080956FA64B ^
F2A31D875D1D7B30874D416C4D2EA6BAF0FFBAFE ^
F4942D3B9E9588DCFDC6312A84DF75D05F111C20 ^
310207DF35B014E4676D30806FA34424813734DD ^
4DA1955B2FA7C7E74E3F47D7360CE530BBF57CA3 ^
74C4BC5B26FB4A08602D40CCEC6C6161B6C11478 ^
0B103CE297338DFC7395F7715EE47539B556DDB6 ^
EFC72D99E3D2311CE14190C0B726BDC68F4B0821 ^
660EDAC0A8F4CE33DA0D8DBAE597650E97687250 ^
FE0A55A988B3B93946A63EB36B23785A5E6EFC3E ^
```

0CBDF2A5781C59F907513147A0DE3CC774B54BF3 ^
663E40FEE5A44BFCB1C99EA5935A6B5BC9F583B0 ^
00162134256952DD9AE6B51EFB159B35C3C138C7 ^
CEB88E4736E354416E2010FC1061B3B53B81664B ^
A6A2C4B6BCC41DDC67278F3DF4D8D0B9DD7784EF ^
C23D083CD8820B57800A869F5F261D45E02DC55D ^
E8AC31927B78DDEC41A31CA7A44EB7177165E7AB ^
E864EC5DBAB0F9FF6984AB6AD43A8C9B81CC9F9C ^
CFED6269069417A84D6DE2347220F4B858BCD530 ^
D9217BFB46C96348722C3783D29D4B1A3FEDA38C ^
DEC24E5554F79697218D317315FA986229CE3350 ^
83A099DF7071437BA5495A5B0BFBFEBE1C0EF7F3 ^
AA3198E30891A83E33CE3BFA0587D86A197D4F80 ^
9B6ACBEB4989CBEE7015C7D515A75672FFDE3442 ^
B021EB08A436B02658EAA7BA3C88D49F1219C035 ^
CAE36DAB8AEA29F62E0855D9CB3CD8E7D39094B1 ^
02DE8BA699F3C1B0CB5AD89A01F2346E630459D7 ^
88021458847DD39B4495368F7254941859FAD44B ^
91A165295C666FE85C2ADBC5A10329DAF0CB81A0 ^
4B31312EAF8B506811151A9DBD162961F7548C4B ^
3FE70971B20558F7E9BAC303ED2BC14BDE659A62 ^
93FB769D5BF49D6C563685954E2AECC024DC02D6 ^
BC8827C3E614D515E83DEA503989DEA4FDA6EA13 ^
E83868DBE4A389AB48E61CFC4ED894F32AE112AC ^
55C95459CDE4B33791B4B2BCAAF840930AF3F3BD ^
36BB0E2BA438A3E03214D9ED2B28A4D5C578FCAA ^
3ACBF874199763EBA20F3789DFC59572ACA4CF33 ^
86BE037C4D509C9202020767D860DAB039CADACE ^
51B57D7080A87394EEC3EB2E0B242E553F2827C9 ^
1EFBFA78866315CE6A71E457F3A750A38FACAB41 ^
57D6CB41AECC20236F365B3A490C61D0CFA39611 ^
C532CB64B4BA826372BCCF2B4B5793D5B88BB715 ^
15833B5631032663E783686A209C6A2B47A1080E ^
D04F2043C96E10CD83B574B1E1C217052CD4A6B2 ^
E8882627C64DB743F7DB8B4413DD033FC63BEB20 ^
CD2D32286B8867BC124A0AF2236FC74BE3622199 ^
019B70D745375091ED5C7B218445EC986D0F5A82 ^
E5FF5FEC1DADBAED02BF2DAD4026BE6A96B3F2AF ^
6F4E23B3F2E2C068D13921FE4E5E053FFED4E146 ^
25E179602A575C915067566FBA6DA930E97F8678 ^
67DED0E68E235C8A523E051E86108EEB757EFBFD ^
AF78536EA83C822796745556D62A3EE82C7BE098 ^
64D7AC52E47834BE72455F6C64325F9C358B610D ^
9D4866BAA3639C13E541F250FFA3D8BC157A491F ^
2E258811961D3EB876F30E7019241A01F9517BEC ^
8E0EBC487146F83BC9077A1630E0FB3AB3C89E63 ^
CE8953741FFF3425D2311FBBF4AB481B669DEF70 ^
789D1D2DAB52086BD90C0E137E2515ED9C6B59B5 ^
B76CE7472700DD68D6328B7AA8437FB051D15745 ^
F218669B596C5FFB0B1C14BD03C467FC873230A0 ^
1FF3BDBE0D504CB0CDFAB17E6C37ABA6B3CFFDED ^
2F3CBACBB14405A4652ED52793C1814FD8C4FCE0 ^

982C8AB6CE164F481915AF59AAED9FFF2A391752 ^
5CD92012D488A07ECE0E47901D0E083B6BD93E3F ^
69603FEC02920851D4B3B8782E07B92BB2963009 ^
3E90F76437B1EA44CF98A08D83EA24CECF6E6191 ^
34C09F107C42D990EB4881D4BF2DDDCAB01563AE ^
474BE0E5892EB2382109BFC5E3C8249A9283B03D ^
A04B4F75051786682483252438F6A75BF4705EC6 ^
BE88A6716083EB50ED9416719D6A247661299383 ^
C67E38717FEE1A5F65EC6C7C7C42AFC00CD37F04 ^
959AC4082388E19E9BE5DE571C047EF10C174A8D ^
BAA7AA7B7753FA0ABDC4A541842B5D238D949F0A ^
351394DCEBC08155D100FCD488578E6AE71D0E9C ^
AB8BE94C5AF60D9477EF1252D604E58E27B2A9EE ^
3429EC74A695FDD3228F152564952308AFE0680A ^
907FA46C029BC67EAA8E4F46E3C2A232F85BD122 ^
2644C87D1FBBBC0FC8D65F64BCA2492DA15BAAE4 ^
110A3EEB408756E2E81ABAF4C5DCD4D4C6AFCF6D ^
CD4FDC35FAC7E1ADB5DE40F47F256EF74D584959 ^
8E6E273208AC256F9ECCF296F3F5A37BC8A0F9F7 ^
FE0606100BDBC268DB39B503E0FDFE3766185828 ^
6C63C3E58047BCDB35A17F74EEBA4E9B14420809 ^
BCC2BD305F0BCDA8CF2D478EF9FE080486CB265F ^
CE5223FD3DD920A3B666481D5625B16457DCB5E8 ^
948886776E42E4F5FAE1B2D0C906AC3759E3F8B0 ^
4C12A51FCFE242F832E3D7329304B11B75161EFB ^
C54BDD2050504D92F551D378AD5FC72C9ED03932 ^
8F53E8FA79EA09FD1B682AF5ED1515ECA965604C ^
2D7E17F6294524CE78B33EAB72CDD08E5FF6E313 ^
64582B4B57F782C9302BFE7D07F74AA176627A3A ^
6D88795B71D3E386BBD1EB830FB9F161BA98869F ^
86AD34A6463F12CEE6DE9596ABA72F0DF1397FD1 ^
7EB46685A57C0D466152DC339C8122548C757ED1 ^
E7A98FB0692684054407CC221ABC60C199D6F52A ^
34DF1306662206FD0A5FC2969A4BEEC4EB0197F7 ^
56CF7EBF08D10F0CB9FE7EE3B63A5C3A02BCB450 ^
3BAE5CB8226642088DA760A6F78B0CF8EDDEA9F1 ^
6475DF681E061FA506672C27CBABFA9AA6DDFF62 ^
79D81991FA4E4957C8062753439DBFD47BBB277D ^
BAE224477B20302E881F5249F52EC6C34DA8ECE ^
EDE4DEB4293CFE4138C2C056B7C46FF821CC0ACC ^
<D

H>SHS Type 2 Hashes<H

D>

A771FA5C812BD0C9596D869EC99E4F4AC988B13F ^
E99D566212BBBCEEE903946F6100C9C96039A8F4 ^
B48CE6B1D13903E3925AE0C88CB931388C013F9C ^
E647D5BAF670D4BF3AFC0A6B72A2424B0C64F194 ^
65C1CD932A06B05CD0B43AFB3BC7891F6BCEF45C ^
70FFAE353A5CD0F8A65A8B2746D0F16281B25EC7 ^
CC8221F2B829B8CF39646BF46888317C3EB378EA ^
26ACCC2D6D51FF7BF3E5895588907765111BB69B ^

01072915B8E868D9B28E759CF2BC1AEA4BB92165 ^
3016115711D74236ADF0C371E47992F87A428598 ^
BF30417999C1368F008C1F19FECA4D18A5E1C3C9 ^
62BA49087185F2742C26E1C1F4844112178BF673 ^
E1F6B9536F384DD3098285BBFD495A474140DC5A ^
B522DAE1D67726EBA7C4136D4E2F6D6D645AC43E ^
E9A021C3EB0B9F2C710554D4BF21B19F78E09478 ^
DF13573188F3BF705E697A3E1F580145F2183377 ^
188835CFE52ECFA0C4135C2825F245DC29973970 ^
41B615A34EE2CEC9D84A91B141CFAB115821950B ^
AB3DD6221D2AFE6613B815DA1C389EEC74AA0337 ^
0706D414B4AA7FB4A9051AA70D6856A7264054FB ^
3CBF8151F3A00B1D5A809CBB8C4F3135055A6BD1 ^
DA5D6A0319272BCCCEA63ACFA6799756FFDA6840 ^
FB4429C95F6277B346D3B389413758DFFFEEDC98 ^
2C6E30D9C895B42DCCCFC84C906EC88C09B20DE1 ^
3DE3189A5E19F225CDCE254DFF23DACD22C61363 ^
93530A9BC9A817F6922518A73A1505C411D05DA2 ^
E31354345F832D31E05C1B842D405D4BD4588EC8 ^
3FF76957E80B60CF74D015AD431FCA147B3AF232 ^
34AE3B806BE143A84DCE82E4B830EB7D3D2BAC69 ^
D7447E53D66BB5E4C26E8B41F83EFD107BF4ADDA ^
77DD2A4482705BC2E9DC96EC0A13395771AC850C ^
EAA1465DB1F59DE3F25EB8629602B568E693BB57 ^
9329D5B40E0DC43AA25FED69A0FA9C211A948411 ^
E94C0B6AA62AA08C625FAF817DDF8F51EC645273 ^
7FF02B909D82AD668E31E547E0FB66CB8E213771 ^
5BB3570858FA1744123BAC2873B0BB9810F53FA1 ^
905F43940B3591CE39D1145ACB1ECA80AB5E43CD ^
336C79FBD82F33E490C577E3F791C3CBFE842AFF ^
5C6D07A6B44F7A75A64F6CE592F3BAE91E022210 ^
7E0D3E9D33127F4A30EB8D9C134A58409FA8695B ^
9A5F50DFCFB19286206C229019F0ABF25283028C ^
DCA737E269F9D8626D488988C996E06B352C0708 ^
B8FFC1D4972FCE63241E0E77850AC46DDE75DBFA ^
E9C9BF41C8549354151B977003CE1D830BE667DB ^
0942908960B54F96CB43452E583F4F9CB66E398A ^
FCE34051C34D4B81B85DDC4B543CDE8007E284B3 ^
61E8916532503627F4024D13884640A46F1D61D4 ^
F008D5D7853B6A17B7466CD9E18BD135E520FAF4 ^
BD8D2E873CF659B5C77AAC1616827EF8A3B1A3B3 ^
B25A04DD425302ED211A1C2412D2410FA10C63B6 ^
A404E21588123E0893718B4B44E91414A785B91F ^
A1E13BC55BF6DAD83CF3AABDA3287AD68681EA64 ^
D5FD35FFABED6733C92365929DF0FB4CAE864D15 ^
C12E9C280EE9C079E0506FF89F9B20536E0A83EF ^
E22769DC00748A9BBD6C05BBC8E81F2CD1DC4E2D ^
F29835A93475740E888E8C14318F3CA45A3C8606 ^
1A1D77C6D0F97C4B620FAA90F3F8644408E4B13D ^
4EC84870E9BDD25F523C6DFB6EDD605052CA4EAA ^
D689513FED08B80C39B67371959BC4E3FECB0537 ^
C4FED58F209FC3C34AD19F86A6DACADC86C04D33 ^

051888C6D00029C176DE792B84DECE2DC1C74B00 ^
1A3540BEE05518505827954F58B751C475AEECE0 ^
DFA19180359D5A7A38E842F172359CAF4208FC05 ^
7B0FA84EBBCFF7D7F4500F73D79660C4A3431B67 ^
9E886081C9ACAAD0F97B10810D1DE6FCDCE6B5F4 ^
A4D46E4BA0AE4B012F75B1B50D0534D578AE9CB6 ^
6342B199EE64C7B2C9CBCD4F2DCB65ACEF51516F ^
AABFD63688EB678357869130083E1B52F6EA861D ^
F732B7372DAF44801F81EFFE3108726239837936 ^
5E9347FE4574CDCB80281ED092191199BADD7B42 ^
D5776B7DFFF75C1358ABDBBB3F27A20BB6CA7C55 ^
022B7ADA472FB7A9DA9219621C9C5F563D3792F6 ^
7F1DE4ECA20362DA624653D225A5B3F7964A9FF2 ^
CA0F2B1BFB4469C11ED006A994734F0F2F5EFD17 ^
833D63F5C2EA0CD43EC15F2B9DD97FF12B030479 ^
14FD356190416C00592B86FF7CA50B622F85593A ^
4AB6B57EDDEF1CE935622F935C1619AE7C1667D6 ^
B456A6A968ACD66CAA974F96A9A916E700AA3C5D ^
FD1C257FE046B2A27E2F0CD55ED2DECA845F01D7 ^
66E0D01780F1063E2929EAAD74826BC64060E38C ^
A8478DF406F179FD4EF97F4574D7F99EA1CE9EB8 ^
248E58CF09A372114FC2F93B09C5FC14F3D0059E ^
F15767DE91796A6816977EFA4FCED4B7FD9B8A57 ^
36A6BC5E680E15675D9696338C88B36248BBBAF4 ^
4DEA6251B2A6DF017A8093AB066EE3863A4EC369 ^
D30E70E357D57E3D82CA554B8A3D58DFF528FA94 ^
70CA84D827F7FD61446233F88CF2F990B0F3E2AA ^
8D500C9CFDE0288530A2106B70BED39326C52C3C ^
F3D4D139EDFC24596377BC97A96FB7621F27FFC7 ^
5509BAFFAC6D507860CEFC5AB5832CB63CD4B687 ^
0C0AEA0C2FD7A620C77866B1A177481E26B4F592 ^
149176007FEE58A591E3F00F8DB658B605F8390C ^
17C0D7B0256159F3626786FFDB20237AE154FA84 ^
741A58618ABEB1D983D67AFDCBC49AA397A3B8E0 ^
B738D6B3409EB9ED2F1719B84D13F7C36169CDEC ^
3D33DE31F64055D3B128AC9A6AA3F92DFD4F5330 ^
B6925F4DF94949B8844C867428BA3DEDF4CF2B51 ^
CF5E7256292ABEC431D8E8B9CBEAF22AF072377E ^
975DCE94902923977F129C0E4ACF40AD28DDB9AA ^
333B0259B18CE64D6B52CF563DD3041E5F63A516 ^
<D

H>SHS Type 3 Hashes<H

D>

80E044703A880C20EC41F645120A8A5B5D194ECE ^
E142829CA08FC9787F17AA16CE727396169B2713 ^
6A2BAF62469D311F9257A0727F52C7EAA87CCEB4 ^
362E3E7136CA611D7FBF687D3BBDC54CDA64843F ^
F5900ADC6223A5D24A7526ABFC60FA8E2D59A5AB ^
AD0CAC6A21D5B10833DDE7FA85927D74EDA142A9 ^
47AD337EAFFFDC177AAF7CBD035BE6F398B9D0536 ^
9CF58595DF80872535BCC7C056E223546F0BB4EE ^

7151CEB1918278CED2902B1D663D596F8D1B986F ^
ADDC9F09AA4026EF6C4B7F1A84D3A13B4CDC65B3 ^
921FE78A863A317B1FA1FB3CA3BE1948DE7EF754 ^
64BE10732D71D52CE8A486DA23E6B453DF7C6FBD ^
4A450659470DD759ABFAE1D73972A6D2E63AC16C ^
0D665E4BBF30B7EAB955BDE84759E185EECAB4CB ^
0C1B8EE94D61CDD0837EAED9FE33DE4A8334B596 ^
D93BFE2A6227A4BF9B7C61EBCE4A8CDE131593FE ^
BDA883F804B470C90BD6AC490DFC34EBC27F9648 ^
46A0969373552213632591C52030C38E5DBDC49E ^
4781289E48B910C550DC23CA7D3AF5324C03532D ^
693A34CFCDDDED0F3AC72E7197FCE9BB66A8E3981 ^
AE088AF1D8865140963B3ABFB63E32E04CD1506F ^
ADF0F8F1D85CA97586F5DC6DC5FD11FA39270F55 ^
E484F5AD86C5F4D09E366ADF6E0DE73449F97B28 ^
81C49842BA3D7072FB42288E03CE737A2672C091 ^
F6CC71AD897C23A16835490DED289BFD45500AB0 ^
23E71AED62FE8E28F34F58E7FE5594EC5EB0486C ^
92BA7934AA5867EE52960F4E0EDFB90AA7B69305 ^
C3D1CC8CBD1B6FFEE0D90CE962CD9C09AB1548AA ^
3CE37A583B71A6A77BE325066A0F00C5D11DFC3E ^
76EF5D236E1042D356A3234A422C092F86003064 ^
8C3F703436C6C882E60263540A8E4C3E5646DC15 ^
6138F9F3AB43B988DD3857422CCB304352459F40 ^
B812DE98775B4690B4FC2ECFCAB61C73C7271DC7 ^
06660985CD80D48E7B9F88455B4233924C3B64BB ^
76AB4B6378D6F63499A94EB67EB1CB31AFF8D775 ^
F31F6B0BE7AB059A1F59A46481967E88392979E6 ^
0C1638498FBB7DB9600B98B4B22EF85E0FE245FB ^
5607C6AF600939736795AC523FA43B736F41A118 ^
8A03244866BDD21B9D8A82E98436C894FAD86ECC ^
8A75BFD911AF87303B9B8FB7A1A47CCA52D3D98A ^
16F0F3B5D37411236A1E3D6B1EDAB74CDA25ED4B ^
AC72BF45477481F58A302628DC5299FFA32E7C9F ^
74CFFD5881F75AC20726E1447DCF7F47024380EF ^
5BFBECEECBC27DA05729C4D1AC8C1286EA6DCEC9 ^
012AACBC0579FA4CB4F107E9A9AD1A86AD2F6A4D ^
F7D552CBC5EF90F1A579388B5A8A9EC71EB67681 ^
10C70115C4C34753274BFED477DF01440A67A361 ^
078D2FACD293B6B6219D89899C16AA1AA8E3DE82 ^
83C6BF9FB0D3091ADF374EBFA0A69916F17E6D26 ^
2CDB1924DA62AB64C007C6505FF657E4ADDEA9C1 ^
E95D209BCB9864B076FF4DFCA8F8BD75D62D1B48 ^
632824CF5025F8F90AD2923BDDF449550D64C0F5 ^
02B1C0B41FC27EC5A32E586F1AC480BF0061E56A ^
28156BC6769AE390BF32C6512C46169181E1536D ^
F730E6E287D992E7F3E013B6F1E088F0B9C41598 ^
B056A6A832FA5FE964EF77FF3E0BE1C32E0D58C0 ^
D5B3D19AFBB48FB56BA6D44A82DE6BD08DB208DE ^
0215AD79BD6B8023C05FD2F8966211897DF6337A ^
EC4CF38C244EB6526A44F70570925247145DA8CA ^
C0D931262ECE93DA5A6ABC89CD6AD3162EA6B09E ^

```
6BB48FAC26AA2B4859BBDEF0CFB53AE4D1D9A0340 ^
58611D43741E67A7F0DA9CB337A59DCD1EBE758E ^
7C2AEC216AF231509E47B7EED06BB17859812B7E ^
F60EE5DBF4A7A676EC98B3DDB1CDD6CDF3CDA33B ^
0492E59B1F4C94E97F29A26C3EE7D57E1B0FDD72 ^
4FCF549D902D9BE1101A756DB9E45415FB61BCD2 ^
95C71D26AD6B38CC771376B4A4F962F12E1E3D4F ^
F6A2449E773C72FB886B3C43E2B30EC2A1B7454A ^
CDE86695E00AEC9A5DB6FDDB5D5A5934448D58E0 ^
502318A758FABFF6AC53844E9E2BCD159C678510 ^
589D295148F95F75DAE964DD743FE981FA236D4E ^
7973DD33AE3599A556BACC77E8656E782E029EFF ^
9F5BE43AADD43C6DB3883C9DA4B52E1A50257AEE ^
454289D8FFB237A56D5214EAE88F0A9D328FEA1A ^
7E686B36595BEB4C0D4528FF960EDB55088A028D ^
F9789D1EF19A0084AC0E9F43A4BC0EE0478939EF ^
2F32B0E7CC8BE19C325545C816E77056D7BBE70F ^
6B1617746F073CFCD2CEBCAFBBE6FD0E28ED2D56 ^
CF8D2EA3888AD76761799383E5A15979F6DB7A88 ^
557AF6D9D5947203C60E98C9A79B92B8BD085E2B ^
C61A217423DE68ED6CD34C91756C8DD3A650A2A2 ^
73F3F79C151B6C1BD9369EDB26B932C2362B0593 ^
364141E5FBCDE83F210C5BBBEB6810F6299DE14B ^
F806BECD025D264FD59E93D9E3606A674C40F216 ^
E0C761A57F00CBFB07D49BCB034C36A7122F4C5B ^
5D3831044B9E0032FBE3C3425FFD13698F413B33 ^
7EB1AB41E9997753C5D530DF118E71E72D7B86FC ^
CC053EA1556269D7E8BCBA30B208FCBF0EE2EE64 ^
A57739B1DD41E7DC0C40D6B6159A7E73CE2748AA ^
90DA527C9DB9ACC2FD530D560A2F1191A80D0567 ^
6AC1F2A0B8CA0E5ABC9FDF1ADCE588FBDF5CC53E ^
43C1A0A0EE4163EC929726989F92B03639B233AB ^
8927F299462413AC29A74080E54D8EE2DB7165E7 ^
0C8D7E22226D91B423E781B508F31517EAAB607B ^
7286E20D7F08D18A893254FBD3CC833F7973DCAF ^
0CB8C235928B8E936C43B8F29EF3758B9FD54A7B ^
F67C24CC23E440CA3F206CEEB5504ECA54CD5CA3 ^
D78A25DEAA1E7ADADB3C145ED0E5263BA4F2910 ^
00AA68174D29492C578AC853FFCD55908292D41A ^
D5570EEDB09A62A5948F7F311F7ED5EF247F9AD9 ^
<D
```

File = byte-messages.sha1

```
# Configuration information for "SHA-1 Test"
# SHA tests are configured for BYTE oriented implementations
H>SHS Type 1 Strings<H
D>
0 1 ^
```

5 0 2 1 2 1 2 ^
5 0 1 3 4 4 4 ^
7 0 4 3 4 4 1 4 4 ^
10 0 4 1 5 3 4 4 3 1 3 4 ^
10 0 3 1 6 5 5 1 3 6 6 4 ^
13 1 3 2 5 3 3 3 4 6 6 1 4 6 2 ^
16 1 3 5 5 1 2 1 3 3 6 3 5 2 3 5 7 2 ^
15 1 8 1 5 3 2 7 4 5 6 7 3 3 1 6 3 ^
15 1 4 6 8 2 1 4 2 5 1 6 8 8 6 4 7 ^
18 1 1 2 7 3 8 6 7 5 4 3 4 3 5 3 3 2 6 8 ^
16 0 9 8 1 8 1 7 6 7 7 1 2 6 9 5 4 7 ^
18 0 7 1 7 3 9 4 7 7 5 2 8 1 7 8 2 7 2 9 ^
19 1 2 3 1 8 8 6 9 10 3 10 8 9 2 4 1 5 1 5 9 ^
19 1 8 5 4 8 1 3 9 5 7 7 2 7 2 7 8 7 4 8 10 ^
20 1 1 9 7 4 1 4 5 1 10 8 6 4 4 9 9 9 8 2 9 10 ^
19 1 11 6 7 7 2 6 2 6 10 6 9 10 5 11 1 6 8 11 4 ^
22 0 10 5 10 3 7 8 9 9 1 1 1 10 2 1 5 10 2 9 9 9 7 8 ^
21 0 1 10 1 6 9 4 2 5 2 11 8 12 12 9 8 1 3 10 7 11 12 ^
24 1 3 9 5 12 3 4 2 9 12 11 6 6 1 1 9 5 9 1 4 9 4 10 8 9 ^
25 1 3 2 3 11 1 12 5 6 2 7 8 4 8 8 9 9 8 4 9 1 4 8 10 9 9 ^
23 0 11 10 7 10 10 6 10 9 4 5 10 5 8 4 1 10 12 4 6 1 8 11 6 ^
22 0 12 8 10 4 3 8 5 5 7 11 13 11 12 11 4 12 3 6 5 11 10 5 ^
26 1 10 9 6 9 7 2 10 4 4 5 5 2 12 13 5 3 1 10 1 4 7 8 13 13 12 9 ^
31 0 2 6 5 4 7 3 10 6 13 6 3 9 6 2 10 5 3 8 4 1 11 3 5 3 7 11 1 12 9
12 5 ^
27 1 14 5 1 3 7 2 3 9 3 4 14 4 4 10 8 5 14 1 11 12 12 10 4 13 7 11 9 ^
30 1 4 9 5 5 8 9 5 10 4 2 4 7 9 9 6 3 5 1 8 3 2 13 3 14 9 8 9 10 14 10
^
27 0 12 9 5 8 7 2 14 12 3 8 14 6 6 4 7 5 7 10 7 11 10 1 9 6 7 12 14 ^
24 0 12 9 9 2 11 13 12 11 11 6 14 13 10 5 6 8 10 4 3 11 11 14 5 14 ^
24 0 15 4 5 3 8 12 15 8 14 15 9 12 12 3 10 13 6 11 10 4 13 14 8 8 ^
28 1 1 8 1 5 11 4 9 12 4 13 15 5 9 11 7 14 11 1 11 7 8 8 11 1 13 15 12
13 ^
32 1 5 8 3 8 10 7 8 1 5 13 12 14 5 3 6 4 12 15 6 6 10 11 13 9 1 11 6
10 3 7 14
2 ^
31 0 10 3 5 1 14 11 11 16 1 2 2 11 6 13 15 12 6 5 16 2 14 2 10 12 2 5
5 6 10 13
15 ^
34 0 3 10 8 16 9 5 12 15 4 11 13 3 6 5 10 8 1 3 9 3 11 1 2 16 12 10 6
1 9 1 16
5 6 14 ^
30 1 1 12 4 4 2 15 13 15 11 15 5 11 9 7 15 16 6 16 12 3 2 10 16 5 5 7
1 7 11 16
^
34 0 7 9 11 2 5 5 5 4 13 13 14 4 7 12 6 4 8 2 9 9 13 13 3 3 6 7 16 7 6
15 5 8
15 14 ^
36 1 4 6 16 15 11 14 14 4 7 10 3 4 10 3 6 7 14 4 6 6 5 2 7 8 16 2 12
16 10 14 3
2 3 7 14 3 ^

32 0 15 10 9 1 14 10 14 6 6 16 3 2 3 8 3 12 8 11 17 3 9 7 16 14 4 11
15 5 13 9
5 17 ^
30 0 17 17 13 8 2 6 8 16 1 12 5 17 2 9 8 10 13 14 11 17 12 5 14 9 11 9
11 4 11
12 ^
30 1 16 6 10 5 8 3 17 16 14 1 15 15 15 6 13 2 11 6 13 11 13 4 6 7 11
11 12 16
13 16 ^
33 1 16 16 14 16 2 4 16 11 6 15 7 4 17 6 5 7 6 3 14 16 5 17 11 13 1 1
14 13 3 6
14 5 16 ^
39 1 2 16 13 7 8 6 2 15 1 9 12 4 4 11 13 7 2 11 9 18 4 5 4 8 2 14 9 9
1 8 13 11
15 8 5 9 10 16 7 ^
34 0 2 7 1 1 17 13 6 11 10 8 5 12 15 6 15 10 12 4 18 1 2 8 11 12 16 10
12 18 11
16 12 11 17 6 ^
34 1 4 7 13 7 10 7 10 6 1 12 7 18 11 18 2 10 15 10 14 8 18 9 9 12 12 3
13 12 6
4 9 17 13 17 ^
40 0 5 7 3 2 1 17 14 4 16 6 13 1 13 6 6 10 1 3 18 3 11 7 9 5 7 11 17 1
9 16 5
15 10 17 3 8 15 17 8 12 ^
40 0 11 3 15 17 11 1 1 4 3 14 18 4 2 18 8 15 6 4 6 3 15 11 16 10 17 17
9 6 3 2
6 16 4 9 12 6 8 1 11 17 ^
37 1 2 19 12 8 16 14 2 9 16 2 6 6 7 9 10 9 11 9 14 11 15 5 16 9 2 17 2
8 15 8 4
3 14 14 16 16 12 ^
37 1 11 10 16 12 11 7 14 14 14 6 10 10 1 6 13 19 5 6 4 7 12 12 10 5 10
15 15 8
5 13 17 13 5 6 14 1 19 ^
38 1 2 6 5 17 9 11 18 18 8 6 13 15 3 3 15 5 13 18 3 2 5 5 14 7 13 4 17
7 2 17 3
18 15 7 15 16 18 11 ^
38 1 12 8 6 3 17 12 13 19 15 9 7 17 16 15 3 11 11 5 2 13 19 16 2 4 16
7 8 1 2 9
17 12 3 5 18 19 11 9 ^
39 1 14 16 14 8 9 16 5 1 6 3 17 18 16 9 1 15 9 10 9 19 1 3 3 20 11 13
17 1 19 8
3 4 3 7 1 14 19 19 19 ^
37 1 18 13 11 5 18 4 19 10 6 19 11 17 10 10 7 9 13 16 9 10 18 4 12 5
16 5 20 12
3 8 10 1 18 1 6 20 14 ^
36 0 8 9 6 12 11 7 7 3 17 13 6 20 17 9 20 16 10 12 17 8 11 8 11 10 5
10 14 18 8
19 9 12 12 2 20 19 ^
39 0 12 16 20 3 9 9 19 17 13 13 4 17 2 11 7 14 3 6 16 13 10 13 5 16 10
2 8 2 17
19 4 17 7 19 6 9 15 15 6 ^

43 0 7 2 18 5 7 18 5 2 15 7 11 10 9 3 2 14 19 3 11 8 18 15 5 3 5 12 15
16 10 17
7 19 16 2 1 16 6 3 19 12 5 18 16 ^
49 1 9 11 2 1 12 11 14 12 14 10 4 11 6 8 16 7 5 11 20 8 17 4 14 4 15 3
2 2 4 3
2 3 14 15 10 2 12 7 3 7 20 20 19 10 2 3 1 10 20 ^
36 0 19 20 12 5 19 21 5 21 11 14 19 1 17 8 9 4 19 3 17 1 14 21 14 7 6
5 20 14
21 20 4 6 21 7 11 12 ^
41 0 12 9 11 6 16 18 18 10 11 20 6 12 11 5 7 21 19 18 6 15 21 10 4 14
9 19 10 3
3 5 13 1 8 12 3 13 9 7 10 17 14 ^
45 0 10 6 8 3 17 18 3 21 19 6 17 15 4 9 15 9 15 14 4 7 14 8 10 13 4 11
10 7 6
21 1 14 5 11 7 7 2 13 13 3 9 13 8 14 20 ^
39 1 3 7 18 4 9 9 5 15 13 17 10 15 16 20 8 19 9 10 9 1 19 14 21 2 18
13 10 4 18
16 4 21 15 10 18 19 3 12 18 ^
41 0 14 4 13 11 1 11 1 10 2 12 4 21 10 21 18 9 2 16 7 20 6 7 12 19 20
1 13 12
10 8 21 15 7 19 13 6 8 19 20 18 19 ^
37 0 11 18 1 17 14 15 20 16 20 8 2 17 10 4 21 5 19 19 14 22 21 18 13
14 1 3 12
11 11 4 22 13 5 18 7 21 21 ^
48 0 9 22 19 12 8 16 5 17 5 9 1 2 9 6 12 6 1 7 4 3 15 1 14 1 12 3 10 2
10 14 21
13 17 6 6 17 1 21 2 14 16 17 9 11 20 21 11 18 ^
50 1 12 8 20 13 2 9 20 9 14 10 1 16 2 22 6 4 16 14 15 1 12 4 14 9 21 3
3 9 8 21
15 14 8 4 14 4 2 3 8 12 8 6 1 2 18 20 15 3 19 10 ^
44 0 10 20 14 6 3 4 21 1 12 4 18 2 6 7 6 9 20 14 10 10 19 17 21 12 15
17 7 10
11 8 10 12 1 19 19 9 18 21 4 18 11 9 22 5 ^
47 0 15 8 15 3 5 6 2 19 12 17 4 20 8 11 20 2 18 4 16 20 12 9 9 6 16 21
16 3 16
18 3 19 5 16 2 4 2 12 11 15 11 14 17 2 10 18 8 ^
48 1 5 13 3 21 5 3 6 18 18 10 1 21 21 7 1 13 12 19 1 14 6 8 21 19 21
11 19 13 2
13 4 1 10 22 16 4 9 4 10 16 3 7 15 11 9 13 17 12 ^
45 0 14 7 6 2 20 3 6 19 19 10 2 22 12 17 12 1 20 7 7 15 20 6 18 8 3 14
23 18 15
4 7 5 23 15 7 14 10 10 19 17 2 4 15 17 21 ^
45 1 15 11 8 9 17 5 12 18 14 6 20 17 21 12 16 9 22 9 20 15 2 22 11 2 6
11 9 8 2
4 14 19 3 21 21 23 8 2 11 4 8 4 20 22 11 ^
38 0 21 18 22 10 19 9 14 17 23 21 10 7 15 13 16 5 4 10 13 14 20 23 12
20 23 18
10 12 8 21 11 6 12 7 19 14 18 17 ^
40 0 18 22 6 9 22 5 23 13 6 8 23 20 22 5 22 15 19 20 9 9 1 13 13 10 14
13 5 22
14 21 9 21 19 14 14 4 18 13 12 14 ^

48 1 7 3 15 5 17 14 23 14 5 17 22 11 1 8 13 23 6 21 3 6 11 7 23 8 6 21
4 4 22
19 13 8 5 19 7 5 23 1 4 19 11 23 11 21 14 1 3 21 ^
43 0 22 14 11 7 18 16 17 24 12 12 3 13 19 16 22 4 16 4 6 23 8 18 11 2
3 20 22 9
21 8 23 1 23 20 7 16 13 23 4 13 3 7 22 ^
47 1 23 6 13 19 2 3 7 2 9 9 15 6 13 4 22 6 19 20 1 9 7 14 1 15 3 23 24
22 18 12
12 17 19 10 8 11 22 12 10 2 20 15 18 17 18 7 19 ^
47 1 12 21 6 12 4 7 18 17 3 2 14 24 14 1 23 1 11 15 10 6 18 20 7 1 8 1
16 6 20
23 23 21 10 10 12 24 10 11 23 2 12 23 9 3 24 24 10 ^
52 0 14 10 18 15 14 5 16 11 22 2 15 24 8 22 1 4 24 9 10 15 3 9 5 4 17
15 9 12
19 19 1 3 10 6 8 3 17 8 18 24 19 3 4 15 4 9 2 24 5 20 13 13 ^
42 0 20 17 19 22 13 8 10 19 15 11 1 14 17 20 22 10 7 11 16 9 21 22 17
23 12 15
4 24 7 21 18 2 21 16 1 19 18 20 11 3 15 17 ^
50 0 18 1 6 14 5 5 5 19 13 10 24 19 16 24 15 13 2 19 15 24 21 17 4 13
17 1 1 9
1 10 2 18 1 21 19 5 18 12 2 22 16 23 15 19 6 18 9 1 23 5 ^
51 0 21 13 14 11 18 12 13 3 19 9 20 22 20 2 11 12 6 1 12 16 18 2 9 8 4
3 11 17
11 5 4 19 16 11 23 13 18 1 20 8 2 16 16 21 4 19 5 5 20 24 16 ^
53 1 20 25 17 11 8 4 19 25 17 7 16 21 6 4 8 2 15 9 2 9 19 3 6 3 3 10
25 13 15 7
8 20 21 12 10 12 5 24 11 20 3 13 13 16 9 13 10 3 9 16 3 7 25 ^
49 1 9 9 14 2 13 17 25 2 18 5 19 23 9 25 9 10 23 12 12 7 13 8 15 7 1 6
21 2 8 7
6 16 14 14 12 15 13 24 10 15 11 10 8 14 15 21 25 21 25 ^
47 0 9 18 20 22 21 20 11 14 23 22 10 13 14 8 19 12 2 11 20 23 13 4 10
6 5 7 23
11 3 16 8 21 4 8 18 5 12 14 8 6 20 19 24 8 23 17 23 ^
48 1 7 19 1 18 1 14 22 13 14 5 8 22 18 14 25 17 11 12 22 2 12 12 16 12
13 18 17
12 17 14 18 8 25 9 23 5 3 8 14 24 17 7 3 3 23 17 22 19 ^
51 1 19 17 16 22 24 14 16 20 23 20 9 19 16 7 12 16 5 8 9 7 10 21 24 10
11 19 1
21 14 14 19 3 22 8 12 20 1 18 5 6 5 12 14 1 1 11 9 22 3 24 4 ^
52 1 6 1 11 16 1 12 8 11 11 17 10 22 7 3 10 2 6 4 24 16 24 19 4 5 18
11 12 9 20
21 25 2 21 18 10 20 25 21 3 17 17 5 8 22 25 19 8 10 19 7 11 18 ^
44 0 26 14 21 25 25 4 9 13 5 8 9 21 8 12 26 24 9 24 15 1 23 22 16 14 8
22 15 19
24 20 7 8 15 24 12 4 4 23 21 13 19 15 21 12 ^
59 1 15 7 3 21 20 8 22 14 23 26 19 2 10 18 3 5 3 1 9 15 15 3 7 13 23 9
7 1 13
17 14 25 9 16 2 2 6 13 7 19 25 17 1 5 21 2 7 22 5 6 25 3 12 19 6 2 4
24 17 ^
60 0 9 18 20 19 4 11 14 1 6 8 26 6 9 22 4 10 2 7 21 9 8 24 25 14 22 12
22 3 23

3 3 20 6 11 23 6 1 7 5 18 5 15 25 26 1 1 10 11 11 4 12 11 20 3 14 2 3
2 23 15 ^
49 0 12 17 24 11 8 6 24 16 15 22 21 14 6 12 20 19 5 5 12 11 6 23 2 16
23 7 24 6
21 2 17 17 5 25 11 25 20 25 24 18 6 12 19 25 7 6 5 2 25 ^
54 1 12 16 1 15 7 1 26 19 19 13 20 11 17 6 20 5 24 24 1 21 11 9 20 21
15 10 19
26 3 2 6 7 12 9 10 8 14 10 15 5 17 8 21 1 20 25 6 19 8 3 22 16 16 20 ^
63 0 17 13 11 10 17 15 12 6 13 14 17 4 12 10 24 5 13 24 3 5 2 5 11 14
8 5 10 17
16 8 4 14 21 15 3 6 17 25 8 2 3 3 19 10 13 22 22 8 2 13 25 17 2 1 19 1
14 20 2
5 4 15 24 ^
49 0 14 20 7 25 20 26 20 16 7 17 17 22 1 13 6 5 1 18 14 15 23 15 10 5
19 18 18
26 12 13 3 25 12 21 16 24 4 16 3 6 26 26 10 20 13 1 20 24 15 ^
56 0 3 8 14 5 5 7 11 13 11 26 11 4 26 17 20 19 11 10 3 10 14 9 6 9 7
16 10 4 4
19 19 2 26 13 19 17 15 24 15 4 21 22 13 13 12 22 2 14 20 5 18 7 17 24
20 20 ^
58 1 6 17 9 20 2 10 19 3 22 4 1 11 3 5 3 21 11 15 12 23 26 5 2 27 6 5
16 6 3 2
23 5 3 20 20 4 24 2 18 21 7 14 10 27 23 6 24 6 19 23 3 9 22 16 21 17
19 23 ^
58 1 17 7 21 19 6 16 15 15 20 14 2 25 19 14 18 19 7 9 1 14 11 10 16 3
23 14 26
10 11 1 18 1 12 24 19 19 1 7 2 3 24 7 12 9 2 8 16 20 24 5 26 26 4 9 2
7 25 17 ^
54 1 8 12 18 14 26 7 17 18 4 20 1 16 14 21 26 4 6 8 24 11 25 15 24 16
23 4 10
23 21 24 15 10 9 26 7 14 24 21 6 20 5 17 16 17 1 3 12 1 4 13 3 9 21 26
^
56 1 7 18 11 1 19 20 23 12 12 27 13 13 15 16 13 1 16 15 12 26 3 16 16
8 17 13
21 4 6 5 19 14 16 4 16 11 14 18 18 27 9 13 21 3 26 22 3 7 6 4 26 3 15
8 25 21 ^
50 1 20 13 9 11 20 6 11 21 27 25 20 7 4 18 26 16 27 5 12 19 7 23 6 25
25 2 11
13 25 21 18 17 6 12 14 13 24 11 14 19 26 27 25 6 1 15 4 7 27 15 ^
51 0 15 16 26 27 23 14 12 28 22 15 8 19 2 20 13 1 24 2 25 1 6 19 19 8
11 24 24
21 13 27 5 11 28 17 7 25 6 23 24 14 25 12 5 13 26 2 5 8 10 16 17 ^
58 1 5 26 18 19 21 3 12 11 13 4 14 22 22 14 16 13 3 22 16 23 5 19 6 13
10 26 17
27 26 4 3 25 6 14 2 3 5 7 23 11 22 8 25 2 9 25 18 17 8 2 14 4 19 1 5
27 13 24 ^
53 0 2 27 28 2 17 23 10 27 18 26 7 22 16 3 27 1 26 21 28 10 3 6 2 2 10
17 13 16
6 17 21 23 13 20 22 5 6 11 12 12 8 23 13 17 9 23 20 3 28 27 12 17 22 ^
59 0 28 19 5 21 4 27 8 1 19 14 20 6 7 9 1 6 22 3 19 26 14 8 6 7 19 15
23 1 17

16 6 26 14 5 22 25 4 7 10 16 21 10 18 19 24 16 23 8 3 17 28 18 10 2 5
3 21 21
15 ^
58 0 6 24 1 4 24 18 10 22 1 21 12 5 4 4 20 25 24 26 8 25 11 2 7 27 22
19 4 18
27 10 28 4 12 24 8 16 12 11 16 17 25 8 12 16 1 9 9 10 5 24 23 18 5 14
18 8 4 28
^
61 0 5 17 8 28 1 22 4 11 3 2 17 3 14 9 27 13 18 24 9 8 7 28 25 14 21
27 24 6 18
16 2 12 15 9 14 10 1 8 17 4 6 15 26 11 15 2 28 20 26 16 3 7 5 8 9 26
10 12 25
11 22 ^
53 0 9 13 24 15 20 2 4 8 2 22 20 19 4 15 14 28 13 25 10 10 12 28 24 22
26 28 15
9 11 26 19 22 27 2 21 8 20 23 26 12 10 21 9 15 13 25 7 26 1 13 5 9 20
^
58 0 3 9 21 22 7 1 23 28 1 2 8 22 12 18 28 5 18 14 7 11 17 20 20 7 21
13 8 28
21 22 2 16 20 15 28 9 3 22 13 10 23 4 16 11 14 1 10 8 14 14 15 18 13
12 21 18
25 28 ^
60 1 29 20 2 29 22 8 16 20 4 12 9 6 12 16 16 7 9 20 29 11 9 4 1 15 25
16 29 10
22 7 2 8 5 18 14 23 24 4 6 26 3 11 6 12 1 7 14 24 14 6 10 21 16 23 29
25 6 14
17 24 ^
64 0 12 10 5 10 15 25 8 15 3 7 13 25 16 14 1 29 22 26 15 27 9 1 8 8 28
6 13 5
13 3 15 5 23 8 23 2 5 5 4 17 13 14 7 17 12 27 3 18 5 7 5 26 18 15 22
28 16 13 7
2 23 19 25 15 ^
56 1 17 7 16 25 23 11 11 15 2 13 9 26 2 24 26 7 28 11 2 29 7 22 23 5
28 19 1 27
29 1 24 11 18 20 3 13 11 7 3 15 17 24 1 18 13 6 3 25 27 16 28 18 24 8
23 22 ^
51 1 29 28 6 28 14 12 28 27 22 4 14 25 1 3 9 7 11 14 15 16 10 19 12 19
11 20 13
28 4 27 28 7 27 12 4 28 21 17 22 20 17 15 15 23 22 13 12 21 22 21 29 ^
64 1 12 14 12 18 27 8 7 4 9 14 16 15 8 11 21 20 10 10 21 23 20 2 11 23
1 11 1 5
3 23 16 15 27 14 5 16 3 22 2 3 24 3 19 29 4 4 10 8 20 14 15 1 26 12 27
25 4 28
22 11 19 19 24 9 ^
60 1 20 8 9 5 25 19 17 19 15 7 24 24 21 3 20 16 8 3 17 28 18 29 9 23 9
10 29 4
12 24 15 5 8 22 17 29 12 3 8 29 15 21 21 4 7 20 7 10 7 26 10 16 24 6 7
12 8 12
15 17 ^
60 0 9 17 11 28 12 26 26 6 29 13 10 20 6 23 10 4 3 26 26 14 20 20 25
14 13 15

24 14 11 4 23 27 24 20 9 16 17 24 13 12 6 1 14 26 25 7 8 21 1 19 3 2 2
17 21 13
5 9 21 11 ^
54 0 25 1 27 24 6 23 16 5 1 20 29 22 25 9 25 10 3 28 28 25 19 18 16 24
14 15 5
28 12 28 26 29 2 15 15 9 5 18 19 22 12 15 4 6 15 24 16 9 4 26 25 18 27
12 ^
61 1 20 4 26 12 3 22 1 22 30 3 28 10 9 24 14 29 6 30 3 10 20 14 6 3 19
21 21 28
16 18 11 30 11 20 30 1 9 8 11 5 19 10 24 4 22 4 2 26 5 15 20 8 3 13 30
18 8 1
25 28 19 ^
56 1 20 15 21 18 18 12 16 13 24 9 21 2 28 6 1 23 9 18 27 27 4 9 13 10
8 14 16
15 12 11 14 21 14 10 11 25 17 17 30 21 13 27 26 26 22 14 13 17 21 19 9
9 20 23
13 28 ^
59 1 10 28 24 10 22 27 23 27 8 17 14 6 4 21 26 15 1 8 29 27 6 28 15 3
27 25 25
14 19 13 29 8 24 2 8 2 4 12 19 11 10 6 26 14 22 24 30 10 11 12 2 12 17
23 8 8
12 28 12 ^
56 0 14 28 2 17 4 8 3 26 9 23 21 30 30 20 4 13 28 29 9 3 17 7 19 30 28
1 2 20 9
12 24 15 30 20 27 3 23 11 6 29 25 23 26 17 20 10 22 15 23 6 25 5 4 30
2 29 ^
63 1 23 15 27 14 26 1 1 7 19 12 7 6 20 18 14 4 15 17 28 7 11 7 8 9 22
17 12 5
23 18 25 18 6 12 26 30 12 30 14 3 1 18 10 20 27 21 8 6 24 26 20 11 24
7 2 4 18
15 14 30 16 19 14 ^
52 0 27 15 4 19 25 29 29 7 14 18 9 11 9 27 11 15 29 9 28 20 2 30 26 21
17 8 28
17 22 29 24 8 11 18 29 15 6 7 27 27 17 24 18 23 11 19 8 30 5 24 22 24
^
66 1 25 15 28 23 5 10 21 5 8 7 3 10 19 17 6 9 15 29 10 7 4 1 16 21 16
29 13 18
5 3 8 15 8 21 29 20 5 27 2 13 27 7 7 30 2 18 26 10 2 5 29 21 15 25 26
24 8 12
20 3 9 10 30 7 12 29 ^
53 1 30 26 20 11 22 19 27 2 16 10 6 4 24 17 20 25 20 15 8 23 23 20 30
18 16 3
30 15 26 23 28 7 21 8 7 31 31 14 26 18 3 1 26 28 15 25 11 31 3 25 9 21
30 ^
67 0 2 6 14 4 9 5 28 8 17 22 1 4 8 7 10 14 19 10 14 8 27 9 24 26 4 30
11 8 19 5
21 7 2 27 20 16 20 20 22 14 13 16 26 14 10 3 25 22 25 23 21 10 15 15
29 8 13 4
2 13 22 20 7 4 20 31 23 ^
65 0 2 2 28 13 19 14 12 23 27 6 2 14 2 22 6 25 30 29 31 13 14 16 31 12
16 30 5

14 31 11 4 1 1 25 21 13 26 22 21 5 22 14 29 1 21 3 14 30 4 2 29 12 15
23 3 15 5
1 6 23 22 13 1 14 23 ^
59 1 25 5 15 6 13 3 22 11 23 31 24 6 5 20 4 14 3 29 8 29 19 7 29 23 25
28 19 11
15 27 21 14 1 19 20 26 12 7 12 1 18 13 29 28 23 29 14 23 7 1 9 29 24 5
30 18 5
25 30 ^
55 1 31 25 13 7 24 25 24 1 12 19 9 7 6 28 20 14 28 21 19 31 20 20 6 24
18 27 24
4 18 21 1 31 15 1 15 2 27 4 26 25 4 23 19 2 31 22 30 21 22 5 27 12 30
28 31 ^
62 0 27 15 18 14 25 15 17 7 28 11 28 29 30 1 17 12 10 2 18 20 21 2 11
12 5 4 12
25 14 5 5 24 22 18 31 15 22 29 11 3 21 31 21 27 3 28 7 10 25 2 15 30 9
30 7 22
15 9 3 20 24 14 ^
60 0 28 14 18 9 27 14 22 27 31 10 8 14 7 15 7 20 5 26 1 29 7 17 17 8 3
13 27 18
8 31 27 28 22 22 17 19 18 18 11 19 13 25 10 19 6 28 4 31 23 10 18 26
31 5 10 13
12 8 15 27 ^
60 1 24 22 4 29 22 31 28 20 4 16 21 3 1 15 5 15 6 30 3 29 29 7 27 20 2
20 31 22
26 9 29 16 4 26 32 17 20 14 28 17 19 6 24 11 26 28 5 18 15 8 16 20 21
4 9 12 4
8 17 29 ^
<D

H>SHS Type 2 Strings<H

D>

69 1 5 3 11 15 12 24 31 23 1 6 28 2 8 31 6 7 30 5 19 23 12 6 9 31 19
17 24 25
22 6 12 16 3 7 9 9 11 29 4 11 2 5 13 29 10 12 30 32 18 28 18 27 3 30 4
4 26 6
13 31 13 2 11 7 24 4 17 29 12 ^
95 0 21 19 21 23 11 42 36 2 13 4 1 33 22 16 27 9 4 33 16 3 30 15 11 32
13 17 38
32 9 38 4 36 15 32 27 19 42 18 6 36 22 10 29 12 25 40 15 29 23 28 30 4
8 11 24
9 10 31 28 43 23 16 29 33 5 40 26 3 19 12 36 43 5 35 37 5 14 11 45 35
16 10 8
32 4 15 35 26 2 39 22 37 22 30 29 ^
106 1 18 14 51 2 6 32 51 9 32 50 44 46 51 8 11 53 45 55 16 10 3 52 8
20 20 46
46 13 32 2 46 50 43 25 54 9 31 29 2 47 15 29 24 45 44 18 37 14 28 39
36 44 47
16 50 10 44 24 53 35 22 40 20 15 51 22 18 22 42 6 54 49 38 21 7 13 30
16 7 52
16 22 13 38 7 11 44 33 9 25 13 37 42 14 45 53 30 38 5 25 5 35 38 22 28
53 ^

127 0 58 35 43 28 5 28 63 8 12 25 9 47 53 29 62 7 37 2 3 48 5 12 55 56
28 35 12
63 6 58 27 27 48 44 35 14 17 22 56 10 8 1 16 15 42 63 14 51 57 19 41 7
8 56 47
34 52 22 48 60 43 9 1 52 4 21 49 61 18 50 23 13 46 62 23 45 62 9 56 18
23 31 8
30 27 36 13 38 4 58 53 47 24 18 41 58 19 12 18 52 42 29 44 45 26 63 34
32 41 64
15 26 55 19 2 49 6 30 53 13 54 12 53 37 12 37 43 ^
148 0 60 4 51 47 58 38 17 63 33 23 28 43 12 69 70 33 17 12 50 18 18 36
45 2 67
4 45 20 4 33 38 29 45 8 22 58 39 71 38 32 53 35 19 53 31 29 51 35 4 63
18 33 26
47 70 9 64 62 63 30 15 1 35 28 16 40 20 14 50 33 19 38 30 27 55 10 16
46 47 7
55 12 53 26 56 33 29 55 25 17 48 43 21 43 18 24 63 27 68 46 38 33 35
10 18 11
27 5 9 58 35 70 36 36 39 47 2 10 66 47 5 18 21 44 71 51 57 3 22 7 56
55 28 25
14 40 16 24 48 37 66 50 24 45 18 39 53 55 ^
165 1 15 62 35 29 15 40 19 76 67 4 5 71 46 61 26 8 77 48 1 23 12 60 40
24 44 33
29 42 73 66 49 61 20 30 1 54 52 42 39 64 23 65 37 24 20 11 26 66 22 77
22 57 7
38 57 33 61 73 7 64 1 49 35 76 14 27 21 45 68 38 58 73 13 72 47 73 33
8 66 23
38 4 56 77 47 10 71 13 20 31 41 6 51 3 18 17 61 47 14 48 76 46 28 34
43 1 56 4
25 7 65 41 1 34 37 23 59 59 27 26 13 15 14 75 60 14 1 28 59 26 65 61
16 23 17
28 6 19 2 35 49 30 29 48 2 63 73 59 1 3 76 41 11 19 18 43 54 63 67 51
4 9 78 60
66 ^
181 0 18 19 84 17 12 10 57 18 77 51 52 16 39 74 49 52 63 38 72 2 15 64
83 62 49
56 11 26 68 58 83 33 23 50 63 71 53 27 84 22 39 41 52 58 11 64 7 60 45
70 22 5
73 38 30 30 48 21 75 80 40 21 8 53 9 26 30 34 81 71 71 51 23 75 33 41
23 32 5 8
66 40 72 40 16 66 45 14 48 34 21 41 27 3 55 27 37 23 41 65 4 57 51 74
22 19 75
42 16 19 46 16 10 48 20 19 37 41 14 57 9 17 55 38 5 60 7 46 20 43 36
39 52 20
10 62 45 23 46 7 35 75 29 70 35 36 34 25 12 15 84 26 10 6 71 29 79 33
32 25 59
76 82 64 58 7 8 19 41 74 2 53 65 24 1 55 51 36 21 79 7 ^
184 1 60 66 66 6 3 9 73 12 7 40 70 18 71 70 65 51 14 14 27 50 9 87 81
50 22 19
40 37 16 79 12 34 37 76 82 10 61 7 81 49 67 26 45 82 50 81 63 45 69 31
31 76 51
9 59 34 51 54 34 83 10 33 51 86 81 82 69 18 8 22 64 19 86 62 58 33 37
17 34 5

29 83 42 76 50 54 66 39 9 1 36 43 17 65 6 35 56 72 71 83 88 10 1 8 87
22 6 21
78 25 89 43 62 40 55 85 31 89 74 63 46 28 24 26 31 17 7 8 27 19 12 85
17 20 27
77 10 2 54 80 17 52 74 76 69 78 11 20 80 4 29 24 85 75 18 39 23 70 83
29 57 67
72 70 33 4 15 46 42 2 69 13 53 33 69 64 33 64 14 40 69 59 78 54 ^
193 1 68 43 95 53 38 58 55 28 20 16 67 48 17 86 32 44 68 67 28 16 14
79 25 15
72 67 50 80 18 30 10 75 1 60 45 87 78 28 95 49 63 70 59 26 6 51 73 60
65 18 26
8 87 5 58 31 25 57 40 46 78 57 34 78 61 36 66 57 38 80 22 32 68 71 30
74 37 81
66 77 66 55 2 51 24 93 61 40 68 45 61 12 63 24 89 59 52 72 43 20 20 69
36 40 88
46 9 62 55 77 84 20 18 6 77 15 52 39 75 3 26 4 85 17 62 29 11 92 46 58
29 59 28
42 80 71 96 2 49 85 37 63 4 61 14 2 53 87 25 86 6 75 76 93 41 39 93 92
42 56 41
63 26 28 18 77 11 50 78 79 1 12 12 91 29 13 58 5 56 92 66 59 4 39 47
95 5 5 62
33 13 80 27 ^
203 1 35 28 11 7 20 7 17 3 3 30 89 13 65 56 66 63 22 82 16 31 55 56 77
91 91 71
101 13 10 85 101 95 17 99 98 91 33 14 20 48 32 7 64 29 38 35 25 4 95
23 34 1 85
81 23 31 96 71 84 50 15 79 47 25 51 45 35 66 19 61 60 9 31 93 64 70 30
42 86 53
1 71 46 42 22 38 96 10 99 34 76 26 55 73 63 63 97 23 92 81 64 46 1 30
31 35 86
91 88 64 87 16 37 69 84 94 60 100 3 47 52 8 71 87 57 29 76 43 18 45 46
15 65 12
44 42 66 60 15 68 19 58 39 62 76 9 92 101 57 32 4 34 15 41 62 32 89 71
43 35 31
41 21 17 82 33 96 27 62 29 82 57 46 62 15 24 99 37 83 40 52 46 56 80
98 3 91 74
6 27 7 58 94 10 41 79 97 84 77 74 26 99 35 ^
212 1 26 101 17 91 45 97 80 59 102 30 68 4 85 9 4 39 16 18 85 70 11 87
62 72 78
38 3 41 53 82 82 35 18 13 94 64 52 39 77 59 26 9 65 46 64 98 32 29 86
79 16 63
54 76 56 98 16 98 78 22 72 33 103 104 52 84 12 65 15 85 101 97 84 31
51 26 100
100 38 80 13 2 78 7 24 44 84 103 27 7 28 16 33 99 25 103 54 14 42 62
87 92 27
22 42 5 52 100 84 73 72 63 24 48 56 52 23 5 17 76 31 1 95 58 43 60 50
62 30 23
35 79 20 35 3 72 32 45 51 87 41 84 27 79 77 70 102 15 54 15 100 8 52
69 105 3
30 84 42 93 66 89 69 74 24 33 42 97 4 38 99 106 13 93 6 106 74 100 54
45 21 59

56 37 9 50 32 75 79 31 77 9 61 1 8 68 6 60 81 7 100 99 14 61 48 25 73
26 70 72
94 34 ^
233 0 11 98 110 88 35 110 35 64 49 88 93 28 85 6 78 65 90 52 24 97 51
39 51 59
23 1 3 49 33 11 78 27 35 55 64 5 102 4 70 25 56 58 38 66 11 31 96 66
104 59 41
86 58 29 79 41 40 72 51 12 92 34 52 44 69 104 21 97 89 96 48 21 4 61
40 28 67
34 23 85 44 22 62 52 33 84 23 30 73 74 4 79 12 81 47 80 53 47 89 40 19
80 62 34
61 29 41 95 43 1 70 63 55 53 18 19 13 48 10 19 89 49 4 52 53 56 76 10
8 104 77
15 28 38 75 109 3 85 90 8 40 8 93 90 43 39 14 60 17 36 78 56 105 80 35
75 36 58
82 50 100 98 45 74 13 66 95 72 71 95 34 14 98 72 33 38 37 52 6 14 107
59 3 29
61 67 98 92 5 93 17 98 36 87 41 75 71 57 88 17 25 91 84 3 58 20 92 69
51 50 36
31 14 25 18 30 18 1 41 104 30 82 59 87 70 34 96 28 47 62 81 103 48 ^
234 1 63 90 108 108 102 64 82 88 4 111 76 97 22 1 108 41 34 91 33 20
25 24 26 8
83 11 31 7 85 109 106 4 105 85 68 28 33 99 53 8 16 12 11 74 17 83 66
70 16 30 9
67 68 34 24 81 47 92 72 47 37 33 38 92 17 8 28 88 22 62 69 32 89 75 3
72 96 85
13 105 24 38 37 94 115 83 72 108 114 24 93 76 103 60 99 102 9 43 10 59
95 46 33
93 15 26 69 44 2 86 107 55 45 61 65 92 66 9 55 39 70 83 29 98 67 13
111 15 20
31 62 8 2 51 20 19 33 44 14 115 71 112 97 10 41 28 53 51 26 57 15 38
98 55 106
22 56 31 50 95 107 110 84 70 10 108 96 73 100 25 36 55 88 71 63 96 30
90 96 79
22 7 30 23 28 59 89 8 51 99 47 86 34 18 43 65 98 104 107 49 7 79 71 8
57 21 29
80 2 74 78 44 57 9 61 22 13 68 52 91 74 98 43 30 58 68 95 101 72 102
76 42 99
61 ^
249 0 27 117 45 119 80 2 59 52 8 76 20 94 102 69 96 42 46 106 67 9 110
89 71 69
34 31 15 85 16 29 100 82 37 62 68 95 108 44 23 114 34 36 56 93 11 30
96 12 31
67 14 114 14 66 70 30 81 46 53 119 85 6 104 47 92 72 70 5 70 15 115 68
105 33
97 13 85 106 14 61 29 22 86 45 57 69 91 38 38 28 66 13 60 95 103 3 15
5 113 38
23 62 5 65 94 107 73 104 37 47 102 117 3 78 35 7 95 56 78 45 52 28 46
43 37 32
53 19 55 29 47 97 76 115 83 71 11 45 62 73 99 116 2 24 116 7 28 41 2
29 37 52

23 5 118 79 31 57 89 61 24 101 78 50 93 73 41 7 33 45 47 24 1 48 73 36
3 25 87
46 28 108 54 68 53 67 119 28 36 118 104 42 88 27 112 4 74 85 1 63 39
97 71 74
75 76 10 49 12 79 11 50 103 118 94 117 118 37 27 12 94 60 28 51 47 82
110 17 15
105 23 52 43 12 21 22 81 41 12 74 90 42 108 117 98 67 4 69 85 ^
243 0 76 81 26 101 13 68 62 106 87 19 98 32 81 63 79 93 31 121 123 75
52 11 66
41 54 87 38 5 104 62 51 38 55 29 31 120 44 16 48 94 46 105 91 66 78 27
43 6 64
2 55 79 75 84 113 22 4 113 109 31 33 17 96 11 29 63 98 103 107 116 34
14 9 95
38 18 51 75 33 109 118 55 66 4 76 7 75 70 82 74 23 1 26 69 40 112 99
47 65 31
70 119 52 103 88 85 86 28 16 12 76 25 22 78 64 21 86 27 61 77 72 108 2
18 106
119 121 54 16 85 72 2 73 26 88 66 60 80 35 24 117 63 24 44 67 52 122
119 33 72
16 99 98 69 54 19 42 28 53 114 32 117 81 100 57 49 123 56 21 68 80 53
95 1 45
95 107 98 87 1 27 24 99 116 16 67 1 113 91 84 25 40 25 72 3 28 90 87
112 80 16
117 45 77 36 90 105 59 88 122 64 108 108 71 98 18 50 115 93 105 77 35
6 46 55
47 102 4 26 87 111 120 81 113 4 57 105 3 84 94 115 61 73 ^
255 1 91 47 51 9 57 9 55 94 61 61 68 46 107 6 35 81 114 78 96 74 14 89
73 67 67
69 113 107 11 98 113 109 20 92 17 67 70 88 57 10 124 9 60 122 93 91 45
7 15 24
51 5 98 115 24 49 90 104 117 66 128 94 64 80 12 43 91 46 111 59 58 77
30 14 88
60 123 68 41 44 68 40 104 118 41 43 93 90 105 92 16 127 26 54 125 114
79 71 24
48 21 25 118 40 103 49 91 44 67 65 25 119 109 18 48 23 69 112 38 61 64
87 84
104 119 110 122 92 22 1 8 83 34 100 32 62 41 46 112 34 102 76 56 39 4
127 30 13
19 110 124 7 16 128 95 4 124 11 104 116 126 49 95 3 55 96 70 90 101 4
122 96 75
118 39 128 99 92 18 42 20 87 83 35 75 111 61 67 71 28 101 9 56 34 105
95 71 23
73 71 26 57 15 23 76 55 99 89 128 98 117 68 43 88 62 38 62 39 2 83 36
15 26 60
128 96 73 74 10 1 12 42 22 2 77 33 33 32 57 13 14 82 57 12 39 3 58 80
14 87 85
44 69 109 119 ^
283 0 102 55 53 41 60 88 25 67 58 76 44 22 68 118 108 40 95 96 81 90
85 28 77
18 11 37 72 93 60 110 124 119 95 131 91 37 109 126 8 73 69 72 80 17 83
5 76 20

32 15 10 1 103 18 22 116 98 9 51 104 102 44 33 15 12 24 31 89 1 6 28
101 8 64
72 106 30 5 52 89 111 39 108 64 85 17 57 124 22 105 78 115 3 40 108 66
108 77
128 103 44 35 38 13 95 10 111 63 98 117 61 51 126 69 96 70 70 59 39 13
97 33
112 2 77 7 123 70 83 29 66 67 49 79 19 104 115 14 60 2 55 40 71 33 28
114 51 91
17 46 45 128 57 87 62 25 115 38 50 55 90 74 8 51 102 79 43 94 36 122
94 12 41
36 25 104 91 24 7 99 80 30 126 32 63 122 107 114 27 28 79 41 12 35 51
115 122
70 22 79 65 2 88 27 17 59 15 23 44 57 5 65 6 26 78 80 125 93 84 100 45
22 129
68 36 111 74 118 11 50 42 120 47 21 8 86 112 26 67 60 99 45 93 47 8 38
59 52 56
124 20 82 18 117 24 18 46 106 19 117 26 41 47 45 130 7 15 1 4 5 100 10
85 50 44
11 48 92 119 108 42 118 125 ^
272 0 8 61 99 70 96 20 87 123 134 82 22 2 110 118 33 86 5 7 5 94 56 15
60 96 54
13 22 55 99 4 25 105 17 37 69 10 38 117 117 30 70 13 9 109 115 62 94
52 66 117
100 135 7 75 23 5 81 110 31 118 29 1 62 11 41 88 109 119 102 37 3 30
123 47 31
56 134 29 124 116 118 99 21 56 77 91 23 37 135 81 44 51 67 95 51 133
30 57 67
116 122 48 100 7 132 97 106 69 93 4 95 125 102 103 119 81 57 133 96 37
118 50
117 113 81 127 17 45 103 32 121 129 60 43 65 127 30 36 132 110 52 53
35 71 12
76 22 72 130 112 99 76 26 21 73 63 63 97 23 58 115 132 114 1 132 31 35
18 23 54
30 53 118 37 35 84 94 60 100 3 47 18 110 105 87 57 63 76 43 52 45 46
49 65 12
10 42 66 60 117 34 19 92 5 28 76 9 126 101 125 32 38 34 15 7 62 32 21
3 43 69
31 109 123 51 116 135 130 129 130 63 14 57 80 62 15 126 31 105 83 108
120 80
124 46 98 105 91 6 6 27 7 58 128 78 7 79 63 84 77 74 128 65 61 95 121
17 24 123
117 51 122 ^
284 0 44 71 43 20 126 58 53 47 98 18 19 119 93 29 70 39 94 112 44 115
135 98 82
10 67 29 102 113 68 80 19 75 1 91 114 87 80 7 40 37 86 120 16 104 136
117 82
138 32 65 114 119 137 121 8 12 46 126 26 119 73 130 60 76 113 100 14
133 26 116
34 120 80 95 84 53 15 24 44 51 4 10 23 77 24 99 66 37 54 63 42 136 21
34 76 5
17 128 101 1 59 40 113 112 32 97 31 93 105 79 91 18 39 1 103 132 51 68
124 111

13 97 43 128 69 84 85 72 15 12 26 87 16 16 92 101 13 77 4 118 89 103
56 42 16
60 44 39 126 46 18 83 93 41 105 3 82 106 115 91 6 4 54 115 15 120 109
113 48 41
9 95 20 62 67 105 111 25 132 7 116 46 138 44 83 61 124 131 35 107 6
109 81 114
67 41 137 77 56 74 73 34 12 14 69 52 11 98 47 54 83 81 6 1 15 88 35
139 80 83
49 89 27 47 130 92 133 87 51 112 76 49 109 49 57 93 73 22 117 50 64 58
97 139
36 131 111 133 58 33 8 88 55 38 90 46 30 118 57 29 82 74 41 117 38 46
94 92 5
105 15 117 70 103 68 60 120 48 21 110 85 40 81 66 ^
291 0 46 113 52 134 79 74 64 57 18 23 9 52 8 16 103 57 138 59 59 65 92
2 7 130
92 8 34 40 86 131 140 100 112 4 42 1 110 108 43 37 15 67 19 35 94 61
130 98 35
88 34 65 104 56 126 118 50 87 10 81 109 90 86 118 32 6 114 88 39 38 39
62 3 12
134 72 137 35 75 81 115 106 140 112 11 123 41 103 45 95 84 71 107 13
26 110 96
62 16 109 84 59 53 38 27 8 28 13 32 137 17 138 41 122 36 99 65 99 83
36 112 29
49 70 96 126 136 131 116 3 18 17 126 142 14 37 141 141 123 42 13 20 83
42 139
83 54 49 58 42 7 137 29 48 16 121 127 34 52 140 106 128 58 36 124 83
24 69 54
61 112 17 6 95 97 24 57 86 124 59 71 119 67 1 109 54 68 49 57 132 32 5
71 113
40 80 104 75 106 133 31 126 130 104 62 9 39 44 66 116 141 135 96 132
19 41 121
126 124 77 8 4 60 82 6 101 124 89 51 123 48 40 85 77 21 112 10 69 66
115 87 16
108 30 84 65 80 103 32 131 134 73 47 10 63 39 50 93 37 135 114 69 48
34 58 23
27 133 37 9 40 98 41 115 99 70 83 29 42 67 133 55 79 80 91 122 12 2
115 112 47 ^
293 1 33 13 99 138 1 42 89 118 87 113 99 12 134 142 100 38 5 55 75 14
110 108
42 64 130 79 138 62 64 69 57 11 123 25 59 16 111 94 24 65 30 51 119 48
107 92
84 69 28 136 143 54 20 6 70 47 142 64 4 65 59 73 99 134 146 102 125
116 57 137
137 72 48 128 78 5 80 63 54 85 30 22 129 68 21 21 74 28 128 107 27 60
2 93 95
71 37 11 37 15 39 102 3 104 65 80 59 52 113 34 20 67 60 27 81 135 46
106 106
102 68 128 17 15 100 124 15 43 136 122 100 67 142 35 14 53 120 2 89 93
99 73 9
122 39 77 15 96 90 43 79 134 60 92 105 55 96 31 119 77 97 72 23 140 38
30 43 83

136 88 107 117 72 109 118 58 91 119 73 95 100 59 138 123 54 49 143 50
133 66
106 45 80 88 42 93 5 59 77 101 74 110 104 40 92 19 77 76 86 102 129 3
144 101
139 134 56 90 18 91 94 85 55 10 137 11 58 1 107 113 70 22 7 56 29 143
111 8 46
45 116 122 129 89 7 121 53 95 14 49 118 62 125 91 37 97 15 35 100 63
140 63 50
51 58 26 127 6 45 59 102 121 114 85 141 135 10 72 19 106 66 66 41 53
13 38 1 21
103 50 108 46 119 ^
297 1 46 31 132 112 28 63 124 97 129 43 40 72 99 107 132 137 96 139 99
145 121
144 118 37 81 39 94 60 55 109 47 109 110 75 42 12 139 137 43 128 106
107 19 126
12 101 148 127 15 117 125 125 62 96 13 76 70 96 101 110 138 8 95 76
143 17 32
97 79 149 39 31 94 123 21 41 135 55 84 70 33 135 118 50 62 121 81 1 45
144 93
60 5 64 137 8 105 91 82 67 27 113 119 53 18 98 79 48 84 32 135 128 5 1
20 76 17
85 108 72 36 141 140 49 150 105 104 3 149 14 54 18 148 64 49 125 37 28
28 101
22 104 91 32 82 117 12 114 69 58 2 58 115 9 108 47 59 65 14 92 7 4 86
98 16 82
92 95 38 94 10 10 48 97 104 66 115 97 142 115 122 119 40 97 16 32 47
34 88 89
26 50 12 76 80 51 40 9 133 24 44 40 122 84 108 22 142 140 99 44 15 54
8 42 125
150 130 21 79 124 62 46 119 15 29 91 57 150 42 138 71 61 68 80 114 6 1
70 121
18 35 113 56 87 86 10 73 14 29 41 72 89 1 133 87 101 123 59 90 142 77
133 52 78
48 34 138 134 27 17 60 131 147 61 93 148 39 132 49 62 71 36 91 4 139
49 100 120
43 113 144 30 94 73 127 40 125 ^
313 1 35 97 95 76 105 88 32 138 30 69 61 40 47 21 107 6 39 81 114 53
125 53 147
14 4 73 146 96 98 13 136 11 98 117 138 153 67 146 71 99 88 7 139 24 13
35 47 97
145 74 36 119 3 51 84 48 119 53 49 15 79 17 120 103 148 64 30 41 97
120 75 111
63 58 131 134 18 13 10 48 18 16 48 43 15 54 18 41 47 122 144 80 92 145
77 1 33
89 54 46 78 48 21 54 43 40 53 24 16 73 42 94 29 44 34 151 152 23 123
12 142 140
43 37 88 29 19 35 72 96 151 130 62 112 34 36 91 120 50 112 138 2 105
60 68 137
131 5 17 19 139 74 11 120 78 149 58 128 15 104 16 126 78 20 57 134 71
49 90 76
108 126 100 54 68 39 132 153 42 147 146 124 62 87 35 75 61 65 46 100
82 105 113

31 63 5 95 54 71 77 127 150 80 36 144 2 130 59 74 39 3 152 121 122 18
117 12
117 141 118 135 62 36 69 5 39 53 150 52 153 143 30 66 96 126 131 56
137 8 7 86
142 14 7 111 141 93 136 137 134 43 12 89 23 44 9 152 146 121 97 19 38
110 91 67
14 32 110 66 68 8 130 84 73 118 59 24 41 72 121 150 55 37 138 27 104
66 124 9
51 109 47 125 109 148 8 29 47 72 146 149 61 93 10 20 54 15 76 133 125
106 110
67 ^
330 0 23 9 26 136 27 51 115 122 44 106 6 146 108 113 85 51 8 96 47 56
137 62 59
89 143 71 140 14 85 156 139 99 154 30 53 115 35 147 108 148 58 52 28
103 19 92
95 152 152 10 11 13 155 67 11 83 101 69 153 152 45 141 14 120 129 140
119 59 2
89 73 70 83 29 16 67 81 29 1 54 65 96 117 2 37 47 128 33 3 89 108 98
139 49 78
27 103 39 119 94 132 90 38 132 55 65 131 90 58 2 54 100 69 118 22 44
19 7 148
93 25 29 123 81 64 131 55 30 1 89 38 97 82 64 9 28 86 123 151 10 133
40 154 102
4 111 65 9 63 59 124 116 72 105 76 57 137 97 32 145 108 78 112 50 43
34 75 20
22 129 68 11 118 74 125 118 57 17 20 129 53 65 61 144 1 17 142 156 52
100 54 15
20 59 52 63 131 20 57 124 31 125 46 106 76 92 8 98 154 152 80 114 15
140 136
112 100 17 92 25 151 150 80 99 69 83 49 43 156 102 19 57 122 96 30 3
39 134 40
32 75 5 76 127 138 99 17 57 52 150 130 18 127 33 23 116 107 78 77 77
42 69 68
48 41 69 33 75 40 49 128 103 4 146 93 10 83 66 96 152 30 38 12 33 5 39
47 41 34
60 74 20 42 156 67 46 56 102 89 3 124 81 99 104 56 50 8 61 74 55 15 87
108 28
138 47 93 60 2 124 46 126 103 91 145 36 25 116 122 51 ^
322 0 75 7 107 158 81 105 154 90 20 125 77 114 69 92 7 58 21 98 154 50
128 149
117 127 153 45 3 18 121 86 29 71 79 101 2 5 22 143 10 27 53 146 157
148 112 33
22 80 123 24 147 1 112 82 159 63 74 97 109 33 151 32 89 87 132 117 46
129 59
115 91 114 118 37 21 9 94 60 25 89 47 79 110 55 12 143 99 87 43 88 56
57 160 76
12 71 128 77 146 117 95 105 42 66 3 76 20 76 101 100 118 149 45 26 143
148 32
57 39 129 19 31 84 123 1 152 135 5 54 30 13 125 68 30 62 101 51 142 5
94 83 20
116 24 107 109 105 91 42 17 27 93 69 3 139 68 79 38 84 2 85 128 126
122 131 46

17 35 98 42 26 111 100 29 120 55 84 114 109 145 14 18 138 14 9 85 7 18
129 91 2
94 51 133 82 87 123 64 39 8 103 38 75 110 78 7 9 45 115 42 138 135 86
78 16 62
52 75 159 54 151 121 149 77 74 16 85 47 102 105 82 119 10 67 137 153
148 135 28
49 26 151 153 36 80 11 130 113 24 44 30 102 24 58 133 122 140 99 24
156 54 119
42 115 140 90 132 19 94 2 157 99 136 19 71 7 130 153 108 51 21 58 70
74 137 1
40 111 149 5 103 6 27 76 141 23 125 140 1 72 29 152 103 87 51 93 29 80
132 77
123 153 68 159 14 98 114 158 121 158 81 131 ^
322 0 35 93 109 125 119 10 10 19 135 26 4 74 135 35 120 129 113 92 17
29 47 88
14 159 149 87 45 36 75 68 22 138 20 59 61 144 151 11 107 6 153 81 114
43 85 157
97 148 118 73 126 56 58 137 96 11 98 67 98 103 57 146 21 59 88 151 139
148 127
25 17 47 115 34 160 109 107 51 64 28 69 13 49 149 69 141 90 93 118 64
10 1 67
80 35 111 13 58 101 124 132 147 154 18 162 6 162 33 5 34 142 41 161 82
114 70
92 145 57 155 137 114 79 44 36 48 48 21 14 13 40 33 14 150 33 32 54
143 14 4
101 142 23 93 136 132 120 147 17 38 163 143 5 52 46 151 130 32 72 34
124 150 51
100 112 128 126 65 10 28 87 81 159 131 19 99 54 125 110 58 119 28 78
129 104
140 126 38 154 27 114 61 153 90 66 98 76 50 158 48 39 82 123 22 147
136 114 52
37 35 75 41 15 150 60 52 55 103 21 23 129 95 24 71 47 97 130 50 140
144 106 100
9 64 19 117 122 71 92 8 77 156 97 121 98 85 2 36 39 109 143 23 120 156
133 93
154 36 66 116 131 160 127 162 161 46 142 14 141 81 141 63 86 117 104 3
146 39
127 34 133 102 106 91 57 9 28 60 61 7 158 12 80 26 8 122 80 44 63 68
49 158 21
32 81 150 15 141 108 161 64 46 124 123 31 99 27 105 109 98 112 144 ^
336 1 34 161 107 149 48 67 138 109 156 104 37 133 60 80 84 81 160 9 16
96 164 1
95 112 4 86 163 116 98 103 55 31 8 56 37 36 127 32 9 89 103 31 100 161
85 106
119 89 154 43 115 162 137 108 128 38 42 155 103 9 62 65 102 122 10 138
160 125
47 158 43 91 69 123 132 35 121 4 110 89 130 69 29 139 69 53 70 83 29
163 67 41
9 108 34 45 76 87 2 144 164 98 33 160 79 78 48 89 9 38 134 93 146 79
54 122 80
38 112 55 55 101 70 8 129 44 70 59 98 149 24 136 124 138 63 25 166 83
51 34 91

45 30 118 59 28 87 72 44 116 28 36 103 101 113 10 114 62 111 71 65 126
53 19
114 86 42 85 36 57 137 57 159 95 88 78 72 20 23 14 65 10 22 129 68 1
68 74 75
108 7 7 147 109 13 35 51 104 158 164 122 126 2 50 4 132 127 59 52 13
81 20 47
107 74 148 115 46 106 46 82 115 68 144 142 60 104 15 90 136 102 100
134 42 15
141 100 40 49 49 73 166 13 156 82 166 37 82 96 137 130 166 134 20 139
45 122 56
107 98 79 124 17 32 130 120 165 77 23 130 96 67 68 47 37 12 29 18 38
158 19 160
55 147 39 118 83 121 96 43 137 33 66 86 112 147 155 149 140 5 19 17
148 161 10
44 159 146 57 16 26 102 49 3 104 61 59 74 56 10 165 31 54 25 142 157
37 58 165
128 154 73 50 149 94 137 ^
330 1 61 51 65 132 23 169 116 122 14 66 7 98 131 72 69 127 72 163 125
68 69 51
47 159 31 164 71 118 50 83 113 81 127 153 45 137 134 121 68 163 26 43
65 127
166 138 98 144 18 53 137 139 148 76 158 4 62 78 167 102 144 94 55 141
63 29 97
91 24 115 166 80 69 132 99 1 120 23 88 64 87 118 37 137 152 94 60 168
71 47 52
110 37 155 125 63 42 43 52 11 12 151 31 12 44 110 32 128 117 68 87 24
39 164 76
145 58 101 91 100 140 151 143 130 32 21 3 111 1 31 75 123 153 116 135
130 27
164 165 116 23 12 62 83 24 133 139 49 74 154 80 158 80 64 105 91 6 142
27 75 24
128 112 41 79 29 84 145 40 128 99 95 95 19 17 160 89 15 17 84 64 11 93
10 66 78
73 127 148 18 129 139 143 49 150 9 84 82 154 85 15 88 82 60 87 19 12
133 58 20
39 65 51 141 134 27 70 167 120 117 86 60 16 44 16 57 132 18 142 85 104
59 47
141 58 2 66 96 46 119 153 40 110 126 103 90 144 13 26 106 144 80 145
134 103 95
24 44 21 84 140 13 97 104 140 99 6 147 54 83 42 106 131 54 96 135 67
118 121 81
109 10 53 132 112 117 81 33 155 49 61 38 119 1 13 102 131 148 94 131
143 67 123
148 89 104 135 72 145 152 76 87 6 66 2 71 123 77 114 108 59 123 166 62
96 140
94 149 116 169 ^
349 0 125 17 93 82 80 110 156 147 156 99 154 4 29 90 163 120 84 113 56
8 157 29
61 169 141 113 78 48 50 13 138 11 50 61 99 106 2 107 6 117 81 114 34
49 112 52
130 82 73 108 20 22 110 60 11 98 22 62 58 48 146 149 23 88 142 139 121
91 16

163 2 88 171 133 100 62 51 46 10 24 150 49 131 60 114 63 84 91 64 165
138 40 44
172 111 141 58 74 115 96 129 145 164 153 170 126 24 169 16 115 41 125
46 87 61
92 145 39 155 92 78 70 35 27 21 48 21 151 159 40 15 5 132 170 23 18
107 160 150
56 133 23 66 109 123 102 102 172 166 145 116 151 34 1 151 130 5 36 34
97 114 15
82 128 112 119 99 29 138 165 42 36 159 95 19 63 36 89 101 40 92 1 33
93 104 113
126 2 136 96 52 108 90 57 89 31 5 113 30 39 37 96 4 147 127 105 43 165
35 75 23
143 105 24 25 10 94 12 160 102 95 170 71 20 70 112 23 95 144 61 73 137
55 1 81
95 26 65 172 41 147 79 103 80 40 121 36 12 64 98 169 93 111 115 48 127
9 39 107
131 115 118 162 161 10 142 14 123 54 141 36 41 99 77 140 128 167 82 25
106 57
70 64 21 19 15 34 126 149 167 53 163 127 86 35 8 54 23 40 140 3 169 45
150 152
96 81 143 28 28 124 87 13 90 9 87 109 53 67 164 28 131 89 149 42 55
126 79 132
74 19 133 30 68 72 75 148 9 10 72 152 144 83 106 153 74 163 98 152 ^
375 1 94 28 13 8 20 28 18 118 5 140 89 67 171 64 152 85 61 101 80 154
149 34
115 135 128 108 110 20 33 128 103 35 38 57 95 10 111 151 98 29 149 7
82 69 96
114 26 103 171 101 53 121 24 2 121 51 35 70 83 29 154 67 5 167 63 16
27 58 60 2
99 128 71 33 160 70 51 3 44 149 2 89 84 101 43 18 113 71 38 94 55 46
74 52 139
102 35 43 50 80 122 6 100 88 129 36 25 148 47 24 7 55 36 30 82 32 19
78 63 26
71 28 167 85 56 167 95 159 78 26 66 35 65 90 44 159 105 59 15 67 57
137 21 132
50 70 78 36 169 5 172 56 1 22 129 68 168 23 74 30 99 138 174 120 91
153 8 42 68
158 155 104 99 133 5 135 96 82 59 52 144 36 20 38 62 29 112 106 46 106
19 73 70
41 135 133 42 95 15 45 136 93 100 98 173 6 132 55 4 4 31 64 130 162
156 64 157
19 46 96 92 103 139 134 2 94 18 86 38 89 62 61 79 157 14 112 111 156
32 14 85
78 31 59 20 1 161 169 149 29 122 150 133 37 102 30 109 65 85 51 174
110 164 66
77 76 111 119 131 95 5 1 166 103 134 141 17 158 123 137 48 165 175 102
13 3 86
43 23 47 56 150 165 4 36 174 115 157 168 13 147 119 109 55 41 140 67
27 31 27
53 126 17 163 116 122 160 60 7 92 113 66 45 109 60 151 125 62 39 39 17
153 13

152 53 94 50 53 89 57 127 153 45 119 104 121 56 145 172 19 41 103 166
108 68
126 12 53 131 127 148 52 134 168 50 48 155 72 132 82 37 129 63 175 160
^
366 1 73 15 79 139 71 51 132 81 135 111 166 61 37 60 118 37 92 134 94
60 150 53
47 25 110 19 137 107 27 176 43 16 145 146 142 165 12 17 92 166 110 117
41 69 6
12 164 76 109 40 101 82 82 131 134 115 143 112 32 164 146 93 162 31 66
123 144
80 135 94 137 156 107 157 173 62 65 176 124 112 4 65 127 44 131 53 19
105 91
149 106 27 57 158 92 85 14 79 20 84 127 174 128 72 68 59 171 17 124 80
167 8 57
28 172 66 144 48 42 37 109 121 18 120 103 116 13 132 39 73 145 76 158
43 82 33
51 153 164 97 13 2 3 20 24 114 98 9 25 131 102 99 86 42 16 26 159 39
105 161
133 49 59 41 20 105 31 136 30 87 10 119 135 13 83 99 58 45 99 156 26
61 135 143
80 118 107 76 77 24 44 12 66 95 147 61 86 140 99 167 138 54 47 42 97
122 18 60
90 40 73 85 63 82 1 35 96 94 81 54 15 128 40 52 2 101 1 165 93 113 130
85 95 98
58 105 112 53 68 108 72 100 152 49 87 140 39 154 62 114 77 105 63 50
87 157 26
78 122 67 140 71 170 119 5 93 64 50 104 144 129 138 75 130 4 178 60
139 120 54
113 32 2 133 17 43 163 129 89 72 149 155 30 38 7 138 5 44 61 69 76 175
107 6 93
81 114 28 25 82 22 118 58 73 96 175 177 92 36 11 98 171 38 28 42 146
125 178 88
136 139 103 67 10 151 151 70 153 115 94 32 51 34 177 173 132 49 119 54
96 45 78
73 64 159 120 22 20 154 111 117 58 56 109 72 143 ^
372 1 136 146 144 170 90 15 169 180 88 41 89 10 60 52 92 145 21 155 47
42 61 26
18 176 48 21 124 141 40 179 178 114 143 14 164 71 142 132 11 124 23 39
82 114
84 57 163 130 127 89 133 16 138 151 130 160 34 70 78 161 64 92 112 110
72 175
102 138 179 173 159 59 19 27 18 53 92 22 65 156 170 57 104 86 126 148
118 155
78 43 63 90 48 80 168 142 68 12 39 174 69 168 147 118 96 34 129 35 75
5 107 60
170 180 147 85 3 133 75 95 152 71 175 43 94 178 50 144 16 46 101 46
165 45 68
163 38 172 5 138 61 85 62 177 76 36 167 19 53 151 66 66 97 3 100 164
12 98 131
70 109 162 161 156 142 14 105 27 141 9 178 81 50 113 110 131 37 16 79
12 34 37

167 173 10 152 7 81 140 158 26 136 82 50 172 154 45 160 31 122 167 142
9 150
125 51 54 125 174 10 124 51 177 81 173 69 109 8 22 155 19 86 62 149 33
37 108
34 96 29 174 133 167 50 54 66 130 9 1 36 134 108 65 97 126 56 163 71
83 88 10 1
8 178 22 6 112 169 116 89 43 153 40 146 85 31 89 74 154 137 28 115 117
122 108
98 8 27 110 103 176 17 20 27 77 10 93 145 80 17 143 165 76 69 78 102
20 91 171
95 29 115 176 166 109 39 23 70 83 29 148 67 163 161 33 4 15 46 42 2 69
104 53
33 160 64 33 155 14 131 160 59 78 71 19 176 107 65 38 82 55 40 56 40
115 84 29
25 44 68 104 176 76 64 123 18 25 136 23 6 171 31 30 30 58 14 13 72 57
14 41 177
^
363 0 135 69 16 167 79 143 46 178 26 3 65 58 36 135 97 35 175 51 152
57 137 173
108 10 54 78 4 153 173 164 48 177 22 129 68 168 167 74 174 91 106 174
96 75 129
168 34 36 158 147 88 75 101 149 103 64 42 59 52 112 180 20 30 22 173
80 98 46
106 179 65 30 17 127 125 26 87 15 5 136 85 100 66 141 182 124 15 156
148 15 56
98 146 156 48 149 3 14 96 52 79 115 134 170 54 178 54 22 73 30 45 39
133 182 96
103 148 176 6 45 62 183 51 180 153 145 145 117 21 90 118 109 21 62 22
101 49 53
11 142 86 132 66 69 44 79 87 115 55 5 169 150 63 110 109 177 150 91
129 40 149
159 102 165 3 70 27 175 23 56 126 165 164 20 158 91 157 136 157 131
111 69 39
33 132 43 19 175 179 37 118 9 155 116 122 128 52 7 84 89 58 13 85 44
135 125 54
183 23 161 145 173 136 29 62 50 13 57 25 127 153 45 95 64 121 40 121
140 171 9
71 166 68 28 102 4 53 123 111 148 20 102 160 34 8 139 32 116 66 13 113
63 143
97 63 10 59 124 66 41 132 71 115 106 151 46 22 45 118 37 67 124 94 60
140 43 47
10 110 9 127 97 7 156 43 180 125 126 137 145 12 2 82 146 100 117 26 59
180 181
164 76 89 30 101 77 72 126 114 95 143 102 32 149 131 83 157 31 61 123
139 60
135 74 169 122 151 102 137 168 62 55 166 119 97 163 60 112 24 116 38
178 105 91
134 86 27 47 138 72 70 183 79 15 84 117 154 128 57 53 39 161 88 ^
393 1 92 72 151 33 183 164 42 112 32 10 5 93 97 18 112 71 92 168 116
179 186 65
137 68 134 3 82 9 19 121 148 65 160 173 158 167 90 66 180 172 99 86 83
86 26 16

10 135 23 81 137 125 17 19 25 183 73 7 104 185 79 165 119 119 176 59
75 18 5 59
132 26 21 127 119 80 94 83 52 61 24 44 4 50 55 115 29 70 140 99 159
130 54 15
42 89 114 173 28 50 16 33 53 47 58 180 19 64 78 49 30 186 104 32 44
157 85 1
149 85 97 114 77 63 58 50 89 80 21 36 84 72 60 152 25 87 108 15 138 54
106 77
97 23 42 55 149 181 62 106 43 132 31 138 111 176 93 40 10 96 128 105
114 43 98
4 146 20 107 120 14 113 181 101 1 19 155 113 57 64 117 131 6 22 186
138 184 36
61 29 36 175 107 6 61 81 114 20 180 42 169 102 26 73 80 151 153 68 4
11 98 139
6 175 34 146 93 154 88 128 139 79 35 2 135 119 46 129 91 86 179 51 18
169 141
108 49 103 46 72 21 70 49 64 151 96 185 175 130 111 85 58 32 101 40
101 131 136
139 170 70 10 169 175 73 41 69 177 45 47 92 145 11 155 22 22 56 21 13
166 48 21
109 131 40 174 178 104 128 9 149 51 132 122 173 119 23 24 67 109 74 32
158 110
117 74 123 6 118 151 130 150 167 34 55 58 146 54 72 112 105 57 160 82
123 159
153 159 39 19 7 8 33 87 12 50 146 150 37 104 71 126 133 108 145 68 38
38 90 43
75 148 122 43 2 39 154 54 163 147 113 91 29 109 35 75 182 87 35 155
170 127 80
185 118 60 95 142 71 165 28 84 168 25 144 178 31 81 41 160 25 53 143 ^
381 1 14 172 163 130 45 69 46 145 36 36 151 169 13 135 42 26 81 153 76
148 178
90 131 30 101 162 161 132 142 14 89 3 141 175 146 65 26 89 94 99 187 8
55 162 2
13 143 173 2 120 173 41 132 150 2 112 42 18 140 130 37 128 23 106 159
118 167
150 101 11 30 109 150 184 124 19 169 73 165 53 109 158 172 147 11 46
38 149 25
21 92 184 64 179 158 133 135 34 38 58 114 9 183 4 118 76 49 89 102 40
163 47 75
80 176 175 8 154 14 180 104 153 84 89 11 129 8 138 85 181 73 66 154
121 20 115
93 114 108 82 182 19 86 103 176 183 186 177 53 10 69 137 56 1 135 141
68 69 54
86 12 75 171 87 187 107 144 150 93 23 7 70 83 29 140 67 139 153 183
178 189 30
18 2 29 72 29 33 160 56 9 123 164 107 136 19 70 31 177 152 99 57 38 66
55 32 32
24 83 60 21 1 36 52 80 168 44 32 115 184 25 120 181 172 155 189 22 30
26 180 5
64 49 188 1 28 111 57 176 167 67 131 22 160 186 169 65 34 30 117 91 17
163 39

134 57 137 155 90 170 42 78 170 141 167 158 42 177 22 129 68 168 143
74 150 85
82 174 78 63 111 156 28 12 158 141 76 57 77 125 79 40 12 59 52 88 156
20 24 182
149 56 92 46 106 167 59 189 121 119 14 81 15 165 136 79 100 42 117 182
118 175
138 124 3 50 74 134 156 36 143 181 180 96 22 61 97 134 164 24 166 30
10 61 6 33
9 115 176 84 97 142 152 15 50 165 45 168 135 133 127 93 15 66 94 91 9
32 16 95
37 29 171 118 68 108 66 63 20 55 186 ^
396 1 97 10 5 160 132 18 83 73 159 141 55 120 31 131 141 102 138 3 52
9 148 189
56 99 165 146 2 140 64 157 100 121 113 102 24 21 24 123 16 10 139 152
19 109
146 116 122 92 43 7 75 62 49 170 58 26 117 125 45 147 5 125 136 155
118 2 26 50
161 21 182 127 153 45 68 19 121 22 94 104 144 166 35 166 23 176 75 188
53 114
93 148 177 66 151 16 156 121 180 98 48 179 95 63 107 97 45 1 23 97 57
23 132 53
79 97 124 19 188 18 118 37 22 106 94 60 122 25 47 176 110 184 109 79
164 120 43
153 89 90 128 109 12 168 64 110 82 117 192 41 171 163 164 76 53 12 101
68 54
117 78 59 143 84 32 122 104 65 148 31 52 123 130 24 135 38 151 95 142
93 101
159 62 37 148 110 70 127 51 85 181 89 11 142 105 91 107 50 27 29 102
36 43 165
79 6 84 99 118 128 30 26 3 143 17 68 66 139 187 15 165 158 24 88 20
179 174 81
79 18 106 47 74 150 104 179 162 59 131 62 116 166 82 184 188 97 136 41
136 167
140 143 175 72 42 174 148 75 74 71 86 14 16 191 117 11 63 119 119 186
182 13
171 49 182 80 167 73 147 119 107 164 41 57 181 168 29 114 26 184 121
101 80 76
65 34 49 24 44 191 38 25 91 5 58 140 99 153 124 54 184 42 83 108 155 4
20 191 3
29 35 40 180 7 40 66 25 12 180 86 26 38 139 73 1 137 79 85 102 71 39
28 44 77
56 190 12 66 72 30 152 7 87 84 190 126 48 100 77 91 186 36 31 143 163
50 94 25
126 1 114 105 170 93 22 173 90 116 87 96 19 74 4 122 183 83 120 177
113 169 181
77 182 1 149 186 ^
384 1 25 56 85 107 177 6 186 138 184 28 61 184 191 175 107 6 29 81 114
12 156 2
137 86 189 73 64 127 129 44 167 11 98 107 169 143 26 146 61 130 88 120
139 55 3
189 119 87 22 105 67 78 147 51 2 161 109 84 49 87 38 48 192 62 25 64
143 72 169

151 106 111 53 58 8 93 8 85 123 120 131 170 38 2 169 167 49 41 37 153
21 39 92
145 190 155 177 185 48 13 5 150 48 21 85 115 40 166 178 88 104 1 125
19 116 106
141 111 23 43 101 58 187 150 78 101 50 107 185 86 151 130 134 143 34
31 26 122
38 40 112 97 33 136 50 99 127 121 159 7 19 170 187 1 79 191 26 130 118
5 104 47
126 109 92 129 52 30 193 90 35 67 116 90 3 181 39 122 30 155 147 105
83 21 77
35 75 174 55 190 131 154 95 72 185 94 36 95 126 71 149 4 68 152 180
144 146 7
49 33 152 188 29 111 194 172 148 125 35 59 36 125 11 36 141 149 183
125 27 1 71
133 61 138 168 85 131 5 96 162 161 117 142 14 79 183 141 165 126 55 11
74 84 79
167 3 40 142 177 193 128 173 192 100 163 16 127 145 182 97 17 193 120
115 32
108 18 96 154 103 152 150 86 181 15 99 135 179 124 194 164 68 160 43
109 138
152 142 6 21 23 149 20 11 82 164 44 159 148 133 115 24 28 53 104 9 183
179 108
56 39 84 87 30 163 32 70 75 166 170 8 139 9 175 99 143 64 89 186 114
183 133 85
161 63 61 154 111 15 115 78 109 108 72 177 14 71 103 176 173 176 157
38 10 54
132 41 186 130 126 63 69 39 76 7 65 171 82 172 102 124 140 83 113 ^
396 1 189 70 83 29 132 67 115 145 151 170 181 14 192 2 187 40 5 33 160
48 183
91 132 83 112 177 62 189 153 128 91 49 38 50 55 24 8 8 51 36 13 175 28
36 56
160 12 107 168 25 104 157 156 139 165 14 30 192 164 195 56 41 180 159
28 79 41
144 167 51 115 188 136 154 145 65 2 22 93 83 191 147 23 110 57 137 131
66 138
26 78 146 125 159 150 34 177 22 129 68 168 111 74 118 77 50 174 54 47
87 140 20
178 158 133 60 33 45 93 47 8 170 59 52 56 124 20 16 150 117 24 84 46
106 151 51
158 173 113 111 196 73 15 133 136 71 100 10 85 182 110 143 114 92 185
42 42 118
156 20 135 173 156 96 180 37 73 134 156 182 150 196 192 45 172 17 167
91 168 68
89 134 120 190 173 34 141 37 152 111 117 103 61 7 34 62 67 191 190 8
87 21 195
139 86 44 76 66 55 186 23 31 87 183 5 155 122 191 68 53 149 136 35 115
26 121
131 102 123 3 42 197 133 179 56 84 165 136 190 130 49 157 80 101 103
97 197 11
19 118 1 5 119 137 9 104 193 141 116 122 72 38 7 70 47 44 155 43 16
107 125 40

127 193 105 131 145 108 185 6 50 141 1 167 127 153 45 53 192 121 12 79
84 129
151 15 166 196 156 60 188 53 109 83 148 162 46 146 6 136 111 160 88 38
169 85
63 87 97 35 194 3 82 52 13 132 43 59 92 109 4 178 3 118 37 195 96 94
60 112 15
47 166 110 179 99 69 149 100 43 138 69 70 123 89 12 158 54 90 72 117
182 31 166
153 164 76 33 2 101 63 44 112 58 39 143 74 32 107 89 55 143 31 47 123
125 4 135
18 141 80 137 88 81 154 187 ^
406 0 23 134 103 49 99 44 64 160 68 190 114 105 91 86 22 27 15 74 8 22
151 79
199 84 85 90 128 9 5 175 129 17 40 59 125 187 194 144 151 3 60 6 158
153 67 58
18 99 19 53 129 90 179 134 52 124 55 95 138 82 170 167 69 122 13 108
160 119
115 161 51 14 167 120 47 60 57 86 16 184 96 197 42 98 112 165 154 199
157 21
168 52 146 66 126 119 93 150 20 36 153 140 194 93 26 156 114 80 80 55
44 13 35
24 44 191 24 190 63 177 44 140 99 146 117 54 163 42 76 101 134 176 185
177 168
1 21 19 180 193 12 52 197 191 173 65 19 31 118 59 1 123 72 71 88 64 11
193 37
63 28 169 184 45 72 195 152 186 87 56 176 112 41 93 77 84 158 29 3 136
142 36
80 4 119 166 86 98 163 93 1 145 83 102 66 75 191 46 4 94 155 55 120
149 113 148
181 49 175 180 142 87 5 51 65 92 167 196 186 138 184 23 61 164 171 175
107 6 9
81 114 7 141 177 117 76 174 73 54 112 114 29 152 11 98 87 154 123 21
146 41 115
88 115 139 40 183 189 109 67 7 90 52 73 127 51 192 156 89 69 49 77 33
33 182 57
10 64 138 57 159 136 91 111 33 58 193 88 188 75 118 110 126 170 18 197
169 162
34 41 17 138 6 34 92 145 185 155 157 170 43 8 140 48 21 70 105 40 161
178 78 89
196 110 199 106 96 121 106 23 185 28 96 48 167 145 58 91 35 97 180 66
151 130
124 128 34 16 6 107 28 20 112 92 18 121 30 84 107 101 159 187 19 155
182 181 74
186 11 120 98 185 104 32 126 94 82 119 42 25 173 90 30 62 96 70 178
176 39 102
15 150 147 100 78 16 57 35 75 169 35 170 116 144 75 146 ^
409 1 185 70 12 95 110 71 133 183 52 136 148 144 114 186 17 25 144 164
5 79 178
172 124 117 19 43 20 93 174 36 125 117 151 109 3 164 55 101 37 122 152
77 131
168 88 162 161 93 142 14 63 167 141 149 94 39 190 50 68 47 135 198 16
110 153

177 104 173 192 68 147 179 119 137 166 73 180 169 88 91 24 76 10 80
146 79 128
150 62 149 194 83 111 171 124 170 156 60 152 27 109 106 120 134 201
184 202 149
12 198 66 132 12 127 132 133 83 8 12 45 88 9 183 155 92 24 23 76 63 14
163 8 62
67 150 162 8 115 1 167 91 127 32 89 162 90 159 125 85 129 47 53 154 95
7 115 54
101 108 56 169 6 47 103 176 157 160 125 14 10 30 124 17 178 122 102 55
69 15 60
202 49 171 74 148 94 92 124 67 200 184 70 83 29 127 67 100 140 131 165
176 4
182 2 167 20 193 33 160 43 173 71 112 68 97 157 57 169 138 113 86 44
38 40 55
19 196 201 31 21 8 165 23 26 41 155 195 183 102 158 25 94 142 146 129
150 9 30
177 154 195 51 36 175 139 28 59 31 124 167 41 105 173 121 134 130 65
185 17 78
78 181 137 13 95 57 137 116 51 118 16 78 131 115 154 145 29 177 22 129
68 168
91 74 98 72 30 174 39 37 72 130 15 163 158 128 50 18 25 73 27 191 150
59 52 36
104 20 11 130 97 4 79 46 106 141 46 138 163 108 106 191 68 15 113 136
66 100
193 65 182 105 123 99 72 180 37 22 108 156 10 130 168 141 96 160 22 58
134 151
162 140 181 187 35 157 7 147 76 163 58 84 129 100 190 153 24 126 32
142 96 107
88 41 2 14 42 52 186 170 3 82 11 180 119 66 29 56 66 50 171 3 11 77
163 5 150
112 128 ^
413 1 47 25 135 129 7 108 19 107 117 102 102 3 28 190 112 165 56 63
165 122 183
116 28 157 52 73 89 90 169 202 12 111 185 203 91 116 200 97 193 134
116 122 44
31 7 63 26 37 134 22 2 93 125 33 99 186 77 124 131 94 171 183 50 113
178 146
127 153 45 32 164 121 203 58 56 108 130 192 166 168 128 39 188 53 102
69 148
141 18 139 197 108 97 132 74 24 155 71 63 59 97 21 194 180 61 45 204
132 29 31
85 88 188 164 187 118 37 167 82 94 60 98 1 47 152 110 172 85 55 128 72
43 117
41 42 116 61 12 144 40 62 58 117 168 17 159 139 164 76 5 193 101 56 30
105 30
11 143 60 32 86 68 41 136 31 40 123 118 181 135 195 127 59 130 81 53
147 62 13
124 98 34 79 39 49 145 53 180 94 105 91 71 2 27 5 54 193 7 141 79 199
84 75 70
128 199 195 160 119 17 20 54 115 187 184 129 146 193 40 201 143 138 57
43 18 94

204 38 114 80 179 114 47 119 50 80 118 82 160 152 49 112 198 88 155
104 95 151
36 199 162 100 27 50 47 86 195 16 179 81 192 27 83 107 150 134 194 147
1 158 32
131 61 111 119 83 140 5 21 133 120 174 78 26 136 109 65 80 40 29 203
25 24 44
191 14 170 43 162 34 140 99 141 112 54 148 42 71 96 119 161 165 167
148 186 11
4 180 188 197 42 182 181 168 50 14 26 103 49 1 113 67 61 78 59 196 173
32 53 8
154 169 30 72 175 152 176 87 36 166 102 36 88 77 79 138 24 188 131 127
26 70
194 114 146 66 93 158 93 191 125 78 92 51 60 176 26 4 74 135 35 120
129 113 133
181 29 170 170 137 77 190 46 45 77 157 191 186 138 184 18 61 144 151
175 107 6
194 81 114 2 126 110 ^
427 1 85 60 150 73 38 88 90 5 128 11 98 55 130 91 13 146 9 91 88 107
139 16 159
189 93 35 191 66 28 65 95 51 184 148 57 45 49 61 25 9 166 49 194 64
130 33 143
112 67 111 1 58 177 80 164 59 110 94 118 170 194 197 169 154 10 41 193
114 190
26 92 145 177 155 125 146 35 200 124 48 21 46 89 40 153 178 62 65 196
86 175 90
80 89 98 23 169 4 88 32 135 137 26 75 11 81 172 34 151 130 108 104 34
200 182
83 12 196 112 84 202 97 206 60 75 69 159 163 19 131 174 157 66 178 195
104 66
161 104 8 126 70 66 103 26 17 141 90 22 54 64 38 146 168 39 70 199 142
147 92
70 8 25 35 75 161 3 138 92 128 43 59 185 55 205 95 100 71 123 173 42
126 128
144 94 176 205 20 139 149 198 59 168 172 109 112 9 33 10 73 154 36 115
97 131
99 196 144 45 81 22 112 142 72 131 148 83 162 161 78 142 14 53 157 141
139 74
29 180 35 58 27 115 198 1 90 138 167 89 173 192 48 137 159 114 132 156
58 160
154 68 76 19 56 5 70 141 64 113 150 47 129 184 73 96 166 124 155 151
55 147 17
109 86 100 129 201 164 192 149 7 193 56 112 200 107 122 133 63 206 2
40 78 9
183 140 82 4 13 71 48 4 163 201 57 62 140 157 8 100 204 162 86 117 12
89 147 75
144 120 85 109 37 48 154 85 2 115 39 96 108 46 164 1 32 103 176 147
150 105 207
10 15 119 2 173 117 87 50 69 50 202 39 171 69 133 89 72 114 57 195 179
70 83 29
122 67 85 135 111 160 171 202 172 2 147 183 33 160 38 163 51 92 53 82
137 52

149 123 98 81 39 38 30 55 14 186 196 11 6 3 155 18 16 26 150 180 168
97 148 25
84 127 136 119 135 4 30 162 144 195 46 31 170 119 28 190 ^
443 1 17 96 167 27 91 152 100 106 109 65 164 10 57 71 167 123 209 74
57 137 95
30 90 2 78 110 101 147 138 22 177 22 129 68 168 63 74 70 65 2 174 18
23 51 116
8 142 158 121 36 207 207 45 209 170 122 59 52 8 76 20 4 102 69 186 72
46 106
127 39 110 149 101 99 184 61 15 85 136 59 100 172 37 182 98 95 78 44
173 30 204
94 156 206 123 161 120 96 132 1 37 134 144 134 126 160 180 21 136 203
119 55
156 44 77 122 72 190 125 10 105 25 128 75 93 67 13 205 196 14 31 179
142 206 75
207 159 91 38 8 28 66 43 150 185 193 63 135 5 143 98 143 32 5 125 124
197 103
14 97 107 102 87 3 18 185 97 155 56 48 165 112 178 106 13 157 32 53 79
85 149
197 7 106 175 203 71 101 195 92 193 129 116 122 24 26 7 58 11 32 119 7
202 83
125 28 79 181 57 119 121 84 161 168 50 93 163 131 127 153 45 17 144
121 198 43
36 93 115 177 166 148 108 24 188 53 97 59 148 126 208 134 192 88 87
112 64 14
145 61 63 39 97 11 194 165 46 40 199 132 19 11 80 73 178 154 177 118
37 147 72
94 60 88 201 47 142 110 167 75 45 113 52 43 102 21 22 111 41 12 134 30
42 48
117 158 7 154 129 164 76 195 188 101 51 20 100 10 201 143 50 32 71 53
31 131 31
35 123 113 166 135 180 117 44 125 76 33 142 62 3 114 93 19 59 34 34
130 38 170
74 105 91 56 192 27 205 34 178 202 131 79 199 84 65 50 128 189 185 145
109 17
49 105 187 174 114 141 183 20 196 128 123 47 28 18 89 189 23 99 70 179
94 42
114 45 65 98 82 150 137 29 102 183 68 150 89 75 141 21 184 157 80 7 40
37 86
190 16 174 66 187 12 68 102 135 114 189 137 191 148 12 116 56 96 119
73 130 200
6 113 100 154 63 26 116 104 50 80 25 14 193 83 ^
436 1 24 44 191 211 138 11 138 18 140 99 133 104 54 124 42 63 88 95
137 133 151
116 162 208 193 180 180 173 26 158 165 160 26 6 18 79 33 1 97 59 45 62
51 172
141 24 37 189 130 145 6 72 143 152 160 87 4 150 86 28 80 77 71 106 16
164 123
103 10 54 178 106 114 34 85 150 93 175 93 70 76 27 36 152 207 4 42 103
3 120 97
113 109 181 210 162 154 129 61 166 38 13 53 141 183 186 138 184 10 61
112 119

175 107 6 170 81 114 207 102 125 65 50 135 73 28 73 75 203 113 11 98
35 115 71
8 146 202 76 88 102 139 1 144 189 83 15 181 51 13 60 75 51 179 143 37
30 49 51
20 207 156 44 184 64 125 18 133 97 52 111 194 58 167 75 149 49 105 84
113 170
179 197 169 149 208 41 178 99 180 21 92 145 172 155 105 131 30 208 200
114 48
21 31 79 40 148 178 52 50 196 71 160 80 70 69 93 23 159 202 83 22 115
132 6 65
209 71 167 14 151 130 98 89 34 190 167 68 2 181 112 79 192 82 191 45
55 49 159
148 19 116 169 142 61 173 185 94 46 146 104 206 126 55 56 93 16 12 121
90 17 49
44 18 126 163 39 50 189 137 147 87 65 3 5 35 75 156 196 118 77 118 23
54 185 40
195 95 90 71 113 163 32 116 108 144 74 166 190 15 134 134 188 39 158
172 94 107
212 23 53 134 36 105 77 111 89 186 124 35 61 7 102 132 67 131 128 78
162 161 63
142 14 43 147 141 129 54 19 170 20 48 7 95 198 199 70 123 157 74 173
192 28 127
139 109 127 146 43 140 139 48 61 14 36 60 136 49 98 150 32 109 174 63
81 161
124 140 146 50 142 7 109 66 80 124 201 144 182 149 2 188 46 92 185 87
112 133
43 201 205 35 68 9 183 125 72 197 3 66 33 207 163 191 52 57 130 152 8
85 204
157 81 107 205 187 ^
462 1 126 54 123 113 85 81 23 41 154 71 210 115 18 89 108 32 157 209
11 103 176
133 136 77 193 10 209 112 196 166 110 66 43 69 194 36 202 25 171 62
112 82 44
100 43 188 172 70 83 29 115 67 64 128 83 153 164 195 158 2 119 187 169
33 160
31 149 23 64 32 61 109 45 121 102 77 74 32 38 16 55 7 172 189 198 200
211 141
11 2 5 143 159 147 90 134 25 70 106 122 105 114 212 30 141 130 195 39
24 163 91
28 11 7 76 167 17 81 137 85 86 94 65 149 5 42 66 157 113 204 59 57 137
80 15 70
207 78 95 91 142 133 17 177 22 129 68 168 43 74 50 60 197 174 3 13 36
106 3 127
158 116 26 197 192 25 194 155 102 59 52 203 56 20 214 82 49 171 67 46
106 117
34 90 139 96 94 179 56 15 65 136 54 100 157 17 182 93 75 63 24 168 25
189 84
156 201 118 156 105 96 112 201 22 134 139 114 116 145 175 11 121 198
99 40 151
34 72 117 52 190 105 90 20 118 60 83 52 208 205 181 209 16 174 122 206
70 202

144 71 18 208 8 66 38 135 170 178 53 115 5 138 88 123 17 200 115 119
182 98 9
87 97 102 72 3 8 180 82 145 56 33 165 102 173 96 213 157 12 33 69 80
129 192 2
101 165 203 51 86 190 87 193 124 116 122 4 21 7 53 211 27 104 207 197
73 125 23
59 176 37 114 111 74 151 153 50 73 148 116 127 153 45 2 124 121 193 28
16 78
100 162 166 128 88 9 188 53 92 49 148 111 193 129 187 68 77 92 54 4
135 51 63
19 97 1 194 150 31 35 194 132 9 206 75 58 168 144 167 118 37 127 62 94
60 78
196 47 132 110 162 65 35 98 32 43 87 1 2 106 21 12 124 20 22 38 117
148 212 149
119 164 76 180 183 101 46 10 95 205 186 143 40 32 56 38 21 126 31 30
123 108
151 135 165 107 29 120 71 13 137 62 208 104 88 4 39 29 19 115 23 160
54 105 91
41 177 27 200 14 163 124 ^
453 0 115 79 199 84 49 18 128 173 169 121 93 17 186 41 89 187 158 90
133 167
206 188 104 99 31 4 18 81 165 217 75 54 179 62 34 106 37 41 66 82 134
113 215
86 159 36 142 65 43 125 215 160 149 48 193 24 21 86 182 16 166 42 179
206 44 94
111 82 181 121 167 132 198 92 48 72 119 57 114 184 200 81 68 122 39 26
84 96 26
80 1 208 177 217 24 44 191 206 118 209 123 8 140 99 128 99 54 109 42
58 83 80
122 113 141 96 147 203 183 180 175 158 16 143 155 155 11 1 13 64 23 1
87 54 35
52 46 157 121 19 27 174 115 130 209 72 123 152 150 87 202 140 76 23 75
77 66 86
11 149 118 88 44 168 101 94 14 80 145 93 165 73 65 66 12 21 137 192 4
22 83 201
120 77 113 94 181 195 157 144 124 51 151 33 211 38 131 178 186 138 184
5 61 92
99 175 107 6 155 81 114 207 87 105 45 40 120 73 18 58 60 193 98 11 98
15 100 51
3 146 187 61 88 97 139 204 129 189 73 213 171 36 216 55 55 51 174 138
17 15 49
41 15 197 146 39 174 64 120 3 123 82 37 111 179 58 157 70 134 39 100
74 108 170
164 197 169 144 198 41 163 84 170 16 92 145 167 155 85 116 25 208 200
104 48 21
16 69 40 143 178 42 35 196 56 145 70 60 49 88 23 149 192 78 12 95 127
204 55
199 61 162 212 151 130 88 74 34 180 152 53 210 166 112 74 182 67 176
30 35 29
159 133 19 101 164 127 56 168 175 84 26 131 104 196 126 40 46 83 6 7
101 90 12

44 24 216 106 158 39 30 179 132 147 82 60 216 203 35 75 151 181 98 62
108 3 49
185 25 185 95 80 71 103 153 22 106 88 144 54 156 175 10 129 119 178 19
148 172
79 102 207 13 208 33 114 36 95 57 91 79 176 104 25 41 210 92 122 62
131 108 73
162 161 48 142 14 33 137 141 119 34 9 160 5 38 205 75 198 189 50 108
112 ^
454 1 53 173 192 113 111 102 120 132 22 112 118 20 40 7 8 213 46 129
28 77 150
11 81 160 49 60 154 124 119 139 43 135 213 109 38 52 117 201 116 168
149 215
181 32 64 164 59 98 133 15 194 198 28 54 9 183 104 58 176 209 59 12
200 163 177
45 50 116 145 8 64 204 150 74 93 184 89 111 39 108 108 85 61 13 36 154
61 210
115 3 84 108 22 152 209 216 103 176 123 126 57 183 10 199 107 186 161
105 51 38
69 184 26 202 15 171 57 97 77 24 90 33 183 167 70 83 29 110 67 49 123
63 148
159 190 148 2 99 172 159 33 160 26 139 3 44 17 46 89 40 101 87 62 69
27 38 6 55
2 162 184 183 190 211 131 6 212 210 138 144 132 85 124 25 60 91 112 95
99 212
30 126 120 195 34 19 158 71 28 211 217 56 167 7 71 122 70 66 79 65 134
27 61
147 103 199 44 57 137 65 50 202 78 80 81 137 128 12 177 22 129 68 168
23 74 30
55 182 174 208 3 21 96 218 112 158 111 16 187 177 5 179 140 82 59 52
188 36 20
214 62 29 156 62 46 106 107 29 70 129 91 89 174 51 15 45 136 49 100
142 217 182
88 55 48 4 163 20 174 74 156 196 113 151 90 96 92 191 7 134 134 94 106
130 170
1 106 193 79 25 146 24 67 112 32 190 85 210 75 15 108 45 73 37 193 205
166 194
1 169 102 206 65 197 129 51 218 198 208 66 33 120 155 163 43 95 5 133
78 103 2
185 105 114 167 93 4 77 87 102 57 3 218 175 67 135 56 18 165 92 168 86
203 157
212 13 59 75 109 187 217 96 155 203 31 71 185 82 193 119 116 122 204
16 7 48
201 22 89 197 192 63 125 18 39 171 17 109 101 64 141 138 50 53 133 101
127 153
45 207 104 121 188 13 216 63 85 147 166 108 68 214 188 53 87 39 148 96
178 124
182 48 67 72 44 214 125 41 63 219 97 211 194 135 16 30 189 132 219 191
70 43
158 181 ^
475 0 153 118 37 99 48 94 60 64 189 47 118 110 155 51 21 77 4 43 66
195 196 99

215 12 110 6 216 24 117 134 205 142 105 164 76 159 176 101 39 218 88
184 165
143 26 32 35 17 7 119 31 23 123 101 130 135 144 93 8 113 64 207 130 62
201 90
81 205 11 22 220 94 2 146 26 105 91 20 156 27 193 208 142 178 107 79
199 84 41
2 128 165 161 109 85 17 174 37 81 187 150 78 129 159 194 184 92 87 23
214 18 77
153 209 63 46 179 46 30 102 33 29 50 82 126 101 203 78 147 20 138 53
27 117 207
148 145 32 181 16 13 86 178 16 162 30 175 198 32 90 99 66 177 113 155
124 186
80 44 60 119 49 106 176 192 65 52 106 27 26 68 92 14 80 211 200 169
213 24 44
191 202 102 197 111 140 99 124 95 54 97 42 54 79 68 110 97 133 80 135
199 175
180 171 146 8 131 147 151 221 219 9 52 15 1 79 50 27 44 42 145 105 15
19 162
103 118 201 72 107 152 142 87 190 132 68 19 71 77 62 70 7 137 114 76
214 36 160
97 78 220 76 141 93 157 57 61 58 9 125 180 4 6 67 189 120 61 113 82
181 183 153
136 120 43 139 29 199 26 123 174 186 138 184 1 61 76 83 175 107 6 143
81 114
207 75 89 29 32 108 73 10 46 48 185 86 11 98 221 88 35 221 146 175 49
88 93 139
196 117 189 65 201 163 24 208 51 39 51 170 134 1 3 49 33 11 189 138 35
166 64
116 213 115 70 25 111 167 58 149 66 122 31 96 66 104 170 152 197 169
140 190 41
151 72 162 12 92 145 163 155 69 104 21 208 200 96 48 21 4 61 40 139
178 34 23
196 44 133 62 52 33 84 23 141 184 74 4 79 123 192 47 191 53 158 200
151 130 80
62 34 172 140 41 206 154 112 70 174 55 164 18 19 13 159 121 19 89 160
115 52
164 167 76 10 119 104 188 126 28 38 75 220 3 85 90 8 40 8 204 90 154
39 14 171
128 147 78 56 216 191 35 75 147 169 82 50 100 209 45 185 13 177 95 150
^
471 0 71 89 139 8 92 60 144 26 142 154 3 122 98 164 216 134 172 58 95
200 224
201 5 86 36 81 29 63 65 162 76 11 13 196 78 108 55 131 80 66 162 161
27 142 14
19 123 141 105 6 220 146 209 24 184 47 198 175 22 87 133 38 173 192
205 103 91
97 115 122 7 92 103 25 2 213 213 36 124 13 62 150 221 61 150 39 45 149
124 104
134 38 130 208 109 18 32 112 201 96 158 149 215 176 22 44 149 39 88
133 220 189
193 23 44 9 183 89 48 161 204 54 222 195 163 167 40 45 106 140 8 49
204 145 69

83 169 89 96 24 93 103 85 41 3 31 154 51 210 115 213 79 108 12 147 209
206 103
176 113 116 37 173 10 189 102 176 156 100 36 33 69 174 16 202 5 171 52
82 72 4
80 23 178 162 70 83 29 105 67 34 118 43 143 154 185 138 2 79 157 149
33 160 21
129 208 24 2 31 69 35 81 72 47 64 22 38 221 55 222 152 179 168 180 211
121 1
207 200 133 129 117 80 114 25 50 76 102 85 84 212 30 111 110 195 29 14
153 51
28 196 212 36 167 222 61 107 55 46 64 65 119 220 12 56 137 93 194 29
57 137 50
210 30 197 78 65 71 132 123 7 177 22 129 68 168 3 74 10 50 167 174 198
218 6 86
218 97 158 106 6 177 162 210 164 125 62 59 52 173 16 20 214 42 9 141
57 46 106
97 24 50 119 86 84 169 46 15 25 136 44 100 127 202 182 83 35 33 209
158 15 159
64 156 191 108 146 75 96 72 181 217 134 129 74 96 115 165 216 91 188
59 10 141
14 62 107 12 190 65 205 60 10 98 30 63 22 178 205 151 179 211 164 82
206 60 192
114 31 203 188 193 66 28 105 140 148 33 75 5 128 68 83 212 170 95 109
152 88
224 67 77 102 42 3 213 170 52 125 56 3 165 82 163 76 193 157 197 218
49 70 89
182 217 91 145 203 11 56 180 77 193 114 116 122 189 11 7 43 191 17 74
187 187
53 125 13 19 166 222 104 91 54 131 123 50 33 118 86 127 167 ^
480 1 45 193 76 121 181 219 195 42 64 126 166 80 40 200 188 53 80 25
148 75 157
117 175 20 53 44 30 207 111 27 63 198 97 204 194 114 222 23 182 132
212 170 63
22 144 120 143 118 37 79 38 94 60 54 184 47 108 110 150 41 11 62 211
43 51 180
181 94 200 12 100 223 201 14 117 124 200 137 95 164 76 144 171 101 34
213 83
169 150 143 16 32 20 2 224 114 31 18 123 96 115 135 129 83 220 108 59
192 125
62 196 80 76 195 218 17 210 79 214 136 6 105 91 5 141 27 188 193 127
168 97 79
199 84 31 209 128 155 151 94 75 17 159 32 71 187 140 63 124 149 179
179 77 72
13 204 18 72 138 199 48 36 179 26 25 97 28 14 30 82 116 86 188 68 132
133 38 7
107 197 133 140 12 166 6 3 86 173 16 157 15 170 188 17 85 84 46 172
103 140 114
171 65 39 45 119 39 96 166 182 45 32 86 12 26 48 87 226 80 201 190 159
208 24
44 191 197 82 182 96 217 140 99 119 90 54 82 42 49 74 53 95 77 123 60
120 194

165 180 166 131 225 116 137 146 211 219 4 37 5 1 69 45 17 34 37 130 85
10 9 147
88 103 191 72 87 152 132 87 175 122 58 14 66 77 57 50 2 122 109 61 209
26 150
92 58 205 71 136 93 147 37 56 48 212 221 110 165 4 213 47 174 120 41
113 67 181
168 148 126 115 33 124 24 184 11 113 169 186 138 184 223 61 56 63 175
107 6 128
81 114 207 60 69 9 22 93 73 31 33 175 71 11 98 206 73 15 221 146 160
34 88 88
139 186 102 189 55 186 153 9 198 46 19 51 165 129 208 215 49 23 6 179
128 30
156 64 111 203 105 55 10 111 152 58 139 61 107 21 91 56 99 170 137 197
169 135
180 41 136 57 152 7 92 145 158 155 49 89 16 208 200 86 48 21 216 51 40
134 178
24 8 196 29 118 52 42 13 79 23 131 174 69 221 59 118 177 37 181 43 153
185 151
130 70 47 34 162 125 26 201 139 112 65 164 40 149 3 226 220 159 106 19
74 155
100 47 159 157 193 ^
471 0 211 98 104 174 126 7 24 61 213 225 57 90 1 33 209 183 62 147 39
215 157
121 147 71 49 216 170 35 75 140 148 54 29 86 188 38 185 221 163 95 58
71 81 131
84 44 144 10 134 142 228 118 86 156 204 126 172 46 91 196 220 197 218
70 36 73
13 47 57 154 60 3 226 188 70 100 51 131 64 62 162 161 15 142 14 11 115
141 97
219 216 138 201 16 172 31 198 167 6 75 125 26 173 192 193 95 75 93 111
114 224
76 91 213 13 227 201 213 28 120 1 50 150 213 45 142 31 33 145 124 92
130 34 126
204 109 2 16 108 201 80 150 149 215 172 14 28 137 23 80 133 208 185
189 19 36 9
183 77 40 149 200 50 214 191 163 159 36 41 98 136 8 37 204 141 65 75
157 89 84
12 81 99 85 25 224 27 154 43 210 115 205 75 108 4 143 209 198 103 176
105 108
21 165 10 181 98 168 152 96 24 29 69 166 8 202 226 171 48 70 68 217 72
15 174
158 70 83 29 101 67 22 114 27 139 150 181 130 2 63 145 141 33 160 17
121 196 8
219 19 53 31 65 60 35 60 18 38 217 55 222 144 175 156 172 211 113 226
203 192
129 117 105 76 106 25 42 64 94 77 72 212 30 99 102 195 25 10 149 35 28
184 208
20 167 218 53 95 43 30 52 65 107 220 52 129 85 190 17 57 137 38 202 14
193 78
53 63 128 119 3 177 22 129 68 168 216 74 223 46 155 174 190 214 223 78
218 85

158 102 227 169 150 198 152 113 46 59 52 161 20 214 26 222 129 53 46
106 89 20
34 111 82 80 165 42 15 9 136 40 100 115 190 182 79 19 21 197 154 11
147 56 156
187 104 142 63 96 56 173 209 134 125 58 88 103 161 212 79 184 43 227
137 6 58
103 225 190 49 201 48 6 90 18 55 10 166 205 139 167 203 160 66 206 56
188 102
15 191 180 181 66 24 93 128 136 25 59 5 124 60 67 204 158 87 105 140
84 224 59
69 102 30 3 209 166 40 117 56 220 165 74 159 68 185 157 185 206 41 66
167 ^
490 1 176 217 85 133 203 218 38 174 71 193 108 116 122 171 5 7 37 179
11 56 175
181 41 125 7 226 160 204 98 79 42 119 105 50 9 100 68 127 153 45 185
60 121 177
211 183 30 52 114 166 64 24 192 188 53 76 17 148 63 145 113 171 4 45
28 22 203
103 19 63 186 97 200 194 102 214 19 178 132 208 158 59 10 136 112 135
118 37 63
30 94 60 46 180 47 100 110 146 33 3 50 199 43 39 168 169 90 188 12 92
219 189 6
117 116 196 133 87 164 76 132 167 101 30 209 79 157 138 143 8 32 8 221
220 110
31 14 123 92 103 135 117 75 212 104 55 180 121 62 192 72 72 187 206 13
202 67
206 128 221 105 91 224 129 27 184 181 115 160 89 79 199 84 23 197 128
147 143
82 67 17 147 28 63 187 132 51 120 141 167 175 65 60 5 196 18 68 126
191 36 28
179 10 21 93 24 2 14 82 108 74 176 60 120 215 129 26 222 99 189 121
136 227 154
229 226 86 169 16 153 3 166 180 5 81 72 30 168 95 128 106 159 53 35 33
119 31
88 158 174 29 16 70 26 32 83 218 80 193 182 151 204 24 44 191 193 66
170 84 213
140 99 115 86 54 70 42 45 70 41 83 61 115 44 108 190 157 180 162 119
221 104
129 142 203 219 25 228 1 61 41 9 26 33 118 69 6 1 135 76 91 183 72 71
152 124
87 163 114 50 10 62 77 53 34 229 110 105 49 205 18 142 88 42 193 67
132 93 139
21 52 40 204 213 98 153 4 201 31 162 120 25 113 55 181 156 144 118 111
25 112
20 172 230 105 165 186 138 184 223 61 40 47 175 107 6 116 81 114 207
48 53 224
14 81 73 223 19 21 167 59 11 98 194 61 230 221 146 148 22 88 84 139
178 90 189
47 174 145 228 190 42 3 51 161 125 196 207 49 15 2 171 120 26 148 64
107 195 97
43 229 111 140 58 131 57 95 13 87 48 95 170 125 197 169 131 172 41 124
45 144 3

92 145 154 155 33 77 12 208 200 78 48 21 208 43 40 130 178 16 227 196
17 106 44
34 228 75 23 123 166 65 217 43 114 165 29 173 35 200 ^
479 0 167 151 130 58 29 34 150 107 8 195 121 112 59 152 22 131 218 208
202 159
88 19 56 149 82 41 153 145 54 199 86 104 166 126 228 16 53 209 225 41
90 230 29
197 171 46 143 39 203 149 117 147 67 45 216 158 35 75 136 136 38 17 78
176 34
185 213 155 95 50 71 73 123 225 76 28 144 227 126 130 228 114 74 148
192 118
172 34 87 192 216 193 206 54 36 65 230 31 49 146 44 228 214 180 62 92
47 131 48
58 162 161 3 142 14 3 107 141 89 207 212 130 193 8 160 15 198 159 223
63 117 14
173 192 181 87 59 89 107 106 216 60 79 201 1 227 189 213 20 116 222 38
150 205
29 134 23 21 141 124 80 126 30 122 200 109 219 104 201 64 142 149 215
168 6 12
125 7 72 133 196 181 185 15 28 9 183 65 32 137 196 46 206 187 163 151
32 37 90
132 8 25 204 137 61 67 145 89 72 69 95 85 9 220 23 154 35 210 115 197
71 108
229 139 209 190 103 176 97 100 5 157 10 173 94 160 148 92 12 25 69 158
202 222
171 44 58 64 205 64 7 170 154 70 83 29 97 67 10 110 11 135 146 177 122
2 47 133
133 33 160 13 113 184 225 211 7 37 27 49 48 23 56 14 38 213 55 222 136
171 144
164 211 105 226 199 184 125 105 93 72 98 25 34 52 86 69 60 212 30 87
94 195 21
6 145 19 28 172 204 4 167 214 45 83 31 14 40 65 95 220 221 48 121 77
186 5 57
137 26 194 231 189 78 41 55 124 115 232 177 22 129 68 168 204 74 211
42 143 174
182 210 215 70 218 73 158 98 223 161 138 186 140 101 30 59 52 149 217
20 214 10
210 117 49 46 106 81 16 18 103 78 76 161 38 15 226 136 36 100 103 178
182 75 3
9 185 150 7 135 48 156 183 100 138 51 96 40 165 201 134 121 42 80 91
157 208 67
180 27 219 133 231 54 99 213 190 33 197 36 2 82 6 47 231 154 205 127
155 195
156 50 206 52 184 90 232 179 172 169 66 20 81 116 124 17 43 5 120 52
51 196 146
79 101 128 80 224 65 ^
503 1 55 102 9 3 202 159 19 103 56 206 165 60 152 54 171 157 164 185
27 59 45
171 217 80 123 203 203 23 169 66 193 103 116 122 156 7 32 169 6 41 165
176 31
125 2 211 155 189 93 69 32 109 90 50 225 85 53 127 153 45 175 40 121
172 201

168 15 37 99 166 44 4 182 188 53 71 7 148 48 130 108 166 220 35 8 12
198 93 9
63 171 97 195 194 87 204 14 173 132 203 143 54 231 126 102 125 118 37
43 20 94
60 36 175 47 90 110 141 23 229 35 184 43 24 153 154 85 173 12 82 214
174 232
117 106 191 128 77 164 76 117 162 101 25 204 74 142 123 143 234 32 229
211 215
105 31 9 123 87 88 135 102 65 202 99 50 165 116 62 187 62 67 177 191 8
192 52
196 118 206 105 91 214 114 27 179 166 100 150 79 79 199 84 13 182 128
137 133
67 57 17 132 23 53 187 122 36 115 131 152 170 50 45 231 186 18 63 111
181 21 18
179 226 16 88 19 223 230 82 98 59 161 50 105 200 124 11 207 89 179 106
131 212
139 224 221 86 164 16 148 224 161 170 226 76 57 10 163 85 113 96 144
38 30 18
119 21 78 148 164 9 232 50 221 26 12 78 208 80 183 172 141 199 24 44
191 188 46
155 69 208 140 99 110 81 54 55 42 40 65 26 68 41 105 24 93 185 147 180
157 104
216 89 119 137 193 219 231 10 223 1 51 36 235 16 28 103 49 1 227 120
61 76 173
72 51 152 114 87 148 104 40 5 57 77 48 14 229 95 100 34 200 8 132 83
22 178 62
127 93 129 1 47 30 194 203 83 138 4 186 11 147 120 5 113 40 181 141
139 108 106
15 97 15 157 220 95 160 186 138 184 223 61 20 27 175 107 6 101 81 114
207 33 33
209 4 66 73 218 4 6 157 44 11 98 179 46 215 221 146 133 7 88 79 139
168 75 189
37 159 135 218 180 37 219 51 156 120 181 197 49 5 233 161 110 21 138
64 102 185
87 28 219 111 125 58 121 52 80 3 82 38 90 170 110 197 169 126 162 41
109 30 134
234 92 145 149 155 13 62 7 208 200 68 48 21 198 33 40 125 178 6 217
196 2 91 34
24 213 70 23 113 161 ^
470 0 58 210 15 107 144 15 159 21 142 152 151 130 48 14 34 140 92 231
190 106
112 54 142 7 116 208 193 187 159 73 19 41 144 67 36 148 135 44 184 71
104 156
126 218 6 43 204 225 21 90 230 24 182 156 26 138 39 188 139 112 147 62
40 216
143 35 75 131 121 18 2 68 161 29 185 203 145 95 40 71 63 113 220 66 8
144 212
116 115 228 109 59 138 177 108 172 19 82 187 211 188 191 34 36 55 215
11 39 136
24 223 199 170 52 82 42 131 28 53 162 161 226 142 14 231 97 141 79 192
207 120

183 236 145 233 198 149 208 48 107 237 173 192 166 77 39 84 102 96 206
40 64
186 224 227 174 213 10 111 212 23 150 195 9 124 13 6 136 124 65 121 25
117 195
109 204 218 99 201 44 132 149 215 163 234 230 110 225 62 133 181 176
180 10 18
9 183 50 22 122 191 41 196 182 163 141 27 32 80 127 8 10 204 132 56 57
130 89
57 223 54 90 85 227 215 18 154 25 210 115 187 66 108 224 134 209 180
103 176 87
90 223 147 10 163 89 150 143 87 235 20 69 148 228 202 217 171 39 43 59
190 54
235 165 149 70 83 29 92 67 233 105 229 130 141 172 112 2 27 118 123 33
160 8
103 169 210 201 230 17 22 29 33 8 51 9 38 208 55 222 126 166 129 154
211 95 226
194 174 120 90 78 67 88 25 24 37 76 59 45 212 30 72 84 195 16 1 140
237 28 157
199 222 167 209 35 68 16 232 25 65 80 220 211 43 111 67 181 228 57 137
11 184
216 184 78 26 45 119 110 232 177 22 129 68 168 189 74 196 37 128 174
172 205
205 60 218 58 158 93 218 151 123 171 125 86 10 59 52 134 202 20 214
228 195 102
44 46 106 71 11 236 93 73 71 156 33 15 211 136 31 100 88 163 182 70
221 232 170
145 2 120 38 156 178 95 133 36 96 20 155 191 134 116 22 70 76 152 203
52 175 7
209 128 226 49 94 198 190 13 192 21 235 72 229 37 221 139 205 112 140
185 151
30 206 47 179 75 217 101 ^
502 0 158 148 66 13 60 95 103 3 15 5 113 38 23 182 125 65 94 107 73
224 37 47
102 237 3 198 155 7 95 56 198 165 52 148 46 163 157 152 173 19 55 29
167 217 76
115 203 191 11 165 62 193 99 116 122 144 236 7 28 161 2 29 157 172 23
125 238
199 151 177 89 61 24 101 78 50 213 73 41 127 153 45 167 24 121 168 193
156 3 25
87 166 28 228 174 188 53 67 239 148 36 118 104 162 208 27 232 4 194 85
1 63 159
97 191 194 75 196 10 169 132 199 131 50 223 118 94 117 118 37 27 12 94
60 28
171 47 82 110 137 15 225 23 172 43 12 141 142 81 161 12 74 210 162 228
117 98
187 124 69 164 76 105 158 101 21 200 70 130 111 143 230 32 221 203 211
101 31 5
123 83 76 135 90 57 194 95 46 153 112 62 183 54 63 169 179 4 184 40
188 110 194
105 91 206 102 27 175 154 88 142 71 79 199 84 5 170 128 129 125 55 49
17 120 19

45 187 114 24 111 123 140 166 38 33 227 178 18 59 99 173 9 10 179 214
12 84 15
215 218 82 90 47 149 42 93 188 120 239 195 81 171 94 127 200 127 220
217 86 160
16 144 216 157 162 218 72 45 234 159 77 101 88 132 26 26 6 119 13 70
140 156
233 220 34 213 26 236 74 200 80 175 164 133 195 24 44 191 184 30 143
57 204 140
99 106 77 54 43 42 36 61 14 56 25 97 8 81 181 139 180 153 92 212 77
111 133 185
219 231 238 219 1 43 32 231 8 24 91 33 237 223 108 49 64 165 72 35 152
106 87
136 96 32 1 53 77 44 238 229 83 96 22 196 124 79 6 166 58 123 93 121
225 43 22
186 195 71 126 4 174 235 135 120 229 113 28 181 129 135 100 102 7 85
11 145 212
87 156 186 138 184 223 61 4 11 175 107 6 89 81 114 207 21 17 197 236
54 73 214
232 234 149 32 11 98 167 34 203 221 146 121 235 88 75 139 160 63 189
29 147 127
210 172 33 207 51 152 116 169 189 49 237 233 153 102 17 130 64 98 177
79 16 211
111 113 58 113 48 68 235 78 30 86 170 98 197 169 122 154 41 97 18 126
234 92
145 145 155 237 50 209 ^
481 1 208 200 56 48 21 186 21 40 119 178 236 205 196 226 73 22 12 195
64 23 101
144 54 206 241 103 132 7 151 13 138 140 151 130 40 2 34 132 80 223 186
94 112
50 134 237 104 200 181 175 159 61 19 29 140 55 32 144 127 36 172 59
104 148 126
210 240 35 200 225 5 90 230 20 170 144 10 134 39 176 131 108 147 58 36
216 131
35 75 127 109 2 232 60 149 25 185 195 137 95 32 71 55 105 216 58 234
144 200
108 103 228 105 47 130 165 100 172 7 78 183 207 184 179 18 36 47 203
237 31 128
8 219 187 162 44 74 38 131 12 49 162 161 218 142 14 227 89 141 71 180
203 112
175 232 133 221 198 141 196 36 99 229 173 192 154 69 23 80 98 88 198
24 52 174
216 227 162 213 2 107 204 11 150 187 235 116 5 236 132 124 53 117 21
113 191
109 192 206 95 201 28 124 149 215 159 230 218 98 213 54 133 169 172
176 6 10 9
183 38 14 110 187 37 188 178 163 133 23 28 72 123 8 240 204 128 52 49
118 89 45
215 42 86 85 215 211 14 154 17 210 115 179 62 108 220 130 209 172 103
176 79 82
211 139 10 155 85 142 139 83 227 16 69 140 224 202 213 171 35 31 55
178 46 231

161 145 70 83 29 88 67 225 101 217 126 137 168 104 2 11 106 115 33 160
4 95 157
198 193 222 1 18 13 21 238 47 5 38 204 55 222 118 162 117 146 211 87
226 190
166 116 78 66 63 80 25 16 25 68 51 33 212 30 60 76 195 12 239 136 225
28 145
195 210 167 205 27 56 4 220 13 65 68 220 203 39 103 59 177 220 57 137
241 176
204 180 78 14 37 115 106 232 177 22 129 68 168 177 74 184 33 116 174
164 201
197 52 218 46 158 89 214 143 111 159 113 74 236 59 52 122 190 20 214
216 183 90
40 46 106 63 7 224 85 69 67 152 29 15 199 136 27 100 76 151 182 66 209
224 158
141 240 108 30 156 174 91 129 24 96 4 147 183 134 112 6 62 64 148 199
40 171
233 201 124 222 45 90 186 190 239 188 9 235 64 221 29 213 127 178 ^
508 1 94 122 173 145 6 206 41 173 57 199 146 150 136 66 9 48 83 91 239
243 5
109 30 7 174 113 57 90 95 69 224 29 39 102 229 3 194 151 239 87 56 190
165 44
144 38 155 157 140 161 11 51 13 163 217 72 107 203 179 243 161 58 193
95 116
122 132 236 7 24 153 242 17 149 168 15 125 238 187 147 165 85 53 16 93
66 50
201 61 29 127 153 45 159 8 121 164 185 144 235 13 75 166 12 216 166
188 53 63
235 148 24 106 100 158 196 19 220 240 190 77 237 63 147 97 187 194 63
188 6 165
132 195 119 46 215 110 86 109 118 37 11 4 94 60 20 167 47 74 110 133 7
221 11
160 43 129 130 77 149 12 66 206 150 224 117 90 183 120 61 164 76 93
154 101 17
196 66 118 99 143 226 32 213 195 207 97 31 1 123 79 64 135 78 49 186
91 42 141
108 62 179 46 59 161 167 176 28 180 102 182 105 91 198 90 27 171 142
76 134 63
79 199 84 241 158 128 121 117 43 41 17 108 15 37 187 106 12 107 115
128 162 26
21 223 170 18 55 87 165 241 2 179 202 8 80 11 207 206 82 82 35 137 34
81 176
116 231 183 73 163 82 123 188 115 216 213 86 156 16 140 208 153 154
210 68 33
222 155 69 89 80 120 14 22 238 119 5 62 132 148 221 208 18 205 26 224
70 192 80
167 156 125 191 24 44 191 180 14 131 45 200 140 99 102 73 54 31 42 32
57 2 44 9
89 236 69 177 131 180 149 80 208 65 103 129 177 219 231 230 215 1 35
28 227 20
79 17 237 219 96 37 52 157 72 19 152 98 87 124 88 24 241 49 77 40 226
229 71 92

10 192 236 116 75 234 154 54 119 93 113 213 39 14 178 187 59 114 4 162
223 123
120 217 113 16 181 117 131 92 98 243 73 7 133 204 79 152 186 138 184
223 61 232
239 175 107 6 77 81 114 207 9 1 185 232 42 73 210 224 226 141 20 11 98
155 22
191 221 146 109 227 88 71 139 152 51 189 21 135 119 202 164 29 195 51
148 112
157 181 49 233 233 145 94 13 122 64 94 169 71 4 203 111 101 58 105 44
56 231 74
22 82 170 86 197 169 118 146 41 85 6 118 234 92 145 141 149 ^
484 1 219 32 243 208 200 48 48 21 178 13 40 115 178 232 197 196 218 61
14 4 183
60 23 93 136 50 202 229 99 120 245 143 5 134 128 151 130 32 236 34 124
68 215
182 82 112 46 126 229 92 192 169 163 159 49 19 17 136 43 28 140 119 28
160 47
104 140 126 202 236 27 196 225 235 90 230 16 158 132 240 130 39 164
123 104 147
54 32 216 119 35 75 123 97 232 224 52 137 21 185 187 129 95 24 71 47
97 212 50
222 144 188 100 91 228 101 35 122 153 92 172 241 74 179 203 180 167 2
36 39 191
225 23 120 238 215 175 154 36 66 34 131 242 45 162 161 210 142 14 223
81 141 63
168 199 104 167 228 121 209 198 133 184 24 91 221 173 192 142 61 7 76
94 80 190
8 40 162 208 227 150 213 240 103 196 245 150 179 223 108 243 228 128
124 41 113
17 109 187 109 180 194 91 201 12 116 149 215 155 226 206 86 201 46 133
157 168
172 2 2 9 183 26 6 98 183 33 180 174 163 125 19 24 64 119 8 232 204
124 48 41
106 89 33 207 30 82 85 203 207 10 154 9 210 115 171 58 108 216 126 209
164 103
176 71 74 199 131 10 147 81 134 135 79 219 12 69 132 220 202 209 171
31 19 51
166 38 227 157 141 70 83 29 84 67 217 97 205 122 133 164 96 2 241 94
107 33 160
87 145 186 185 214 231 14 243 9 230 43 1 38 200 55 222 110 158 105 138
211 79
226 186 158 112 66 54 59 72 25 8 13 60 43 21 212 30 48 68 195 8 239
132 213 28
133 191 198 167 201 19 44 238 208 1 65 56 220 195 35 95 51 173 212 57
137 233
168 192 176 78 2 29 111 102 232 177 22 129 68 168 165 74 172 29 104
174 156 197
189 44 218 34 158 85 210 135 99 147 101 62 224 59 52 110 178 20 214
204 171 78
36 46 106 55 3 212 77 65 63 148 25 15 187 136 23 100 64 139 182 62 197
216 146

137 240 96 22 156 170 87 125 12 96 234 139 175 134 108 236 54 52 144
195 28 167
221 193 120 218 41 86 174 190 227 184 243 235 56 213 21 205 115 205 71
^
506 0 110 165 141 238 206 37 169 45 187 134 142 124 66 5 36 71 79 235
231 5 105
22 239 166 101 49 86 83 65 224 21 31 102 221 3 190 147 231 79 56 182
165 36 140
30 147 157 128 149 3 47 245 159 217 68 99 203 167 235 157 54 193 91
116 122 120
236 7 20 145 242 5 141 164 7 125 238 175 143 153 81 45 8 85 54 50 189
49 17 127
153 45 151 240 121 160 177 132 227 1 63 166 244 204 158 188 53 59 231
148 12 94
96 154 184 11 208 236 186 69 233 63 135 97 183 194 51 180 2 161 132
191 107 42
207 102 78 101 118 37 243 244 94 60 12 163 47 66 110 129 247 217 247
148 43 236
117 118 73 137 12 58 202 138 220 117 82 179 116 53 164 76 81 150 101
13 192 62
106 87 143 222 32 205 187 203 93 31 245 123 75 52 135 66 41 178 87 38
129 104
62 175 38 55 153 155 244 168 16 172 94 170 105 91 190 78 27 167 130 64
126 55
79 199 84 237 146 128 113 109 31 33 17 96 11 29 187 98 103 107 116 158
14 9 219
162 18 51 75 157 233 242 179 190 4 76 7 199 194 82 74 23 125 26 69 164
112 223
171 65 155 70 119 176 103 212 209 86 152 16 136 200 149 146 202 64 21
210 151
61 77 72 108 2 18 230 119 245 54 124 140 209 196 2 197 26 212 66 184
80 159 148
117 187 24 44 191 176 246 119 33 196 140 99 98 69 54 19 42 28 53 238
32 241 81
224 57 173 123 180 145 68 204 53 95 125 169 219 231 222 211 1 27 24
223 240 16
67 1 237 215 84 25 40 149 72 3 152 90 87 112 80 16 241 45 77 36 214
229 59 88
246 188 232 108 71 222 142 50 115 93 105 201 35 6 170 179 47 102 4 150
211 111
120 205 113 4 181 105 127 84 94 239 61 3 121 196 71 148 186 138 184
223 61 220
227 175 107 6 65 81 114 207 245 233 173 228 30 73 206 216 218 133 8 11
98 143
10 179 221 146 97 219 88 67 139 144 39 189 13 123 111 194 156 25 183
51 144 108
145 173 49 229 233 137 86 9 114 64 90 161 63 240 195 111 89 58 97 40
44 227 70
14 78 170 74 197 169 114 138 41 73 242 110 234 177 ^
491 0 145 135 155 207 20 243 208 200 40 48 21 170 5 40 111 178 228 189
196 210

49 6 246 171 56 23 85 128 46 198 217 95 108 241 135 247 130 116 151
130 24 228
34 116 56 207 178 70 112 42 118 221 80 184 157 151 159 37 19 5 132 31
24 136
111 20 148 35 104 132 126 194 232 19 192 225 223 90 230 12 146 120 228
126 39
152 115 100 147 50 28 216 107 35 75 119 85 220 216 44 125 17 185 179
121 95 16
71 39 89 208 42 210 144 176 92 79 228 97 23 114 141 84 172 233 70 175
199 176
155 236 36 31 179 213 15 112 226 211 163 146 28 58 30 131 230 41 162
161 202
142 14 219 73 141 55 156 195 96 159 224 109 197 198 125 172 12 83 213
173 192
130 53 241 72 90 72 182 242 28 150 200 227 138 213 236 99 188 237 150
171 211
100 239 220 124 124 29 109 13 105 183 109 168 182 87 201 246 108 149
215 151
222 194 74 189 38 133 145 164 168 248 244 9 183 14 248 86 179 29 172
170 163
117 15 20 56 115 8 224 204 120 44 33 94 89 21 199 18 78 85 191 203 6
154 1 210
115 163 54 108 212 122 209 156 103 176 63 66 187 123 10 139 77 126 131
75 211 8
69 124 216 202 205 171 27 7 47 154 30 223 153 137 70 83 29 80 67 209
93 193 118
129 160 88 2 229 82 99 33 160 246 79 133 174 177 206 219 10 231 247
222 39 247
38 196 55 222 102 154 93 130 211 71 226 182 150 108 54 42 55 64 25 1
52 35 9
212 30 36 60 195 4 239 128 201 28 121 187 186 167 197 11 32 230 196
239 65 44
220 187 31 87 43 169 204 57 137 225 160 180 172 78 240 21 107 98 232
177 22 129
68 168 153 74 160 25 92 174 148 193 181 36 218 22 158 81 206 127 87
135 89 50
212 59 52 98 166 20 214 192 159 66 32 46 106 47 249 200 69 61 59 144
21 15 175
136 19 100 52 127 182 58 185 208 134 133 240 84 14 156 166 83 121 96
222 131
167 134 104 224 46 40 140 191 16 163 209 185 116 214 37 82 162 190 215
180 235
235 48 205 13 197 103 205 76 104 161 139 232 146 ^
516 0 33 165 33 175 122 134 112 66 1 24 59 67 231 219 5 101 14 227 158
89 41 82
71 61 224 13 23 102 213 3 186 143 223 71 56 174 165 28 136 22 139 157
116 137
247 43 233 155 217 64 91 203 155 227 153 50 193 87 116 122 108 236 7
16 137 242
245 133 160 251 125 238 163 139 141 77 37 77 42 50 177 37 5 127 153 45
143 228

121 156 169 120 219 241 51 166 232 192 150 188 53 55 227 148 82 92 150
172 3
196 232 182 61 229 63 123 97 179 194 39 172 250 157 132 187 95 38 199
94 70 93
118 37 231 240 94 60 4 159 47 58 110 125 243 213 239 136 43 228 105
106 69 125
12 50 198 126 216 117 74 175 112 45 164 76 69 146 101 9 188 58 94 75
143 218 32
197 179 199 89 31 245 123 71 40 135 54 33 170 83 34 117 100 62 171 30
51 145
143 244 160 4 164 86 158 105 91 182 66 27 163 118 52 118 47 79 199 84
233 134
128 105 101 19 25 17 84 7 21 187 90 240 99 99 104 154 2 249 215 154 18
47 63
149 225 238 179 178 72 3 191 182 82 66 11 113 18 57 152 108 215 159 57
147 58
115 164 91 208 205 86 148 16 132 192 145 138 194 60 9 198 147 53 65 64
96 242
14 222 119 241 46 116 132 197 184 238 189 26 200 62 176 80 151 140 109
183 24
44 191 172 234 107 21 192 140 99 94 65 54 7 42 24 49 230 20 229 73 212
45 169
115 180 141 56 200 41 87 121 161 219 231 214 207 1 19 20 219 236 12 55
237 237
211 72 13 28 141 72 239 152 82 87 100 72 8 241 41 77 32 202 229 47 84
238 184
228 100 67 210 130 46 111 93 97 189 31 250 162 171 35 90 4 138 199 99
120 193
113 244 181 93 123 76 90 235 49 251 109 188 63 144 186 138 184 223 61
208 215
175 107 6 53 81 114 207 237 221 161 224 18 73 202 208 210 125 248 11
98 131 250
167 221 146 85 211 88 63 139 136 27 189 5 111 103 186 148 21 171 51
140 104 133
165 49 225 233 129 78 5 106 64 86 153 55 232 187 111 77 58 89 36 32
223 66 6 74
170 62 197 169 110 130 41 61 234 102 234 92 145 133 155 201 14 243 208
200 36
48 21 166 1 40 109 178 147 ^
522 0 179 196 200 34 251 241 156 51 23 75 118 41 193 202 90 93 236 125
242 125
101 151 130 14 218 34 106 41 197 173 55 112 37 108 211 65 174 142 136
159 22 19
245 127 16 19 131 101 10 133 20 104 122 126 184 227 9 187 225 208 90
230 7 131
105 213 121 39 137 105 95 147 45 23 216 92 35 75 114 70 205 206 34 110
12 185
169 111 95 6 71 29 79 203 32 195 144 161 82 64 228 92 8 104 126 74 172
223 65
170 194 171 140 221 36 21 164 198 5 102 211 206 148 136 18 48 25 131
215 36 162

161 192 142 14 214 63 141 45 141 190 86 149 219 94 182 198 115 157 252
73 203
173 192 115 43 226 67 85 62 172 227 13 135 190 227 123 213 231 94 178
227 150
161 196 90 234 210 119 124 14 104 8 100 178 109 153 167 82 201 231 98
149 215
146 217 179 59 174 28 133 130 159 163 248 239 9 183 254 243 71 174 24
162 165
163 107 10 15 46 110 8 214 204 115 39 23 79 89 6 189 3 73 85 176 198 1
154 246
210 115 153 49 108 207 117 209 146 103 176 53 56 172 113 10 129 72 116
126 70
201 3 69 114 211 202 200 171 22 247 42 139 20 218 148 132 70 83 29 75
67 199 88
178 113 124 155 78 2 214 67 89 33 160 246 69 118 159 167 196 204 5 216
237 212
34 247 38 191 55 222 92 149 78 120 211 61 226 177 140 103 39 27 50 54
25 245
241 42 25 249 212 30 21 50 195 254 239 123 186 28 106 182 171 167 192
1 17 220
181 229 65 29 220 177 26 77 33 164 194 57 137 215 150 165 167 78 230
11 102 93
232 177 22 129 68 168 138 74 145 20 77 174 138 188 171 26 218 7 158 76
201 117
72 120 74 35 197 59 52 83 151 20 214 177 144 51 27 46 106 37 249 185
59 56 54
139 16 15 160 136 14 100 37 112 182 53 170 198 119 128 240 69 4 156
161 78 116
240 96 207 121 157 134 99 209 36 25 135 186 1 158 194 175 111 209 32
77 147 190
200 175 225 235 38 195 3 187 88 205 61 89 151 134 217 206 30 162 24
166 113 128
103 66 253 15 50 58 228 210 5 98 8 218 152 80 35 79 62 58 224 7 17 102
207 3
183 140 217 65 56 168 165 22 133 16 133 157 107 128 244 12 ^
517 1 218 150 217 59 81 203 140 217 148 45 193 82 116 122 93 236 7 11
127 242
235 123 155 246 125 238 148 134 126 72 27 247 67 27 50 162 22 247 127
153 45
133 213 121 151 159 105 209 231 36 166 217 177 140 188 53 50 222 148
242 67 87
145 157 250 181 227 177 51 224 63 108 97 174 194 24 162 250 152 132
182 80 33
189 84 60 83 118 37 216 235 94 60 251 154 47 48 110 120 238 208 229
121 43 218
90 91 64 110 12 40 193 111 211 117 64 170 107 35 164 76 54 141 101 4
183 53 79
60 143 213 32 187 169 194 84 31 245 123 66 25 135 39 23 160 78 29 102
95 62 166
20 46 135 128 244 150 246 154 76 143 105 91 172 51 27 158 103 37 108
37 79 199

84 228 119 128 95 91 4 15 17 69 2 11 187 80 230 94 89 89 149 244 239
210 144 18
42 48 139 215 233 179 163 252 67 255 181 167 82 56 253 98 8 42 137 103
205 144
47 137 43 110 149 76 203 200 86 143 16 127 182 140 128 184 55 251 183
142 43 50
54 81 232 9 212 119 236 36 106 122 182 169 223 179 26 185 57 166 80
141 130 99
178 24 44 191 167 219 92 6 187 140 99 89 60 54 249 42 19 44 220 5 214
63 197 30
164 105 180 136 41 195 26 77 116 151 219 231 204 202 1 9 15 214 231 7
40 222
237 206 57 255 13 131 72 224 152 72 87 85 62 255 241 36 77 27 187 229
32 79 228
179 223 90 62 195 115 41 106 93 87 174 26 245 152 161 20 75 4 123 184
84 120
178 113 234 181 78 118 66 85 230 34 251 94 178 53 139 186 138 184 223
61 193
200 175 107 6 38 81 114 207 227 206 146 219 3 73 197 198 200 115 238
11 98 116
240 152 221 146 70 201 88 58 139 126 12 189 252 96 93 176 138 16 156
51 135 99
118 155 49 220 233 119 68 96 64 81 143 45 222 177 111 62 58 79 31 17
218 61 253
69 170 47 197 169 105 120 41 46 224 92 234 92 145 128 155 186 256 243
208 200
26 48 21 156 248 40 104 178 221 175 196 196 28 249 239 150 49 23 71
114 39 191
196 88 87 234 121 240 123 95 151 130 10 214 34 102 35 193 171 49 112
35 104 207
59 170 136 130 159 16 19 241 125 213 ^
529 1 15 127 93 2 121 8 104 114 126 176 223 1 183 225 196 90 230 3 119
93 201
117 39 125 97 91 147 41 19 216 80 35 75 110 58 193 198 26 98 8 185 161
103 95
257 71 21 71 199 24 183 144 149 74 52 228 88 255 96 114 66 172 215 61
166 190
167 128 209 36 13 152 186 256 94 199 202 136 128 10 40 21 131 203 32
162 161
184 142 14 210 55 141 37 129 186 78 141 215 82 170 198 107 145 244 65
195 173
192 103 35 214 63 81 54 164 215 1 123 182 227 111 213 227 90 170 219
150 153
184 82 230 202 115 124 2 100 4 96 174 109 141 155 78 201 219 90 149
215 142 213
167 47 162 20 133 118 155 159 248 235 9 183 246 239 59 170 20 154 161
163 99 6
11 38 106 8 206 204 111 35 15 67 89 253 181 250 69 85 164 194 256 154
242 210
115 145 45 108 203 113 209 138 103 176 45 48 160 105 10 121 68 108 122
66 193

258 69 106 207 202 196 171 18 239 38 127 12 214 144 128 70 83 29 71 67
191 84
166 109 120 151 70 2 202 55 81 33 160 246 61 106 147 159 188 192 1 204
229 204
30 247 38 187 55 222 84 145 66 112 211 53 226 173 132 99 27 15 46 46
25 241 233
34 17 241 212 30 9 42 195 254 239 119 174 28 94 178 159 167 188 252 5
212 169
221 65 17 220 169 22 69 25 160 186 57 137 207 142 153 163 78 222 3 98
89 232
177 22 129 68 168 126 74 133 16 65 174 130 184 163 18 218 254 158 72
197 109 60
108 62 23 185 59 52 71 139 20 214 165 132 39 23 46 106 29 249 173 51
52 50 135
12 15 148 136 10 100 25 100 182 49 158 190 107 124 240 57 255 156 157
74 112
232 96 195 113 149 134 95 197 28 13 131 182 248 154 182 167 107 205 28
73 135
190 188 171 217 235 30 187 254 179 76 205 49 77 143 130 205 206 26 158
12 154
101 120 91 66 253 3 38 46 224 198 5 94 206 144 68 27 75 50 54 224 258
9 102 199
3 179 136 209 57 56 160 165 14 129 8 125 157 95 116 240 36 212 148 217
57 77
203 134 213 146 43 193 80 116 122 87 236 7 9 123 242 231 119 153 244
125 238
142 132 120 70 23 245 63 21 50 156 16 243 127 153 45 129 207 121 149
155 99 205
227 30 166 211 57 ^
548 0 134 188 53 47 219 148 236 58 84 142 148 247 172 224 174 45 221
63 99 97
171 194 15 156 250 149 132 179 71 30 183 78 54 77 118 37 207 232 94 60
248 151
47 42 110 117 235 205 223 112 43 212 81 82 61 101 12 34 190 102 208
117 58 167
104 29 164 76 45 138 101 1 180 50 70 51 143 210 32 181 163 191 81 31
245 123 63
16 135 30 17 154 75 26 93 92 62 163 14 43 129 119 244 144 240 148 70
134 105 91
166 42 27 155 94 28 102 31 79 199 84 225 110 128 89 85 255 9 17 60 259
5 187 74
224 91 83 80 146 238 233 207 138 18 39 39 133 209 230 179 154 252 64
255 175
158 82 50 247 89 2 33 128 100 199 135 41 131 34 107 140 67 200 197 86
140 16
124 176 137 122 178 52 245 174 139 37 41 48 72 226 6 206 119 233 30
100 116 173
160 214 173 26 176 54 160 80 135 124 93 175 24 44 191 164 210 83 257
184 140 99
86 57 54 243 42 16 41 214 256 205 57 188 21 161 99 180 133 32 192 17
71 113 145

219 231 198 199 1 3 12 211 228 4 31 213 237 203 48 249 4 125 72 215
152 66 87
76 56 252 241 33 77 24 178 229 23 76 222 176 220 84 59 186 106 38 103
93 81 165
23 242 146 155 11 66 4 114 175 75 120 169 113 228 181 69 115 60 82 227
25 251
85 172 47 136 186 138 184 223 61 184 191 175 107 6 29 81 114 207 221
197 137
216 254 73 194 192 194 109 232 11 98 107 234 143 221 146 61 195 88 55
139 120 3
189 249 87 87 170 132 13 147 51 132 96 109 149 49 217 233 113 62 257
90 64 78
137 39 216 171 111 53 58 73 28 8 215 58 250 66 170 38 197 169 102 114
41 37 218
86 234 92 145 125 155 177 250 243 208 200 20 48 21 150 245 40 101 178
218 169
196 190 19 246 236 141 46 23 65 108 36 188 187 85 78 231 115 237 120
86 151 130
4 208 34 96 26 187 168 40 112 32 98 201 50 164 127 121 159 7 19 235
122 1 14
126 91 118 5 104 112 126 174 222 259 182 225 193 90 230 2 116 90 198
116 39 122
95 90 147 40 18 216 77 35 75 109 55 190 196 24 95 7 185 159 101 95 256
71 19 69
198 22 180 144 146 72 49 228 87 253 94 111 64 172 213 60 165 189 166
125 206 36
11 149 183 255 92 196 201 133 126 8 38 20 131 213 ^
547 0 29 162 161 178 142 14 207 49 141 31 120 183 72 135 212 73 161
198 101 136
238 59 189 173 192 94 29 205 60 78 48 158 206 254 114 176 227 102 213
224 87
164 213 150 147 175 76 227 196 112 124 255 97 1 93 171 109 132 146 75
201 210
84 149 215 139 210 158 38 153 14 133 109 152 156 248 232 9 183 240 236
50 167
17 148 158 163 93 3 8 32 103 8 200 204 108 32 9 58 89 247 175 244 66
85 155 191
256 154 239 210 115 139 42 108 200 110 209 132 103 176 39 42 151 99 10
115 65
102 119 63 187 258 69 100 204 202 193 171 15 233 35 118 6 211 141 125
70 83 29
68 67 185 81 157 106 117 148 64 2 193 46 75 33 160 246 55 97 138 153
182 183
260 195 223 198 27 247 38 184 55 222 78 142 57 106 211 47 226 170 126
96 18 6
43 40 25 238 227 28 11 235 212 30 36 195 254 239 116 165 28 85 175 150
167 185
249 258 206 160 215 65 8 220 163 19 63 19 157 180 57 137 201 136 144
160 78 216
259 95 86 232 177 22 129 68 168 117 74 124 13 56 174 124 181 157 12
218 248 158

69 194 103 51 99 53 14 176 59 52 62 130 20 214 156 123 30 20 46 106 23
249 164
45 49 47 132 9 15 139 136 7 100 16 91 182 46 149 184 98 121 240 48 252
156 154
71 109 226 96 186 107 143 134 92 188 22 4 128 179 242 151 173 161 104
202 25 70
126 190 179 168 211 235 24 181 251 173 67 205 40 68 137 127 196 206 23
155 3
145 92 114 82 66 253 256 29 37 221 189 5 91 256 197 138 59 21 72 41 51
224 255
3 102 193 3 176 133 203 51 56 154 165 8 126 2 119 157 86 107 237 33
203 145 217
54 71 203 125 207 143 40 193 77 116 122 78 236 7 6 117 242 225 113 150
241 125
238 133 129 111 67 17 242 57 12 50 147 7 237 127 153 45 123 198 121
146 149 90
199 221 21 166 202 162 130 188 53 45 217 148 232 52 82 140 142 245 166
222 172
41 219 63 93 97 169 194 9 152 250 147 132 177 65 28 179 74 50 73 118
37 201 230
94 60 246 149 47 38 110 115 233 203 219 106 43 208 75 76 59 95 12 30
188 96 206
117 54 165 102 25 164 76 39 136 101 261 178 48 64 45 143 208 32 177
159 189 79
31 245 123 61 10 135 24 13 150 73 24 87 90 62 161 10 41 125 209 ^
542 1 244 136 232 140 62 122 105 91 158 30 27 151 82 16 94 23 79 199
84 221 98
128 81 77 247 1 17 48 259 261 187 66 216 87 75 68 142 230 225 203 130
18 35 27
125 201 226 179 142 252 60 255 167 146 82 42 239 77 258 21 116 96 191
123 33
123 22 103 128 55 196 193 86 136 16 120 168 133 114 170 48 237 162 135
29 29 40
60 218 2 198 119 229 22 92 108 161 148 202 165 26 164 50 152 80 127
116 85 171
24 44 191 160 198 71 249 180 140 99 82 53 54 235 42 12 37 206 248 193
49 176 9
157 91 180 129 20 188 5 63 109 137 219 231 190 195 1 259 8 207 224 19
201 237
199 36 241 256 117 72 203 152 58 87 64 48 248 241 29 77 20 166 229 11
72 214
172 216 76 55 174 94 34 99 93 73 153 19 238 138 147 263 54 4 102 163
63 120 157
113 220 181 57 111 52 78 223 13 251 73 164 39 132 186 138 184 223 61
172 179
175 107 6 17 81 114 207 213 185 125 212 246 73 190 184 186 101 224 11
98 95 226
131 221 146 49 187 88 51 139 112 255 189 245 75 79 162 124 9 135 51
128 92 97
141 49 213 233 105 54 257 82 64 74 129 31 208 163 111 41 58 65 24 260
211 54

246 62 170 26 197 169 98 106 41 25 210 78 234 92 145 121 155 165 242
243 208
200 12 48 21 142 241 40 97 178 214 161 196 182 7 242 232 129 42 23 57
100 32
184 175 81 66 227 107 233 116 74 151 130 260 200 34 88 14 179 164 28
112 28 90
193 38 156 115 109 159 259 19 227 118 253 10 122 83 256 106 257 104
104 126 166
218 255 178 225 181 90 230 262 104 78 186 112 39 110 87 86 147 36 14
216 65 35
75 105 43 178 188 16 83 3 185 151 93 95 252 71 11 61 194 14 168 144
134 64 37
228 83 245 86 99 56 172 205 56 161 185 162 113 194 36 3 137 171 251 84
184 197
121 118 30 16 131 188 27 162 161 174 142 14 205 45 141 27 114 181 68
131 210 67
155 198 97 130 234 55 185 173 192 88 25 199 58 76 44 154 200 250 108
172 227 96
213 222 85 160 209 150 143 169 72 225 192 110 124 251 95 263 91 169
109 126 140
73 201 204 80 149 215 137 208 152 32 147 10 133 103 150 154 248 230 9
183 236
234 44 165 15 144 156 163 89 1 6 28 101 8 196 204 106 164 ^
567 0 1 46 89 239 167 236 62 85 143 187 256 154 235 210 115 131 38 108
196 106
209 124 103 176 31 34 139 91 10 107 61 94 115 59 179 258 69 92 200 202
189 171
11 225 31 106 264 207 137 121 70 83 29 64 67 177 77 145 102 113 144 56
2 181 34
67 33 160 246 47 85 126 145 174 171 260 183 215 190 23 247 38 180 55
222 70 138
45 98 211 39 226 166 118 92 6 260 39 32 25 234 219 20 3 227 212 30 254
28 195
254 239 112 153 28 73 171 138 167 181 245 250 198 148 207 65 262 220
155 15 55
11 153 172 57 137 193 128 132 156 78 208 255 91 82 232 177 22 129 68
168 105 74
112 9 44 174 116 177 149 4 218 240 158 65 190 95 39 87 41 2 164 59 52
50 118 20
214 144 111 18 16 46 106 15 249 152 37 45 43 128 5 15 127 136 3 100 4
79 182 42
137 176 86 117 240 36 248 156 150 67 105 218 96 174 99 135 134 88 176
14 258
124 175 234 147 161 153 100 198 21 66 114 190 167 164 203 235 16 173
247 165 55
205 28 56 129 123 184 206 19 151 257 133 80 106 70 66 253 248 17 25
217 177 5
87 252 185 130 47 13 68 29 47 224 251 261 102 185 3 172 129 195 43 56
146 165
122 260 111 157 74 95 233 29 191 141 217 50 63 203 113 199 139 36 193
73 116

122 66 236 7 2 109 242 217 105 146 237 125 238 121 125 99 63 9 238 49
50 135
261 229 127 153 45 115 186 121 142 141 78 191 213 9 166 190 150 122
188 53 41
213 148 224 40 78 136 130 241 154 218 168 33 215 63 81 97 165 194 263
144 250
143 132 173 53 24 171 66 42 65 118 37 189 226 94 60 242 145 47 30 110
111 229
199 211 94 43 200 63 64 55 83 12 22 184 84 202 117 46 161 98 17 164 76
27 132
101 261 174 44 52 33 143 204 32 169 151 185 75 31 245 123 57 264 135
12 5 142
69 20 75 86 62 157 2 37 117 101 244 132 228 136 58 116 105 91 154 24
27 149 76
10 90 19 79 199 84 219 92 128 77 73 243 263 17 42 259 259 187 62 212
85 71 62
140 226 221 201 126 18 33 21 121 197 224 179 136 252 58 255 163 140 82
38 235
71 256 15 110 94 187 117 29 119 16 101 122 49 194 191 86 134 16 118
164 131 110
166 46 233 156 133 25 23 36 54 214 194 119 227 18 88 104 155 142 196
161 26 158
48 148 80 123 112 81 169 24 44 191 158 192 65 245 178 140 223 ^
551 1 78 49 54 227 42 8 33 198 240 181 41 164 265 153 83 180 125 8 184
261 55
105 129 219 231 182 191 1 255 4 203 220 264 7 189 237 195 24 233 248
109 72 191
152 50 87 52 40 244 241 25 77 16 154 229 267 68 206 168 212 68 51 162
82 30 95
93 65 141 15 234 130 139 255 42 4 90 151 51 120 145 113 212 181 45 107
44 74
219 1 251 61 156 31 128 186 138 184 223 61 160 167 175 107 6 5 81 114
207 205
173 113 208 238 73 186 176 178 93 216 11 98 83 218 119 221 146 37 179
88 47 139
104 247 189 241 63 71 154 116 5 123 51 124 88 85 133 49 209 233 97 46
257 74 64
70 121 23 200 155 111 29 58 57 20 252 207 50 242 58 170 14 197 169 94
98 41 13
202 70 234 92 145 117 155 153 234 243 208 200 4 48 21 134 237 40 93
178 210 153
196 174 263 238 228 117 38 23 49 92 28 180 163 77 54 223 99 229 112 62
151 130
256 192 34 80 2 171 160 16 112 24 82 185 26 148 103 97 159 251 19 219
114 245 6
118 75 252 94 249 104 96 126 158 214 251 174 225 169 90 230 262 92 66
174 108
39 98 79 82 147 32 10 216 53 35 75 101 31 166 180 8 71 267 185 143 85
95 248 71
3 53 190 6 156 144 122 56 25 228 79 237 78 87 48 172 197 52 157 181
158 101 182

36 263 125 159 247 76 172 193 109 110 260 22 12 131 176 23 162 161 166
142 14
201 37 141 19 102 177 60 123 206 55 143 198 89 118 226 47 177 173 192
76 17 187
54 72 36 146 188 242 96 164 227 84 213 218 81 152 201 150 135 157 64
221 184
106 124 243 91 263 87 165 109 114 128 69 201 192 72 149 215 133 204
140 20 135
2 133 91 146 150 248 226 9 183 228 230 32 161 11 136 152 163 81 265 2
20 97 8
188 204 102 26 265 40 89 235 163 232 60 85 137 185 256 154 233 210 115
127 36
108 194 104 209 120 103 176 27 30 133 87 10 103 59 90 113 57 175 258
69 88 198
202 187 171 9 221 29 100 262 205 135 119 70 83 29 62 67 173 75 139 100
111 142
52 2 175 28 63 33 160 246 43 79 120 141 170 165 260 177 211 186 21 247
38 178
55 222 66 136 39 94 211 35 226 164 114 90 256 37 28 25 232 215 16 267
223 212
30 250 24 195 254 239 110 147 28 67 169 132 167 179 243 246 194 142
203 65 263 ^
578 0 220 147 11 47 3 149 164 57 137 185 120 120 152 78 200 251 87 78
232 177
22 129 68 168 93 74 100 5 32 174 108 173 141 266 218 232 158 61 186 87
27 75 29
260 152 59 52 38 106 20 214 132 99 6 12 46 106 7 249 140 29 41 39 124
1 15 115
136 269 100 262 67 182 38 125 168 74 113 240 24 244 156 146 63 101 210
96 162
91 127 134 84 164 6 250 120 171 226 143 149 145 96 194 17 62 102 190
155 160
195 235 8 165 243 157 43 205 16 44 121 119 172 206 15 147 249 121 68
98 58 66
253 240 5 13 213 165 5 83 248 173 122 35 5 64 17 43 224 247 257 102
177 3 168
125 187 35 56 138 165 262 118 256 103 157 62 83 229 25 179 137 217 46
55 203
101 191 135 32 193 69 116 122 54 236 7 268 101 242 209 97 142 233 125
238 109
121 87 59 1 234 41 258 50 123 253 221 127 153 45 107 174 121 138 133
66 183 205
267 166 178 138 114 188 53 37 209 148 216 28 74 132 118 237 142 214
164 25 211
63 69 97 161 194 255 136 250 139 132 169 41 20 163 58 34 57 118 37 177
222 94
60 238 141 47 22 110 107 225 195 203 82 43 192 51 52 51 71 12 14 180
72 198 117
38 157 94 9 164 76 15 128 101 261 170 40 40 21 143 200 32 161 143 181
71 31 245
123 53 256 135 267 134 65 16 63 82 62 153 264 33 109 89 244 124 220
128 50 104

105 91 146 12 27 145 64 268 82 11 79 199 84 215 80 128 69 65 235 259
17 30 259
255 187 54 204 81 63 50 136 218 213 197 118 18 29 9 113 189 220 179
124 252 54
255 155 128 82 30 227 59 252 3 98 90 179 105 21 111 4 97 110 37 190
187 86 130
16 114 156 127 102 158 42 225 144 129 17 11 28 42 206 266 186 119 223
10 80 96
143 130 184 153 26 146 44 140 80 115 104 73 165 24 44 191 154 180 53
237 174
140 99 76 47 54 223 42 6 31 194 236 175 37 158 261 151 79 180 123 2
182 257 51
103 125 219 231 178 189 1 253 2 201 218 264 1 183 237 193 18 229 244
105 72 185
152 46 87 46 36 242 241 23 77 14 148 229 263 66 202 166 210 64 49 156
76 28 93
93 61 135 13 232 126 135 251 36 4 84 145 45 120 139 113 208 181 39 105
40 72
217 265 251 55 152 27 126 186 138 184 223 61 154 161 175 107 6 269 81
114 207
201 167 107 206 234 73 184 172 174 89 212 11 98 77 214 113 221 146 31
175 88 45
139 100 243 189 213 ^
578 1 51 63 146 108 1 111 51 120 84 73 125 49 205 233 89 38 257 66 64
66 113 15
192 147 111 17 58 49 16 244 203 46 238 54 170 2 197 169 90 90 41 1 194
62 234
92 145 113 155 141 226 243 208 200 268 48 21 126 233 40 89 178 206 145
196 166
255 234 224 105 34 23 41 84 24 176 151 73 42 219 91 225 108 50 151 130
252 184
34 72 262 163 156 4 112 20 74 177 14 140 91 85 159 243 19 211 110 237
2 114 67
248 82 241 104 88 126 150 210 247 170 225 157 90 230 262 80 54 162 104
39 86 71
78 147 28 6 216 41 35 75 97 19 154 172 59 267 185 135 77 95 244 71 267
45 186
270 144 144 110 48 13 228 75 229 70 75 40 172 189 48 153 177 154 89
170 36 259
113 147 243 68 160 189 97 102 256 14 8 131 164 19 162 161 158 142 14
197 29 141
11 90 173 52 115 202 43 131 198 81 106 218 39 169 173 192 64 9 175 50
68 28 138
176 234 84 156 227 72 213 214 77 144 193 150 127 145 56 217 176 102
124 235 87
263 83 161 109 102 116 65 201 180 64 149 215 129 200 128 8 123 266 133
79 142
146 248 222 9 183 220 226 20 157 7 128 148 163 73 265 270 12 93 8 180
204 98 22
261 28 89 227 155 224 56 85 125 181 256 154 229 210 115 119 32 108 190
100 209

112 103 176 19 22 121 79 10 95 55 82 109 53 167 258 69 80 194 202 183
171 5 213
25 88 258 201 131 115 70 83 29 58 67 165 71 127 96 107 138 44 2 163 16
55 33
160 246 35 67 108 133 162 153 260 165 203 178 17 247 38 174 55 222 58
132 27 86
211 27 226 160 106 86 260 248 33 20 25 228 207 8 263 215 212 30 242 16
195 254
239 106 135 28 55 165 120 167 175 239 238 186 130 195 65 250 220 143 9
43 271
147 160 57 137 181 116 114 150 78 196 249 85 76 232 177 22 129 68 168
87 74 94
3 26 174 104 171 137 264 218 228 158 59 184 83 21 69 23 256 146 59 52
32 100 20
214 126 93 10 46 106 3 249 134 25 39 37 122 271 15 109 136 269 100 258
61 182
36 119 164 68 111 240 18 242 156 144 61 99 206 96 156 87 123 134 82
158 2 246
118 169 222 141 143 141 94 192 15 60 96 190 149 158 191 235 4 161 241
153 37
205 10 38 117 117 166 206 13 145 245 115 62 94 52 66 253 236 271 7 211
159 5 81
246 167 118 29 1 62 11 41 224 245 255 102 173 3 166 123 183 31 56 134
165 260
116 254 81 ^
583 1 157 50 71 225 21 167 133 217 42 47 203 89 183 131 28 193 65 116
122 42
236 7 268 93 242 201 89 138 229 125 238 97 117 75 55 267 230 33 250 50
111 245
213 127 153 45 99 162 121 134 125 54 175 197 259 166 166 126 106 188
53 33 205
148 208 16 70 128 106 233 130 210 160 17 207 63 57 97 157 194 247 128
250 135
132 165 29 16 155 50 26 49 118 37 165 218 94 60 234 137 47 14 110 103
221 191
195 70 43 184 39 40 47 59 12 6 176 60 194 117 30 153 90 1 164 76 3 124
101 261
166 36 28 9 143 196 32 153 135 177 67 31 245 123 49 248 135 262 263
126 61 12
51 78 62 149 260 29 101 77 244 116 212 120 42 92 105 91 138 27 141 52
260 74 3
79 199 84 211 68 128 61 57 227 255 17 18 259 251 187 46 196 77 55 38
132 210
205 193 110 18 25 271 105 181 216 179 112 252 50 255 147 116 82 22 219
47 248
265 86 86 171 93 13 103 266 93 98 25 186 183 86 126 16 110 148 123 94
150 38
217 132 125 9 273 20 30 198 266 178 119 219 2 72 88 131 118 172 145 26
134 40
132 80 107 96 65 161 24 44 191 150 168 41 229 170 140 99 72 43 54 215
42 2 27

186 228 163 29 146 253 147 71 180 119 264 178 249 43 99 117 219 231
170 185 1
249 272 197 214 264 263 171 237 189 6 221 236 97 72 173 152 38 87 34
28 238 241
19 77 10 136 229 255 62 194 162 206 56 45 144 64 24 89 93 53 123 9 228
118 127
243 24 4 72 133 33 120 127 113 200 181 27 101 32 68 213 257 251 43 144
19 122
186 138 184 223 61 142 149 175 107 6 261 81 114 207 193 155 95 202 226
73 180
164 166 81 204 11 98 65 206 101 221 146 19 167 88 41 139 92 235 189
235 45 59
142 104 273 105 51 118 82 67 121 49 203 233 85 34 257 62 64 64 109 11
188 143
111 11 58 45 14 240 201 44 236 52 170 270 197 169 88 86 41 269 190 58
234 92
145 111 155 135 222 243 208 200 266 48 21 122 231 40 87 178 204 141
196 162 251
232 222 99 32 23 37 80 22 174 145 71 36 217 87 223 106 44 151 130 250
180 34 68
258 159 154 272 112 18 70 173 8 136 85 79 159 239 19 207 108 233 112
63 246 76
237 104 84 126 146 208 245 168 225 151 90 230 262 74 48 156 102 39 80
67 76 147
26 4 216 35 35 75 95 13 148 168 270 53 267 185 131 73 95 242 71 265 41
184 268
138 144 104 44 7 228 73 225 66 69 36 272 ^
588 0 181 44 149 173 150 77 158 36 255 101 135 239 60 148 185 85 94
252 6 4 131
152 15 162 161 150 142 14 193 21 141 3 78 169 44 107 198 31 119 198 73
94 210
31 161 173 192 52 1 163 46 64 20 130 164 226 72 148 227 60 213 210 73
136 185
150 119 133 48 213 168 98 124 227 83 263 79 157 109 90 104 61 201 168
56 149
215 125 196 116 272 111 262 133 67 138 142 248 218 9 183 212 222 8 153
3 120
144 163 65 265 270 4 89 8 172 204 94 18 257 16 89 219 147 216 52 85
113 177 256
154 225 210 115 111 28 108 186 96 209 104 103 176 11 14 109 71 10 87
51 74 105
49 159 258 69 72 190 202 179 171 1 205 21 76 254 197 127 111 70 83 29
54 67 157
67 115 92 103 134 36 2 151 4 47 33 160 246 27 55 96 125 154 141 260
153 195 170
13 247 38 170 55 222 50 128 15 78 211 19 226 156 98 82 252 240 29 12
25 224 199
259 207 212 30 234 8 195 254 239 102 123 28 43 161 108 167 171 235 230
178 118
187 65 242 220 135 5 35 267 143 152 57 137 173 108 102 146 78 188 245
81 72 232

177 22 129 68 168 75 74 82 275 14 174 96 167 129 260 218 220 158 55
180 75 9 57
11 248 134 59 52 20 88 20 214 114 81 264 6 46 106 271 249 122 17 35 33
118 271
15 97 136 269 100 250 49 182 32 107 156 56 107 240 6 238 156 140 57 95
198 96
144 79 115 134 78 146 270 238 114 165 214 137 131 133 90 188 11 56 84
190 137
154 183 235 272 153 237 145 25 205 274 26 109 113 154 206 9 141 237
103 50 86
40 66 253 228 263 271 207 147 5 77 242 155 110 17 269 58 275 37 224
241 251 102
165 3 162 119 175 23 56 126 165 256 112 250 91 157 44 65 223 19 161
131 217 40
43 203 83 179 129 26 193 63 116 122 36 236 7 268 89 242 197 85 136 227
125 238
91 115 69 53 265 228 29 246 50 105 241 209 127 153 45 95 156 121 132
121 48 171
193 255 166 160 120 102 188 53 31 203 148 204 10 68 126 100 231 124
208 158 13
205 63 51 97 155 194 243 124 250 133 132 163 23 14 151 46 22 45 118 37
159 216
94 60 232 135 47 10 110 101 219 189 191 64 43 180 33 34 45 53 12 2 174
54 192
117 26 151 88 273 164 76 273 122 101 261 164 34 22 3 143 194 32 149
131 175 65
31 245 123 47 244 135 258 261 122 59 10 45 76 62 147 258 27 97 71 244
112 208
116 38 86 105 91 134 270 27 139 46 256 70 275 79 199 160 ^
594 0 207 56 128 53 49 219 251 17 6 259 247 187 38 188 73 47 26 128
202 197 189
102 18 21 263 97 173 212 179 100 252 46 255 139 104 82 14 211 35 244
257 74 82
163 81 5 95 258 89 86 13 182 179 86 122 16 106 140 119 86 142 34 209
120 121 1
265 12 18 190 266 170 119 215 272 64 80 119 106 160 137 26 122 36 124
80 99 88
57 157 24 44 191 146 156 29 221 166 140 99 68 39 54 207 42 276 23 178
220 151
21 134 245 143 63 180 115 256 174 241 35 95 109 219 231 162 181 1 245
272 193
210 264 255 159 237 185 272 213 228 89 72 161 152 30 87 22 20 234 241
15 77 6
124 229 247 58 186 158 202 48 41 132 52 20 85 93 45 111 5 224 110 119
235 12 4
60 121 21 120 115 113 192 181 15 97 24 64 209 249 251 31 136 11 118
186 138 184
223 61 130 137 175 107 6 253 81 114 207 185 143 83 198 218 73 176 156
158 73
196 11 98 53 198 89 221 146 7 159 88 37 139 84 227 189 231 33 51 134
96 273 93

51 114 78 55 113 49 199 233 77 26 257 54 64 60 101 3 180 135 111 277
58 37 10
232 197 40 232 48 170 262 197 169 84 78 41 261 182 50 234 92 145 107
155 123
214 243 208 200 262 48 21 114 227 40 83 178 200 133 196 154 243 228
218 87 28
23 29 72 18 170 133 67 24 213 79 219 102 32 151 130 246 172 34 60 250
151 150
264 112 14 62 165 274 128 73 67 159 231 19 199 104 225 274 108 55 242
64 229
104 76 126 138 204 241 164 225 139 90 230 262 62 36 144 98 39 68 59 72
147 22
216 23 35 75 91 1 136 160 266 41 267 185 123 65 95 238 71 261 33 180
264 126
144 92 36 273 228 69 217 58 57 28 172 177 42 147 171 148 71 152 36 253
95 129
237 56 142 183 79 90 250 2 2 131 146 13 162 161 146 142 14 191 17 141
277 72
167 40 103 196 25 113 198 69 88 206 27 157 173 192 46 275 157 44 62 16
126 158
222 66 144 227 54 213 208 71 132 181 150 115 127 44 211 164 96 124 223
81 263
77 155 109 84 98 59 201 162 52 149 215 123 194 110 268 105 260 133 61
136 140
248 216 9 183 208 220 2 151 1 116 142 163 61 265 270 87 8 168 204 92
16 255 10
89 215 143 212 50 85 107 175 256 154 223 210 115 107 26 108 184 94 209
100 103
176 7 10 103 67 10 83 49 70 103 47 155 258 69 68 188 202 177 171 277
201 19 70
252 195 125 109 70 83 29 52 67 153 65 109 90 101 132 32 2 145 276 43
33 160 246
23 259 ^
600 1 87 119 148 132 260 144 189 164 10 247 38 167 55 222 44 125 6 72
211 13
226 153 92 79 246 234 26 6 25 221 193 273 256 201 212 30 228 2 195 254
239 99
114 28 34 158 99 167 168 232 224 172 109 181 65 236 220 129 2 29 264
140 146 57
137 167 102 93 143 78 182 242 78 69 232 177 22 129 68 168 66 74 73 275
5 174 90
164 123 257 218 214 158 52 177 69 48 2 242 125 59 52 11 79 20 214 105
72 258 3
46 106 268 249 113 11 32 30 115 271 15 88 136 269 100 244 40 182 29 98
150 47
104 240 276 235 156 137 54 92 192 96 135 73 109 134 75 137 267 232 111
162 208
134 122 127 87 185 8 53 75 190 128 151 177 235 269 147 234 139 16 205
268 17
103 110 145 206 6 138 231 94 41 80 31 66 253 222 257 265 204 138 5 74
239 146

104 8 266 55 269 34 224 238 248 102 159 3 159 116 169 17 56 120 165
253 109 247
85 157 35 56 220 16 152 128 217 37 37 203 74 173 126 23 193 60 116 122
27 236 7
268 83 242 191 79 133 224 125 238 82 112 60 50 262 225 23 240 50 96
235 203 127
153 45 89 147 121 129 115 39 165 187 249 166 151 111 96 188 53 28 200
148 198 1
65 123 91 228 115 205 155 7 202 63 42 97 152 194 237 118 250 130 132
160 14 11
145 40 16 39 118 37 150 213 94 60 229 132 47 4 110 98 216 186 185 55
43 174 24
25 42 44 12 275 171 45 189 117 20 148 85 270 164 76 267 119 101 261
161 31 13
273 143 191 32 143 125 172 62 31 245 123 44 238 135 252 258 116 56 7
36 73 62
144 255 24 91 62 244 106 202 110 32 77 105 91 128 264 27 136 37 250 64
272 79
199 84 206 53 128 51 47 217 250 17 3 259 246 187 36 186 72 45 23 127
200 195
188 100 18 20 261 95 171 211 179 97 252 45 255 137 101 82 12 209 32
243 255 71
81 161 78 3 93 256 88 83 10 181 178 86 121 16 105 138 118 84 140 33
207 117 120
278 263 10 15 188 266 168 119 214 271 62 78 116 103 157 135 26 119 35
122 80 97
86 55 156 24 44 191 145 153 26 219 165 140 99 67 38 54 205 42 276 22
176 218
148 19 131 243 142 61 180 114 254 173 239 33 94 107 219 231 160 180 1
244 272
192 209 264 253 156 237 184 270 211 226 87 72 158 152 28 87 19 18 233
241 14 77
5 121 229 245 57 184 157 201 46 40 129 49 19 84 93 43 108 4 223 108
117 233 9 4
57 118 18 120 112 113 190 181 12 96 22 63 208 247 251 28 134 9 117 186
138 184
223 61 127 216 ^
590 1 175 107 6 247 81 114 207 179 134 74 195 212 73 173 150 152 67
190 11 98
44 192 80 221 146 279 153 88 34 139 78 221 189 228 24 45 128 90 273 84
51 111
75 46 107 49 196 233 71 20 257 48 64 57 95 278 174 129 111 271 58 31 7
226 194
37 229 45 170 256 197 169 81 72 41 255 176 44 234 92 145 104 155 114
208 243
208 200 259 48 21 108 224 40 80 178 197 127 196 148 237 225 215 78 25
23 23 66
15 167 124 64 15 210 73 216 99 23 151 130 243 166 34 54 244 145 147
258 112 11
56 159 268 122 64 58 159 225 19 193 101 219 274 105 49 239 55 223 104
70 126

132 201 238 161 225 130 90 230 262 53 27 135 95 39 59 53 69 147 19 278
216 14
35 75 88 273 127 154 263 32 267 185 117 59 95 235 71 258 27 177 261
117 144 83
30 267 228 66 211 52 48 22 172 171 39 144 168 145 62 143 36 250 86 120
234 50
133 180 70 84 247 277 280 131 137 10 162 161 140 142 14 188 11 141 274
63 164
34 97 193 16 104 198 63 79 200 21 151 173 192 37 272 148 41 59 10 120
149 216
57 138 227 45 213 205 68 126 175 150 109 118 38 208 158 93 124 217 78
263 74
152 109 75 89 56 201 153 46 149 215 120 191 101 262 96 257 133 52 133
137 248
213 9 183 202 217 274 148 279 110 139 163 55 265 270 275 84 8 162 204
89 13 252
1 89 209 137 206 47 85 98 172 256 154 220 210 115 101 23 108 181 91
209 94 103
176 1 4 94 61 10 77 46 64 100 44 149 258 69 62 185 202 174 171 277 195
16 61
249 192 122 106 70 83 29 49 67 147 62 100 87 98 129 26 2 136 270 37 33
160 246
17 40 81 115 144 126 260 138 185 160 8 247 38 165 55 222 40 123 68 211
9 226
151 88 77 242 230 24 2 25 219 189 271 254 197 212 30 224 279 195 254
239 97 108
28 28 156 93 167 166 230 220 168 103 177 65 232 220 125 25 262 138 142
57 137
163 98 87 141 78 178 240 76 67 232 177 22 129 68 168 60 74 67 275 280
174 86
162 119 255 218 210 158 50 175 65 275 42 277 238 119 59 52 5 73 20 214
99 66
254 1 46 106 266 249 107 7 30 28 113 271 15 82 136 269 100 240 34 182
27 92 146
41 102 240 272 233 156 135 52 90 188 96 129 69 105 134 73 131 265 228
109 160
204 132 116 123 85 183 6 51 69 190 122 149 173 235 267 143 232 135 10
205 264
11 99 108 139 206 4 136 227 88 35 76 25 66 253 218 253 261 202 132 5
72 237 140
203 ^
620 0 279 262 51 261 30 224 234 244 102 151 3 155 112 161 9 56 112 165
249 105
243 77 157 23 44 216 12 140 124 217 33 29 203 62 165 122 19 193 56 116
122 15
236 7 268 75 242 183 71 129 220 125 238 70 108 48 46 258 221 15 232 50
84 227
195 127 153 45 81 135 121 125 107 27 157 179 241 166 139 99 88 188 53
24 196
148 190 272 61 119 79 224 103 201 151 282 198 63 30 97 148 194 229 110
250 126

132 156 2 7 137 32 8 31 118 37 138 209 94 60 225 128 47 279 110 94 212
182 177
43 43 166 12 13 38 32 12 271 167 33 185 117 12 144 81 266 164 76 259
115 101
261 157 27 1 265 143 187 32 135 117 168 58 31 245 123 40 230 135 244
254 108 52
3 24 69 62 140 251 20 83 50 244 98 194 102 24 65 105 91 120 256 27 132
25 242
56 268 79 199 84 202 41 128 43 39 209 246 17 274 259 242 187 28 178 68
37 11
123 192 187 184 92 18 16 253 87 163 207 179 85 252 41 255 129 89 82 4
201 20
239 247 59 77 153 66 278 85 248 84 71 281 177 174 86 117 16 101 130
114 76 132
29 199 105 116 274 255 2 3 180 266 160 119 210 267 54 70 104 91 145
127 26 107
31 114 80 89 78 47 152 24 44 191 141 141 14 211 161 140 99 63 34 54
197 42 276
18 168 210 136 11 119 235 138 53 180 110 246 169 231 25 90 99 219 231
152 176 1
240 272 188 205 264 245 144 237 180 262 203 218 79 72 146 152 20 87 7
10 229
241 10 77 1 109 229 237 53 176 153 197 38 36 117 37 15 80 93 35 96 219
100 109
225 280 4 45 106 6 120 100 113 182 181 92 14 59 204 239 251 16 126 1
113 186
138 184 223 61 115 122 175 107 6 243 81 114 207 175 128 68 193 208 73
171 146
148 63 186 11 98 38 188 74 221 146 275 149 88 32 139 74 217 189 226 18
41 124
86 273 78 51 109 73 40 103 49 194 233 67 16 257 44 64 55 91 276 170
125 111 267
58 27 5 222 192 35 227 43 170 252 197 169 79 68 41 251 172 40 234 92
145 102
155 108 204 243 208 200 257 48 21 104 222 40 78 178 195 123 196 144
233 223 213
72 23 23 19 62 13 165 118 62 9 208 69 214 97 17 151 130 241 162 34 50
240 141
145 254 112 9 52 155 264 118 58 52 159 221 19 189 99 215 274 103 45
237 49 219
104 66 126 128 199 236 159 225 124 90 230 262 47 21 129 93 39 53 49 67
147 17
278 216 8 35 75 86 269 121 150 261 26 267 185 113 55 95 233 71 256 23
175 259
111 144 77 26 263 228 64 207 48 42 18 172 167 37 142 166 143 56 137 36
248 80
114 232 46 127 257 ^
605 1 58 76 243 273 280 131 125 6 162 161 132 142 14 184 3 141 270 51
160 26 89
189 4 92 198 55 67 192 13 143 173 192 25 268 136 37 55 2 112 137 208
45 130 227

33 213 201 64 118 167 150 101 106 30 204 150 89 124 209 74 263 70 148
109 63 77
52 201 141 38 149 215 116 187 89 254 84 253 133 40 129 133 248 209 9
183 194
213 266 144 279 102 135 163 47 265 270 271 80 8 154 204 85 9 248 274
89 201 129
198 43 85 86 168 256 154 216 210 115 93 19 108 177 87 209 86 103 176
278 281 82
53 10 69 42 56 96 40 141 258 69 54 181 202 170 171 277 187 12 49 245
188 118
102 70 83 29 45 67 139 58 88 83 94 125 18 2 124 262 29 33 160 246 9 28
69 107
136 114 260 126 177 152 4 247 38 161 55 222 32 119 273 60 211 1 226
147 80 73
234 222 20 279 25 215 181 267 250 189 212 30 216 275 195 254 239 93 96
28 16
152 81 167 162 226 212 160 91 169 65 224 220 117 281 17 258 134 134 57
137 155
90 75 137 78 170 236 72 63 232 177 22 129 68 168 48 74 55 275 272 174
78 158
111 251 218 202 158 46 171 57 267 30 269 230 107 59 52 278 61 20 214
87 54 246
282 46 106 262 249 95 284 26 24 109 271 15 70 136 269 100 232 22 182
23 80 138
29 98 240 264 229 156 131 48 86 180 96 117 61 97 134 69 119 261 220
105 156 196
128 104 115 81 179 2 47 57 190 110 145 165 235 263 135 228 127 283 205
256 284
91 104 127 206 132 219 76 23 68 13 66 253 210 245 253 198 120 5 68 233
128 92
275 260 49 257 28 224 232 242 102 147 3 153 110 157 5 56 108 165 247
103 241 73
157 17 38 214 10 134 122 217 31 25 203 56 161 120 17 193 54 116 122 9
236 7 268
71 242 179 67 127 218 125 238 64 106 42 44 256 219 11 228 50 78 223
191 127 153
45 77 129 121 123 103 21 153 175 237 166 133 93 84 188 53 22 194 148
186 268 59
117 73 222 97 199 149 280 196 63 24 97 146 194 225 106 250 124 132 154
281 5
133 28 4 27 118 37 132 207 94 60 223 126 47 277 110 92 210 180 173 37
43 162 6
7 36 26 12 269 165 27 183 117 8 142 79 264 164 76 255 113 101 261 155
25 280
261 143 185 32 131 113 166 56 31 245 123 38 226 135 240 252 104 50 1
18 67 62
138 249 18 79 44 244 94 190 98 20 59 105 91 116 252 27 130 19 238 52
266 79 199
84 200 35 128 39 35 205 244 17 270 259 240 187 24 174 66 33 5 121 188
183 182
88 18 14 249 83 159 205 179 79 252 186 ^

615 0 255 121 77 82 283 193 8 235 239 47 73 145 54 274 77 240 80 59
273 173 170
86 113 16 97 122 110 68 124 25 191 93 112 270 247 281 278 172 266 152
119 206
263 46 62 92 79 133 119 26 95 27 106 80 81 70 39 148 24 44 191 137 129
2 203
157 140 99 59 30 54 189 42 276 14 160 202 124 3 107 227 134 45 180 106
238 165
223 17 86 91 219 231 144 172 1 236 272 184 201 264 237 132 237 176 254
195 210
71 72 134 152 12 87 282 2 225 241 6 77 284 97 229 229 49 168 149 193
30 32 105
25 11 76 93 27 84 283 215 92 101 217 272 4 33 94 281 120 88 113 174
181 275 88
6 55 200 231 251 4 118 280 109 186 138 184 223 61 103 110 175 107 6
235 81 114
207 167 116 56 189 200 73 167 138 140 55 178 11 98 26 180 62 221 146
267 141 88
28 139 66 209 189 222 6 33 116 78 273 66 51 105 69 28 95 49 190 233 59
8 257 36
64 51 83 272 162 117 111 259 58 19 1 214 188 31 223 39 170 244 197 169
75 60 41
243 164 32 234 92 145 98 155 96 196 243 208 200 253 48 21 96 218 40 74
178 191
115 196 136 225 219 209 60 19 23 11 54 9 161 106 58 284 204 61 210 93
5 151 130
237 154 34 42 232 133 141 246 112 5 44 147 256 110 46 40 159 213 19
181 95 207
274 99 37 233 37 211 104 58 126 120 195 232 155 225 112 90 230 262 35
9 117 89
39 41 41 63 147 13 278 216 283 35 75 82 261 109 142 257 14 267 185 105
47 95
229 71 252 15 171 255 99 144 65 18 255 228 60 199 40 30 10 172 159 33
138 162
139 44 125 36 244 68 102 228 38 115 174 52 72 241 271 280 131 119 4
162 161 128
142 14 182 286 141 268 45 158 22 85 187 285 86 198 51 61 188 9 139 173
192 19
266 130 35 53 285 108 131 204 39 126 227 27 213 199 62 114 163 150 97
100 26
202 146 87 124 205 72 263 68 146 109 57 71 50 201 135 34 149 215 114
185 83 250
78 251 133 34 127 131 248 207 9 183 190 211 262 142 279 98 133 163 43
265 270
269 78 8 150 204 83 7 246 270 89 197 125 194 41 85 80 166 256 154 214
210 115
89 17 108 175 85 209 82 103 176 276 279 76 49 10 65 40 52 94 38 137
258 69 50
179 202 168 171 277 183 10 43 243 186 116 100 70 83 29 43 67 135 56 82
81 92
123 14 2 118 258 25 33 160 246 5 22 63 103 132 108 260 120 173 148 2
247 38 159

55 222 28 117 269 56 211 284 226 145 76 71 230 218 18 277 25 213 177
265 248
185 212 30 212 273 195 254 239 91 90 28 10 150 75 167 160 224 175 ^
613 0 152 79 161 65 216 220 109 281 9 254 130 126 57 137 147 82 63 133
78 162
232 68 59 232 177 22 129 68 168 36 74 43 275 264 174 70 154 103 247
218 194 158
42 167 49 259 18 261 222 95 59 52 270 49 20 214 75 42 238 282 46 106
258 249 83
280 22 20 105 271 15 58 136 269 100 224 10 182 19 68 130 17 94 240 256
225 156
127 44 82 172 96 105 53 89 134 65 107 257 212 101 152 188 124 92 107
77 175 287
43 45 190 98 141 157 235 259 127 224 119 275 205 248 276 83 100 115
206 285 128
211 64 11 60 1 66 253 202 237 245 194 108 5 64 229 116 84 267 256 45
249 24 224
228 238 102 139 3 149 106 149 286 56 100 165 243 99 237 65 157 5 26
210 6 122
118 217 27 17 203 44 153 116 13 193 50 116 122 286 236 7 268 63 242
171 59 123
214 125 238 52 102 30 40 252 215 3 220 50 66 215 183 127 153 45 69 117
121 119
95 9 145 167 229 166 121 81 76 188 53 18 190 148 178 260 55 113 61 218
85 195
145 276 192 63 12 97 142 194 217 98 250 120 132 150 273 1 125 20 285
19 118 37
120 203 94 60 219 122 47 273 110 88 206 176 165 25 43 154 283 284 32
14 12 265
161 15 179 117 138 75 260 164 76 247 109 101 261 151 21 272 253 143
181 32 123
105 162 52 31 245 123 34 218 135 232 248 96 46 286 6 63 62 134 245 14
71 32 244
86 182 90 12 47 105 91 108 244 27 126 7 230 44 262 79 199 84 196 23
128 31 27
197 240 17 262 259 236 187 16 166 62 25 282 117 180 175 178 80 18 10
241 75 151
201 179 67 252 35 255 117 71 82 281 189 2 233 235 41 71 141 48 272 73
236 78 53
269 171 168 86 111 16 95 118 108 64 120 23 187 87 110 268 243 279 274
168 266
148 119 204 261 42 58 86 73 127 115 26 89 25 102 80 77 66 35 146 24 44
191 135
123 285 199 155 140 99 57 28 54 185 42 276 12 156 198 118 288 101 223
132 41
180 104 234 163 219 13 84 87 219 231 140 170 1 234 272 182 199 264 233
126 237
174 250 191 206 67 72 128 152 8 87 278 287 223 241 4 77 284 91 229 225
47 164
147 191 26 30 99 19 9 74 93 23 78 283 213 88 97 213 268 4 27 88 277
120 82 113

170 181 271 86 2 53 198 227 251 287 114 278 107 186 138 184 223 61 97
104 175
107 6 231 81 114 207 163 110 50 187 196 73 165 134 136 51 174 11 98 20
176 56
221 146 263 137 88 26 139 62 205 189 220 29 112 74 273 60 51 103 67 22
91 49
188 233 55 4 257 32 64 49 79 270 158 113 111 255 58 15 288 210 213 ^
624 1 28 220 36 170 238 197 169 72 54 41 237 158 26 234 92 145 95 155
87 190
243 208 200 250 48 21 90 215 40 71 178 188 109 196 130 219 216 206 51
16 23 5
48 6 158 97 55 278 201 55 207 90 286 151 130 234 148 34 36 226 127 138
240 112
2 38 141 250 104 37 31 159 207 19 175 92 201 274 96 31 230 28 205 104
52 126
114 192 229 152 225 103 90 230 262 26 108 86 39 32 35 60 147 10 278
216 277 35
75 79 255 100 136 254 5 267 185 99 41 95 226 71 249 9 168 252 90 144
56 12 249
228 57 193 34 21 4 172 153 30 135 159 136 35 116 36 241 59 93 225 32
106 171 43
66 238 268 280 131 110 1 162 161 122 142 14 179 283 141 265 36 155 16
79 184
279 77 198 45 52 182 3 133 173 192 10 263 121 32 50 282 102 122 198 30
120 227
18 213 196 59 108 157 150 91 91 20 199 140 84 124 199 69 263 65 143
109 48 62
47 201 126 28 149 215 111 182 74 244 69 248 133 25 124 128 248 204 9
183 184
208 256 139 279 92 130 163 37 265 270 266 75 8 144 204 80 4 243 264 89
191 119
188 38 85 71 163 256 154 211 210 115 83 14 108 172 82 209 76 103 176
273 276 67
43 10 59 37 46 91 35 131 258 69 44 176 202 165 171 277 177 7 34 240
183 113 97
70 83 29 40 67 129 53 73 78 89 120 8 2 109 252 19 33 160 246 289 13 54
97 126
99 260 111 167 142 289 247 38 156 55 222 22 114 263 50 211 281 226 142
70 68
224 212 15 274 25 210 171 262 245 179 212 30 206 270 195 254 239 88 81
28 1 147
66 167 157 221 202 150 76 159 65 214 220 107 281 7 253 129 124 57 137
145 80 60
132 78 160 231 67 58 232 177 22 129 68 168 33 74 40 275 262 174 68 153
101 246
218 192 158 41 166 47 257 15 259 220 92 59 52 268 46 20 214 72 39 236
282 46
106 257 249 80 279 21 19 104 271 15 55 136 269 100 222 7 182 18 65 128
14 93
240 254 224 156 126 43 81 170 96 102 51 87 134 64 104 256 210 100 151
186 123

89 105 76 174 287 42 42 190 95 140 155 235 258 125 223 117 273 205 246
274 81
99 112 206 285 127 209 61 8 58 288 66 253 200 235 243 193 105 5 63 228
113 82
265 255 44 247 23 224 227 237 102 137 3 148 105 147 285 56 98 165 242
98 236 63
157 2 23 209 5 119 117 217 26 15 203 41 151 115 12 193 49 116 122 284
236 7 268
61 242 169 57 122 213 125 238 49 101 27 39 251 214 1 218 50 63 213 181
127 153
45 67 114 121 118 93 6 143 165 227 166 118 78 74 188 53 17 189 148 176
258 54
112 58 217 82 194 144 275 191 141 ^
628 1 3 97 139 194 211 92 250 117 132 147 267 290 119 14 282 13 118 37
111 200
94 60 216 119 47 270 110 85 203 173 159 16 43 148 277 278 29 5 12 262
158 6 176
117 286 135 72 257 164 76 241 106 101 261 148 18 266 247 143 178 32
117 99 159
49 31 245 123 31 212 135 226 245 90 43 286 289 60 62 131 242 11 65 23
244 80
176 84 6 38 105 91 102 238 27 123 290 224 38 259 79 199 84 193 14 128
25 21 191
237 17 256 259 233 187 10 160 59 19 276 114 174 169 175 74 18 7 235 69
145 198
179 58 252 32 255 111 62 82 278 183 285 230 229 32 68 135 39 269 67
230 75 44
263 168 165 86 108 16 92 112 105 58 114 20 181 78 107 265 237 276 268
162 266
142 119 201 258 36 52 77 64 118 109 26 80 22 96 80 71 60 29 143 24 44
191 132
114 279 193 152 140 99 54 25 54 179 42 276 9 150 192 109 285 92 217
129 35 180
101 228 160 213 7 81 81 219 231 134 167 1 231 272 179 196 264 227 117
237 171
244 185 200 61 72 119 152 2 87 272 284 220 241 1 77 284 82 229 219 44
158 144
188 20 27 90 10 6 71 93 17 69 283 210 82 91 207 262 4 18 79 271 120 73
113 164
181 265 83 288 50 195 221 251 281 108 275 104 186 138 184 223 61 88 95
175 107
6 225 81 114 207 157 101 41 184 190 73 162 128 130 45 168 11 98 11 170
47 221
146 257 131 88 23 139 56 199 189 217 283 23 106 68 273 51 51 100 64 13
85 49
185 233 49 290 257 26 64 46 73 267 152 107 111 249 58 9 288 204 183 26
218 34
170 234 197 169 70 50 41 233 154 22 234 92 145 93 155 81 186 243 208
200 248 48
21 86 213 40 69 178 186 105 196 126 215 214 204 45 14 23 1 44 4 156 91
53 274

199 51 205 88 282 151 130 232 144 34 32 222 123 136 236 112 34 137 246
100 31
25 159 203 19 171 90 197 274 94 27 228 22 201 104 48 126 110 190 227
150 225 97
90 230 262 20 286 102 84 39 26 31 58 147 8 278 216 273 35 75 77 251 94
132 252
291 267 185 95 37 95 224 71 247 5 166 250 84 144 50 8 245 228 55 189
30 15 172
149 28 133 157 134 29 110 36 239 53 87 223 28 100 169 37 62 236 266
280 131 104
291 162 161 118 142 14 177 281 141 263 30 153 12 75 182 275 71 198 41
46 178
291 129 173 192 4 261 115 30 48 280 98 116 194 24 116 227 12 213 194
57 104 153
150 87 85 16 197 136 82 124 195 67 263 63 141 109 42 56 45 201 120 24
149 215
109 180 68 240 63 246 133 19 122 126 248 202 9 183 180 206 252 137 279
88 128
163 33 265 270 264 73 8 140 204 78 2 241 76 ^
622 0 89 183 111 180 34 85 59 159 256 154 207 210 115 75 10 108 168 78
209 68
103 176 269 272 55 35 10 51 33 38 87 31 123 258 69 36 172 202 161 171
277 169 3
22 236 179 109 93 70 83 29 36 67 121 49 61 74 85 116 2 97 244 11 33
160 246 285
1 42 89 118 87 260 99 159 134 289 247 38 152 55 222 14 110 255 42 211
277 226
138 62 64 216 204 11 270 25 206 163 258 241 171 212 30 198 266 195 254
239 84
69 28 283 143 54 167 153 217 194 142 64 151 65 206 220 99 281 293 249
125 116
57 137 137 72 48 128 78 152 227 63 54 232 177 22 129 68 168 21 74 28
275 254
174 60 149 93 242 218 184 158 37 162 39 249 3 251 212 80 59 52 260 34
20 214 60
27 228 282 46 106 253 249 68 275 17 15 100 271 15 43 136 269 100 214
289 182 14
53 120 2 89 240 246 220 156 122 39 77 162 96 90 43 79 134 60 92 252
202 96 147
178 119 77 97 72 170 287 38 30 190 83 136 147 235 254 117 219 109 265
205 238
266 73 95 100 206 285 123 201 49 290 50 280 66 253 192 227 235 189 93
5 59 224
101 74 257 251 40 239 19 224 223 233 102 129 3 144 101 139 281 56 90
165 238 94
232 55 157 284 11 205 1 107 113 217 22 7 203 29 143 111 8 193 45 116
122 276
236 7 268 53 242 161 49 118 209 125 238 37 97 15 35 247 210 287 210 50
51 205
173 127 153 45 59 102 121 114 85 288 135 157 219 166 106 66 66 188 53
13 185

148 168 250 50 108 46 213 70 190 140 271 187 63 291 97 137 194 207 88
250 115
132 145 263 290 115 10 280 9 118 37 105 198 94 60 214 117 47 268 110
83 201 171
155 10 43 144 273 274 27 293 12 260 156 174 117 284 133 70 255 164 76
237 104
101 261 146 16 262 243 143 176 32 113 95 157 47 31 245 123 29 208 135
222 243
86 41 286 285 58 62 129 240 9 61 17 244 76 172 80 2 32 105 91 98 234
27 121 286
220 34 257 79 199 84 191 8 128 21 17 187 235 17 252 259 231 187 6 156
57 15 272
112 170 165 173 70 18 5 231 65 141 196 179 52 252 30 255 107 56 82 276
179 281
228 225 26 66 131 33 267 63 226 73 38 259 166 163 86 106 16 90 108 103
54 110
18 177 72 105 263 233 274 264 158 266 138 119 199 256 32 48 71 58 112
105 26 74
20 92 80 67 56 25 141 24 44 191 130 108 275 189 150 140 99 52 23 54
175 42 276
7 146 188 103 283 86 213 127 31 180 99 224 158 209 3 79 77 219 231 130
165 1
229 272 177 194 264 223 111 237 169 240 181 196 57 72 113 152 292 87
268 282
218 241 293 77 284 76 229 80 ^
635 1 40 150 140 184 12 23 78 294 2 67 93 9 57 283 206 74 83 199 254 4
6 67 263
120 61 113 156 181 257 79 284 46 191 213 251 273 100 271 100 186 138
184 223 61
76 83 175 107 6 217 81 114 207 149 89 29 180 182 73 158 120 122 37 160
11 98
295 162 35 221 146 249 123 88 19 139 48 191 189 213 275 15 98 60 273
39 51 96
60 1 77 49 181 233 41 286 257 18 64 42 65 263 144 99 111 241 58 1 288
196 179
22 214 30 170 226 197 169 66 42 41 225 146 14 234 92 145 89 155 69 178
243 208
200 244 48 21 78 209 40 65 178 182 97 196 118 207 210 200 33 10 23 289
36 152
79 49 266 195 43 201 84 274 151 130 228 136 34 24 214 115 132 228 112
292 26
129 238 92 19 13 159 195 19 163 86 189 274 90 19 224 10 193 104 40 126
102 186
223 146 225 85 90 230 262 8 278 90 80 39 14 23 54 147 4 278 216 265 35
75 73
243 82 124 248 283 267 185 87 29 95 220 71 243 293 162 246 72 144 38
237 228 51
181 22 3 288 172 141 24 129 153 130 17 98 36 235 41 75 219 20 88 165
25 54 232
262 280 131 92 291 162 161 110 142 14 173 277 141 259 18 149 4 67 178
267 59

198 33 34 170 287 121 173 192 288 257 103 26 44 276 90 104 186 12 108
227 213
190 53 96 145 150 79 73 8 193 128 78 124 187 63 263 59 137 109 30 44
41 201 108
16 149 215 105 176 56 232 51 242 133 7 118 122 248 198 9 183 172 202
244 133
279 80 124 163 25 265 270 260 69 8 132 204 74 294 237 252 89 179 107
176 32 85
53 157 256 154 205 210 115 71 8 108 166 76 209 64 103 176 267 270 49
31 10 47
31 34 85 29 119 258 69 32 170 202 159 171 277 165 1 16 234 177 107 91
70 83 29
34 67 117 47 55 72 83 114 292 2 91 240 7 33 160 246 283 291 36 85 114
81 260 93
155 130 289 247 38 150 55 222 10 108 251 38 211 275 226 136 58 62 212
200 9 268
25 204 159 256 239 167 212 30 194 264 195 254 239 82 63 28 279 141 48
167 151
215 190 138 58 147 65 202 220 95 281 291 247 123 112 57 137 133 68 42
126 78
148 225 61 52 232 177 22 129 68 168 15 74 22 275 250 174 56 147 89 240
218 180
158 35 160 35 245 293 247 208 74 59 52 256 28 20 214 54 21 224 282 46
106 251
249 62 273 15 13 98 271 15 37 136 269 100 210 285 182 12 47 116 292 87
240 242
218 156 120 37 75 158 96 84 39 75 134 58 86 250 198 94 145 174 117 71
93 70 168
287 36 24 190 77 134 143 235 252 113 217 105 261 205 234 262 69 93 94
206 285
121 197 43 286 46 276 66 253 188 223 231 187 87 5 57 222 95 285 ^
636 0 251 248 37 233 16 224 220 230 102 123 3 141 98 133 278 56 84 165
235 91
229 49 157 278 2 202 295 98 110 217 19 1 203 20 137 108 5 193 42 116
122 270
236 7 268 47 242 155 43 115 206 125 238 28 94 6 32 244 207 284 204 50
42 199
167 127 153 45 53 93 121 111 79 282 129 151 213 166 97 57 60 188 53 10
182 148
162 244 47 105 37 210 61 187 137 268 184 63 285 97 134 194 201 82 250
112 132
142 257 290 109 4 277 3 118 37 96 195 94 60 211 114 47 265 110 80 198
168 149 1
43 138 267 268 24 287 12 257 153 288 171 117 281 130 67 252 164 76 231
101 101
261 143 13 256 237 143 173 32 107 89 154 44 31 245 123 26 202 135 216
240 80 38
286 279 55 62 126 237 6 55 8 244 70 166 74 293 23 105 91 92 228 27 118
280 214
28 254 79 199 84 188 296 128 15 11 181 232 17 246 259 228 187 150 54 9
266 109

164 159 170 64 18 2 225 59 135 193 179 43 252 27 255 101 47 82 273 173
275 225
219 17 63 125 24 264 57 220 70 29 253 163 160 86 103 16 87 102 100 48
104 15
171 63 102 260 227 271 258 152 266 132 119 196 253 26 42 62 49 103 99
26 65 17
86 80 61 50 19 138 24 44 191 127 99 269 183 147 140 99 49 20 54 169 42
276 4
140 182 94 280 77 207 124 25 180 96 218 155 203 294 76 71 219 231 124
162 1 226
272 174 191 264 217 102 237 166 234 175 190 51 72 104 152 289 87 262
279 215
241 293 77 284 67 229 209 39 148 139 183 10 22 75 292 1 66 93 7 54 283
205 72
81 197 252 4 3 64 261 120 58 113 154 181 255 78 283 45 190 211 251 271
98 270
99 186 138 184 223 61 73 80 175 107 6 215 81 114 207 147 86 26 179 180
73 157
118 120 35 158 11 98 293 160 32 221 146 247 121 88 18 139 46 189 189
212 273 13
96 58 273 36 51 95 59 295 75 49 180 233 39 285 257 16 64 41 63 262 142
97 111
239 58 296 288 194 178 21 213 29 170 224 197 169 65 40 41 223 144 12
234 92 145
88 155 66 176 243 208 200 243 48 21 76 208 40 64 178 181 95 196 116
205 209 199
30 9 23 288 34 296 151 76 48 264 194 41 200 83 272 151 130 227 134 34
22 212
113 131 226 112 292 24 127 236 90 16 10 159 193 19 161 85 187 274 89
17 223 7
191 104 38 126 100 185 222 145 225 82 90 230 262 5 276 87 79 39 11 21
53 147 3
278 216 263 35 75 72 241 79 122 247 281 267 185 85 27 95 219 71 242
292 161 245
69 144 35 295 235 228 50 179 20 287 172 139 23 128 152 129 14 95 36
234 38 72
218 18 85 164 22 52 231 261 280 131 89 291 162 161 108 142 14 172 276
91 ^
635 1 256 9 146 297 61 175 261 50 198 27 25 164 284 115 173 192 282
254 94 23
41 273 84 95 180 3 102 227 290 213 187 50 90 139 150 73 64 2 190 122
75 124 181
60 263 56 134 109 21 35 38 201 99 10 149 215 102 173 47 226 42 239 133
297 115
119 248 195 9 183 166 199 238 130 279 74 121 163 19 265 270 257 66 8
126 204 71
294 234 246 89 173 101 170 29 85 44 154 256 154 202 210 115 65 5 108
163 73 209
58 103 176 264 267 40 25 10 41 28 28 82 26 113 258 69 26 167 202 156
171 277
159 297 7 231 174 104 88 70 83 29 31 67 111 44 46 69 80 111 289 2 82
234 1 33

160 246 280 285 27 79 108 72 260 84 149 124 289 247 38 147 55 222 4
105 245 32
211 272 226 133 52 59 206 194 6 265 25 201 153 253 236 161 212 30 188
261 195
254 239 79 54 28 273 138 39 167 148 212 184 132 49 141 65 196 220 89
281 288
244 120 106 57 137 127 62 33 123 78 142 222 58 49 232 177 22 129 68
168 6 74 13
275 244 174 50 144 83 237 218 174 158 32 157 29 239 287 241 202 65 59
52 250 19
20 214 45 12 218 282 46 106 248 249 53 270 12 10 95 271 15 28 136 269
100 204
279 182 9 38 110 286 84 240 236 215 156 117 34 72 152 96 75 33 69 134
55 77 247
192 91 142 168 114 62 87 67 165 287 33 15 190 68 131 137 235 249 107
214 99 255
205 228 256 63 90 85 206 285 118 191 34 280 40 270 66 253 182 217 225
184 78 5
54 219 86 64 247 246 35 229 14 224 218 228 102 119 3 139 96 129 276 56
80 165
233 89 227 45 157 274 295 200 295 92 108 217 17 296 203 14 133 106 3
193 40 116
122 266 236 7 268 43 242 151 39 113 204 125 238 22 92 30 242 205 282
200 50 36
195 163 127 153 45 49 87 121 109 75 278 125 147 209 166 91 51 56 188
53 8 180
148 158 240 45 103 31 208 55 185 135 266 182 63 281 97 132 194 197 78
250 110
132 140 253 290 105 275 298 118 37 90 193 94 60 209 112 47 263 110 78
196 166
145 294 43 134 263 264 22 283 12 255 151 284 169 117 279 128 65 250
164 76 227
99 101 261 141 11 252 233 143 171 32 103 85 152 42 31 245 123 24 198
135 212
238 76 36 286 275 53 62 124 235 4 51 2 244 66 162 70 291 17 105 91 88
224 27
116 276 210 24 252 79 199 84 186 292 128 11 7 177 230 17 242 259 226
187 295
146 52 5 262 107 160 155 168 60 18 221 55 131 191 179 37 252 25 255 97
41 82
271 169 271 223 215 11 61 121 18 262 53 216 68 23 249 161 158 86 101
16 85 98
98 44 100 13 167 57 100 258 223 269 254 148 266 128 119 194 251 22 38
56 43 275
^
642 0 91 26 53 13 78 80 53 42 11 134 24 44 191 123 87 261 175 143 140
99 45 16
54 161 42 276 132 174 82 276 65 199 120 17 180 92 210 151 195 290 72
63 219 231
116 158 1 222 272 170 187 264 209 90 237 162 226 167 182 43 72 92 152
285 87

254 275 211 241 293 77 284 55 229 201 35 140 135 179 2 18 63 284 298
62 93 300
42 283 201 64 73 189 244 4 292 52 253 120 46 113 146 181 247 74 279 41
186 203
251 263 90 266 95 186 138 184 223 61 61 68 175 107 6 207 81 114 207
139 74 14
175 172 73 153 110 112 27 150 11 98 285 152 20 221 146 239 113 88 14
139 38 181
189 208 265 5 88 50 273 24 51 91 55 287 67 49 176 233 31 281 257 8 64
37 55 258
134 89 111 231 58 292 288 186 174 17 209 25 170 216 197 169 61 32 41
215 136 4
234 92 145 84 155 54 168 243 208 200 239 48 21 68 204 40 60 178 177 87
196 108
197 205 195 18 5 23 284 26 296 147 64 44 256 190 33 196 79 264 151 130
223 126
34 14 204 105 127 218 112 292 16 119 228 82 4 299 159 185 19 153 81
179 274 85
9 219 296 183 104 30 126 92 181 218 141 225 70 90 230 262 294 268 75
75 39 300
13 49 147 300 278 216 255 35 75 68 233 67 114 243 273 267 185 77 19 95
215 71
238 288 157 241 57 144 23 291 227 228 46 171 12 289 283 172 131 19 124
148 125
2 83 36 230 26 60 214 10 73 160 10 44 227 257 280 131 77 291 162 161
100 142 14
168 272 141 254 3 144 295 57 173 257 44 198 23 19 160 282 111 173 192
278 252
88 21 39 271 80 89 176 298 98 227 286 213 185 48 86 135 150 69 58 299
188 118
73 124 177 58 263 54 132 109 15 29 36 201 93 6 149 215 100 171 41 222
36 237
133 293 113 117 248 193 9 183 162 197 234 128 279 70 119 163 15 265
270 255 64
8 122 204 69 294 232 242 89 169 97 166 27 85 38 152 256 154 200 210
115 61 3
108 161 71 209 54 103 176 262 265 34 21 10 37 26 24 80 24 109 258 69
22 165 202
154 171 277 155 297 1 229 172 102 86 70 83 29 29 67 107 42 40 67 78
109 287 2
76 230 298 33 160 246 278 281 21 75 104 66 260 78 145 120 289 247 38
145 55 222
103 241 28 211 270 226 131 48 57 202 190 4 263 25 199 149 251 234 157
212 30
184 259 195 254 239 77 48 28 269 136 33 167 146 210 180 128 43 137 65
192 220
85 281 286 242 118 102 57 137 123 58 27 121 78 138 220 56 47 232 177
22 129 68
168 74 7 275 240 174 46 142 79 235 218 170 158 30 155 25 235 283 237
198 59 59
52 246 13 20 214 39 6 214 282 46 106 246 249 47 268 10 8 93 271 15 22
136 269

100 200 275 42 ^
644 0 6 29 104 280 81 240 230 212 156 114 31 69 146 96 66 27 63 134 52
68 244
186 88 139 162 111 53 81 64 162 287 30 6 190 59 128 131 235 246 101
211 93 249
205 222 250 57 87 76 206 285 115 185 25 274 34 264 66 253 176 211 219
181 69 5
51 216 77 58 241 243 32 223 11 224 215 225 102 113 3 136 93 123 273 56
74 165
230 86 224 39 157 268 289 197 295 83 105 217 14 293 203 5 127 103 193
37 116
122 260 236 7 268 37 242 145 33 110 201 125 238 13 89 293 27 239 202
279 194 50
27 189 157 127 153 45 43 78 121 106 69 272 119 141 203 166 82 42 50
188 53 5
177 148 152 234 42 100 22 205 46 182 132 263 179 63 275 97 129 194 191
72 250
107 132 137 247 290 99 296 272 295 118 37 81 190 94 60 206 109 47 260
110 75
193 163 139 288 43 128 257 258 19 277 12 252 148 278 166 117 276 125
62 247 164
76 221 96 101 261 138 8 246 227 143 168 32 97 79 149 39 31 245 123 21
192 135
206 235 70 33 286 269 50 62 121 232 1 45 295 244 60 156 64 288 8 105
91 82 218
27 113 270 204 18 249 79 199 84 183 286 128 5 1 171 227 17 236 259 223
187 292
140 49 301 256 104 154 149 165 54 18 299 215 49 125 188 179 28 252 22
255 91 32
82 268 163 265 220 209 2 58 115 9 259 47 210 65 14 243 158 155 86 98
16 82 92
95 38 94 10 161 48 97 255 217 266 248 142 266 122 119 191 248 16 32 47
34 88 89
26 50 12 76 80 51 40 9 133 24 44 191 122 84 259 173 142 140 99 44 15
54 159 42
276 301 130 172 79 275 62 197 119 15 180 91 208 150 193 289 71 61 219
231 114
157 1 221 272 169 186 264 207 87 237 161 224 165 180 41 72 89 152 284
87 252
274 210 241 293 77 284 52 229 199 34 138 134 178 17 60 282 298 61 93
299 39 283
200 62 71 187 242 4 290 49 251 120 43 113 144 181 245 73 278 40 185
201 251 261
88 265 94 186 138 184 223 61 58 65 175 107 6 205 81 114 207 137 71 11
174 170
73 152 108 110 25 148 11 98 283 150 17 221 146 237 111 88 13 139 36
179 189 207
263 3 86 48 273 21 51 90 54 285 65 49 175 233 29 280 257 6 64 36 53
257 132 87
111 229 58 291 288 184 173 16 208 24 170 214 197 169 60 30 41 213 134
2 234 92

145 83 155 51 166 243 208 200 238 48 21 66 203 40 59 178 176 85 196
106 195 204
194 15 4 23 283 24 296 146 61 43 254 189 31 195 78 262 151 130 222 124
34 12
202 103 126 216 112 292 14 117 226 80 1 297 159 183 19 151 80 177 274
84 7 218
294 181 104 28 126 90 180 217 140 225 67 90 230 262 292 266 72 74 39
298 11 48
147 300 278 216 253 265 ^
638 1 75 65 227 58 108 240 267 267 185 71 13 95 212 71 235 285 154 238
48 144
14 288 221 228 43 165 6 283 280 172 125 16 121 145 122 297 74 36 227
17 51 211
4 64 157 1 38 224 254 280 131 68 291 162 161 94 142 14 165 269 141 251
298 141
292 51 170 251 35 198 17 10 154 279 105 173 192 272 249 79 18 36 268
74 80 170
292 92 227 280 213 182 45 80 129 150 63 49 296 185 112 70 124 171 55
263 51 129
109 6 20 33 201 84 149 215 97 168 32 216 27 234 133 287 110 114 248
190 9 183
156 194 228 125 279 64 116 163 9 265 270 252 61 8 116 204 66 294 229
236 89 163
91 160 24 85 29 149 256 154 197 210 115 55 108 158 68 209 48 103 176
259 262 25
15 10 31 23 18 77 21 103 258 69 16 162 202 151 171 277 149 297 296 226
169 99
83 70 83 29 26 67 101 39 31 64 75 106 284 2 67 224 295 33 160 246 275
275 12 69
98 57 260 69 139 114 289 247 38 142 55 222 298 100 235 22 211 267 226
128 42 54
196 184 1 260 25 196 143 248 231 151 212 30 178 256 195 254 239 74 39
28 263
133 24 167 143 207 174 122 34 131 65 186 220 79 281 283 239 115 96 57
137 117
52 18 118 78 132 217 53 44 232 177 22 129 68 168 295 74 302 275 234
174 40 139
73 232 218 164 158 27 152 19 229 277 231 192 50 59 52 240 4 20 214 30
301 208
282 46 106 243 249 38 265 7 5 90 271 15 13 136 269 100 194 269 182 4
23 100 276
79 240 226 210 156 112 29 67 142 96 60 23 59 134 50 62 242 182 86 137
158 109
47 77 62 160 287 28 190 53 126 127 235 244 97 209 89 245 205 218 246
53 85 70
206 285 113 181 19 270 30 260 66 253 172 207 215 179 63 5 49 214 71 54
237 241
30 219 9 224 213 223 102 109 3 134 91 119 271 56 70 165 228 84 222 35
157 264
285 195 295 77 103 217 12 291 203 303 123 101 302 193 35 116 122 256
236 7 268

33 242 141 29 108 199 125 238 7 87 289 25 237 200 277 190 50 21 185
153 127 153
45 39 72 121 104 65 268 115 137 199 166 76 36 46 188 53 3 175 148 148
230 40 98
16 203 40 180 130 261 177 63 271 97 127 194 187 68 250 105 132 135 243
290 95
294 270 293 118 37 75 188 94 60 204 107 47 258 110 73 191 161 135 284
43 124
253 254 17 273 12 250 146 274 164 117 274 123 60 245 164 76 217 94 101
261 136
6 242 223 143 166 32 93 75 147 37 31 245 123 19 188 135 202 233 66 31
286 265
48 62 119 230 303 41 291 244 56 152 60 286 2 105 91 78 214 27 111 266
200 14
247 79 199 84 181 282 128 1 301 167 225 17 232 259 221 187 290 136 47
299 252
102 150 145 163 157 ^
653 0 18 299 207 41 117 184 179 16 252 18 255 83 20 82 264 155 257 216
201 296
54 107 303 255 39 202 61 2 235 154 151 86 94 16 78 84 91 30 86 6 153
36 93 251
209 262 240 134 266 114 119 187 244 8 24 35 22 76 81 26 38 8 68 80 43
32 1 129
24 44 191 118 72 251 165 138 140 99 40 11 54 151 42 276 301 122 164 67
271 50
189 115 7 180 87 200 146 185 285 67 53 219 231 106 153 1 217 272 165
182 264
199 75 237 157 216 157 172 33 72 77 152 280 87 244 270 206 241 293 77
284 40
229 191 30 130 130 174 298 13 48 274 298 57 93 295 27 283 196 54 63
179 234 4
282 37 243 120 31 113 136 181 237 69 274 36 181 193 251 253 80 261 90
186 138
184 223 61 46 53 175 107 6 197 81 114 207 129 59 305 170 162 73 148
100 102 17
140 11 98 275 142 5 221 146 229 103 88 9 139 28 171 189 203 255 301 78
40 273 9
51 86 50 277 57 49 171 233 21 276 257 304 64 32 45 253 124 79 111 221
58 287
288 176 169 12 204 20 170 206 197 169 56 22 41 205 126 300 234 92 145
79 155 39
158 243 208 200 234 48 21 58 199 40 55 178 172 77 196 98 187 200 190 3
23 279
16 296 142 49 39 246 185 23 191 74 254 151 130 218 116 34 4 194 95 122
208 112
292 6 109 218 72 295 289 159 175 19 143 76 169 274 80 305 214 286 173
104 20
126 82 176 213 136 225 55 90 230 262 284 258 60 70 39 290 3 44 147 300
278 216
245 35 75 63 223 52 104 238 263 267 185 67 9 95 210 71 233 283 152 236
42 144 8

286 217 228 41 161 2 279 278 172 121 14 119 143 120 293 68 36 225 11
45 209 58
155 301 34 222 252 280 131 62 291 162 161 90 142 14 163 267 141 249
294 139 290
47 168 247 29 198 13 4 150 277 101 173 192 268 247 73 16 34 266 70 74
166 288
88 227 276 213 180 43 76 125 150 59 43 294 183 108 68 124 167 53 263
49 127 109
14 31 201 78 302 149 215 95 166 26 212 21 232 133 283 108 112 248 188
9 183 152
192 224 123 279 60 114 163 5 265 270 250 59 8 112 204 64 294 227 232
89 159 87
156 22 85 23 147 256 154 195 210 115 51 304 108 156 66 209 44 103 176
257 260
19 11 10 27 21 14 75 19 99 258 69 12 160 202 149 171 277 145 297 292
224 167 97
81 70 83 29 24 67 97 37 25 62 73 104 282 2 61 220 293 33 160 246 273
271 6 65
94 51 260 63 135 110 289 247 38 140 55 222 296 98 231 18 211 265 226
126 38 52
192 180 305 258 25 194 139 246 229 147 212 30 174 254 195 254 239 72
33 28 259
131 18 167 141 205 170 118 28 127 65 182 220 75 281 281 237 113 92 57
137 113
48 12 116 78 128 215 51 42 232 177 22 129 68 168 291 161 ^
653 0 296 275 228 174 34 136 67 229 218 158 158 24 149 13 223 271 225
186 41 59
52 234 302 20 214 21 295 202 282 46 106 240 249 29 262 4 2 87 271 15 4
136 269
100 188 263 182 1 14 94 270 76 240 220 207 156 109 26 64 136 96 51 17
53 134 47
53 239 176 83 134 152 106 38 71 59 157 287 25 298 190 44 123 121 235
241 91 206
83 239 205 212 240 47 82 61 206 285 110 175 10 264 24 254 66 253 166
201 209
176 54 5 46 211 62 48 231 238 27 213 6 224 210 220 102 103 3 131 88
113 268 56
64 165 225 81 219 29 157 258 279 192 295 68 100 217 9 288 203 297 117
98 302
193 32 116 122 250 236 7 268 27 242 135 23 105 196 125 238 305 84 283
22 234
197 274 184 50 12 179 147 127 153 45 33 63 121 101 59 262 109 131 193
166 67 27
40 188 53 172 148 142 224 37 95 7 200 31 177 127 258 174 63 265 97 124
194 181
62 250 102 132 132 237 290 89 291 267 290 118 37 66 185 94 60 201 104
47 255
110 70 188 158 129 278 43 118 247 248 14 267 12 247 143 268 161 117
271 120 57
242 164 76 211 91 101 261 133 3 236 217 143 163 32 87 69 144 34 31 245
123 16

182 135 196 230 60 28 286 259 45 62 116 227 303 35 285 244 50 146 54
283 300
105 91 72 208 27 108 260 194 8 244 79 199 84 178 276 128 302 298 161
222 17 226
259 218 187 287 130 44 296 246 99 144 139 160 44 18 299 205 39 115 183
179 13
252 17 255 81 17 82 263 153 255 215 199 294 53 105 301 254 37 200 60
306 233
153 150 86 93 16 77 82 90 28 84 5 151 33 92 250 207 261 238 132 266
112 119 186
243 6 22 32 19 73 79 26 35 7 66 80 41 30 306 128 24 44 191 117 69 249
163 137
140 99 39 10 54 149 42 276 301 120 162 64 270 47 187 114 5 180 86 198
145 183
284 66 51 219 231 104 152 1 216 272 164 181 264 197 72 237 156 214 155
170 31
72 74 152 279 87 242 269 205 241 293 77 284 37 229 189 29 128 129 173
297 12 45
272 298 56 93 294 24 283 195 52 61 177 232 4 280 34 241 120 28 113 134
181 235
68 273 35 180 191 251 251 78 260 89 186 138 184 223 61 43 50 175 107 6
195 81
114 207 127 56 303 169 160 73 147 98 100 15 138 11 98 273 140 2 221
146 227 101
88 8 139 26 169 189 202 253 300 76 38 273 6 51 85 49 275 55 49 170 233
19 275
257 303 64 31 43 252 122 77 111 219 58 286 288 174 168 11 203 19 170
204 197
169 55 20 41 203 124 299 234 92 145 78 155 36 156 243 208 200 233 48
21 56 198
40 54 178 171 75 196 96 185 199 189 306 23 278 14 296 141 46 38 244
184 21 190
73 252 151 130 217 114 34 2 192 93 121 206 112 292 4 107 216 70 293
112 ^
646 1 159 169 19 137 73 163 274 77 302 211 280 167 104 14 126 76 173
210 133
225 46 90 230 262 278 252 51 67 39 284 306 41 147 300 278 216 239 35
75 60 217
43 98 235 257 267 185 61 3 95 207 71 230 280 149 233 33 144 308 283
211 228 38
155 305 273 275 172 115 11 116 140 117 287 59 36 222 2 36 206 303 49
152 295 28
219 249 280 131 53 291 162 161 84 142 14 160 264 141 246 288 136 287
41 165 241
20 198 7 304 144 274 95 173 192 262 244 64 13 31 263 64 65 160 282 82
227 270
213 177 40 70 119 150 53 34 291 180 102 65 124 161 50 263 46 124 109
300 5 28
201 69 299 149 215 92 163 17 206 12 229 133 277 105 109 248 185 9 183
146 189
218 120 279 54 111 163 308 265 270 247 56 8 106 204 61 294 224 226 89
153 81

150 19 85 14 144 256 154 192 210 115 45 304 108 153 63 209 38 103 176
254 257
10 5 10 21 18 8 72 16 93 258 69 6 157 202 146 171 277 139 297 286 221
164 94 78
70 83 29 21 67 91 34 16 59 70 101 279 2 52 214 290 33 160 246 270 265
306 59 88
42 260 54 129 104 289 247 38 137 55 222 293 95 225 12 211 262 226 123
32 49 186
174 305 255 25 191 133 243 226 141 212 30 168 251 195 254 239 69 24 28
253 128
9 167 138 202 164 112 19 121 65 176 220 69 281 278 234 110 86 57 137
107 42 3
113 78 122 212 48 39 232 177 22 129 68 168 285 74 292 275 224 174 30
134 63 227
218 154 158 22 147 9 219 267 221 182 35 59 52 230 298 20 214 15 291
198 282 46
106 238 249 23 260 2 85 271 15 307 136 269 100 184 259 182 308 8 90
266 74 240
216 205 156 107 24 62 132 96 45 13 49 134 45 47 237 172 81 132 148 104
32 67 57
155 287 23 294 190 38 121 117 235 239 87 204 79 235 205 208 236 43 80
55 206
285 108 171 4 260 20 250 66 253 162 197 205 174 48 5 44 209 56 44 227
236 25
209 4 224 208 218 102 99 3 129 86 109 266 56 60 165 223 79 217 25 157
254 275
190 295 62 98 217 7 286 203 293 113 96 302 193 30 116 122 246 236 7
268 23 242
131 19 103 194 125 238 301 82 279 20 232 195 272 180 50 6 175 143 127
153 45 29
57 121 99 55 258 105 127 189 166 61 21 36 188 53 307 170 148 138 220
35 93 1
198 25 175 125 256 172 63 261 97 122 194 177 58 250 100 132 130 233
290 85 289
265 288 118 37 60 183 94 60 199 102 47 253 110 68 186 156 125 274 43
114 243
244 12 263 12 245 141 264 159 117 269 118 55 240 164 76 207 89 101 261
131 1
232 213 143 161 32 83 65 142 32 31 245 123 14 178 135 192 228 56 26
286 255 43
62 114 225 303 31 281 244 46 142 50 281 296 105 91 238 ^
647 0 200 27 104 252 186 240 79 199 84 174 268 128 298 294 153 218 17
218 259
214 187 283 122 40 292 238 95 136 131 156 36 18 299 197 31 107 179 179
1 252 13
255 73 5 82 259 145 247 211 191 286 49 97 293 250 29 192 56 298 225
149 146 86
89 16 73 74 86 20 76 1 143 21 88 246 199 257 230 124 266 104 119 182
239 309 14
20 7 61 71 26 23 3 58 80 33 22 302 124 24 44 191 113 57 241 155 133
140 99 35 6

54 141 42 276 301 112 154 52 266 35 179 110 308 180 82 190 141 175 280
62 43
219 231 96 148 1 212 272 160 177 264 189 60 237 152 206 147 162 23 72
62 152
275 87 234 265 201 241 293 77 284 25 229 181 25 120 125 169 293 8 33
264 298 52
93 290 12 283 191 44 53 169 224 4 272 22 233 120 16 113 126 181 227 64
269 31
176 183 251 243 70 256 85 186 138 184 223 61 31 38 175 107 6 187 81
114 207 119
44 295 165 152 73 143 90 92 7 130 11 98 265 132 301 221 146 219 93 88
4 139 18
161 189 198 245 296 68 30 273 305 51 81 45 267 47 49 166 233 11 271
257 299 64
27 35 248 114 69 111 211 58 282 288 166 164 7 199 15 170 196 197 169
51 12 41
195 116 295 234 92 145 74 155 24 148 243 208 200 229 48 21 48 194 40
50 178 167
67 196 88 177 195 185 299 306 23 274 6 296 137 34 34 236 180 13 186 69
244 151
130 213 106 34 305 184 85 117 198 112 292 307 99 208 62 285 279 159
165 19 133
71 159 274 75 300 209 276 163 104 10 126 72 171 208 131 225 40 90 230
262 274
248 45 65 39 280 304 39 147 300 278 216 235 35 75 58 213 37 94 233 253
267 185
57 310 95 205 71 228 278 147 231 27 144 304 281 207 228 36 151 303 269
273 172
111 9 114 138 115 283 53 36 220 307 30 204 301 43 150 291 24 217 247
280 131 47
291 162 161 80 142 14 158 262 141 244 284 134 285 37 163 237 14 198 3
300 140
272 91 173 192 258 242 58 11 29 261 60 59 156 278 78 227 266 213 175
38 66 115
150 49 28 289 178 98 63 124 157 48 263 44 122 109 296 310 26 201 63
297 149 215
90 161 11 202 6 227 133 273 103 107 248 183 9 183 142 187 214 118 279
50 109
163 306 265 270 245 54 8 102 204 59 294 222 222 89 149 77 146 17 85 8
142 256
154 190 210 115 41 304 108 151 61 209 34 103 176 252 255 4 1 10 17 16
4 70 14
89 258 69 2 155 202 144 171 277 135 297 282 219 162 92 76 70 83 29 19
67 87 32
10 57 68 99 277 2 46 210 288 33 160 246 268 261 302 55 84 36 260 48
125 100 289
247 38 135 55 222 291 93 221 8 211 260 226 121 28 47 182 170 305 253
25 189 129
241 224 137 212 30 164 249 195 254 239 86 ^
665 0 15 28 247 125 167 135 199 158 106 10 115 65 170 220 63 281 275
231 107 80

57 137 101 36 306 110 78 116 209 45 36 232 177 22 129 68 168 279 74
286 275 218
174 24 131 57 224 218 148 158 19 144 3 213 261 215 176 26 59 52 224
292 20 214
6 285 192 282 46 106 235 249 14 257 311 309 82 271 15 301 136 269 100
178 253
182 308 311 84 260 71 240 210 202 156 104 21 59 126 96 36 7 43 134 42
38 234
166 78 129 142 101 23 61 54 152 287 20 288 190 29 118 111 235 236 81
201 73 229
205 202 230 37 77 46 206 285 105 165 307 254 14 244 66 253 156 191 199
171 39 5
41 206 47 38 221 233 22 203 1 224 205 215 102 93 3 126 83 103 263 56
54 165 220
76 214 19 157 248 269 187 295 53 95 217 4 283 203 287 107 93 302 193
27 116 122
240 236 7 268 17 242 125 13 100 191 125 238 295 79 273 17 229 192 269
174 50
309 169 137 127 153 45 23 48 121 96 49 252 99 121 183 166 52 12 30 188
53 307
167 148 132 214 32 90 304 195 16 172 122 253 169 63 255 97 119 194 171
52 250
97 132 127 227 290 79 286 262 285 118 37 51 180 94 60 196 99 47 250
110 65 183
153 119 268 43 108 237 238 9 257 12 242 138 258 156 117 266 115 52 237
164 76
201 86 101 261 128 310 226 207 143 158 32 77 59 139 29 31 245 123 11
172 135
186 225 50 23 286 249 40 62 111 222 303 25 275 244 40 136 44 278 290
105 91 62
198 27 103 250 184 310 239 79 199 84 173 266 128 297 293 151 217 17
216 259 213
187 282 120 39 291 236 94 134 129 155 34 18 299 195 29 105 178 179 310
252 12
255 71 2 82 258 143 245 210 189 284 48 95 291 249 27 190 55 296 223
148 145 86
88 16 72 72 85 18 74 141 18 87 245 197 256 228 122 266 102 119 181 238
308 12
17 4 58 69 26 20 2 56 80 31 20 301 123 24 44 191 112 54 239 153 132
140 99 34 5
54 139 42 276 301 110 152 49 265 32 177 109 307 180 81 188 140 173 279
61 41
219 231 94 147 1 211 272 159 176 264 187 57 237 151 204 145 160 21 72
59 152
274 87 232 264 200 241 293 77 284 22 229 179 24 118 124 168 292 7 30
262 298 51
93 289 9 283 190 42 51 167 222 4 270 19 231 120 13 113 124 181 225 63
268 30
175 181 251 241 68 255 84 186 138 184 223 61 28 35 175 107 6 185 81
114 207 117
41 293 164 150 73 142 88 90 5 128 11 98 263 130 299 221 146 217 91 88
3 139 16

159 189 197 243 295 66 28 273 303 51 80 44 265 45 49 165 233 9 270 257
298 64
26 33 247 112 67 111 209 58 281 288 164 163 6 198 14 170 194 197 169
50 10 41
193 114 294 234 92 145 73 155 21 146 243 208 200 228 48 21 46 193 40
49 178 166
65 196 86 175 194 184 297 306 23 273 206 ^
641 0 296 134 25 31 230 177 7 183 66 238 151 130 210 100 34 302 178 79
114 192
112 292 304 93 202 56 279 273 159 159 19 127 68 153 274 72 297 206 270
157 104
4 126 66 168 205 128 225 31 90 230 262 268 242 36 62 39 274 301 36 147
300 278
216 229 35 75 55 207 28 88 230 247 267 185 51 307 95 202 71 225 275
144 228 18
144 298 278 201 228 33 145 300 263 270 172 105 6 111 135 112 277 44 36
217 301
21 201 298 34 147 285 18 214 244 280 131 38 291 162 161 74 142 14 155
259 141
241 278 131 282 31 160 231 5 198 311 294 134 269 85 173 192 252 239 49
8 26 258
54 50 150 272 72 227 260 213 172 35 60 109 150 43 19 286 175 92 60 124
151 45
263 41 119 109 290 304 23 201 54 294 149 215 87 158 2 196 311 224 133
267 100
104 248 180 9 183 136 184 208 115 279 44 106 163 303 265 270 242 51 8
96 204 56
294 219 216 89 143 71 140 14 85 313 139 256 154 187 210 115 35 304 108
148 58
209 28 103 176 249 252 309 309 10 11 13 312 67 11 83 258 69 310 152
202 141 171
277 129 297 276 216 159 89 73 70 83 29 16 67 81 29 1 54 65 96 274 2 37
204 285
33 160 246 265 255 296 49 78 27 260 39 119 94 289 247 38 132 55 222
288 90 215
2 211 257 226 118 22 44 176 164 305 250 25 186 123 238 221 131 212 30
158 246
195 254 239 64 9 28 243 123 308 167 133 197 154 102 4 111 65 166 220
59 281 273
229 105 76 57 137 97 32 302 108 78 112 207 43 34 232 177 22 129 68 168
275 74
282 275 214 174 20 129 53 222 218 144 158 17 142 313 209 257 211 172
20 59 52
220 288 20 214 281 188 282 46 106 233 249 8 255 311 309 80 271 15 297
136 269
100 174 249 182 308 307 80 256 69 240 206 200 156 102 19 57 122 96 30
3 39 134
40 32 232 162 76 127 138 99 17 57 52 150 287 18 284 190 23 116 107 235
234 77
199 69 225 205 198 226 33 75 40 206 285 103 161 303 250 10 240 66 253
152 187

195 169 33 5 39 204 41 34 217 231 20 199 313 224 203 213 102 89 3 124
81 99 261
56 50 165 218 74 212 15 157 244 265 185 295 47 93 217 2 281 203 283
103 91 302
193 25 116 122 236 236 7 268 13 242 121 9 98 189 125 238 291 77 269 15
227 190
267 170 50 305 165 133 127 153 45 19 42 121 94 45 248 95 117 179 166
46 6 26
188 53 307 165 148 128 210 30 88 300 193 10 170 120 251 167 63 251 97
117 194
167 48 250 95 132 125 223 290 75 284 260 283 118 37 45 178 94 60 194
97 47 248
110 63 181 151 115 264 43 104 233 234 7 253 12 240 136 254 154 117 264
113 50
235 164 76 197 84 101 261 126 310 222 203 143 276 ^
656 0 32 69 51 135 25 31 245 123 7 164 135 178 221 42 19 286 241 36 62
107 218
303 17 267 244 32 128 36 274 282 105 91 54 190 27 99 242 176 306 235
79 199 84
169 258 128 293 289 143 213 17 208 259 209 187 278 112 35 287 228 90
126 121
151 26 18 299 187 21 97 174 179 302 252 8 255 63 306 82 254 135 237
206 181 276
44 87 283 245 19 182 51 288 215 144 141 86 84 16 68 64 81 10 66 312
133 6 83
241 189 252 220 114 266 94 119 177 234 304 4 5 308 46 61 26 8 314 48
80 23 12
297 119 24 44 191 108 42 231 145 128 140 99 30 1 54 131 42 276 301 102
144 37
261 20 169 105 303 180 77 180 136 165 275 57 33 219 231 86 143 1 207
272 155
172 264 179 45 237 147 196 137 152 13 72 47 152 270 87 224 260 196 241
293 77
284 10 229 171 20 110 120 164 288 3 18 254 298 47 93 285 313 283 186
34 43 159
214 4 262 7 223 120 1 113 116 181 217 59 264 26 171 173 251 233 60 251
80 186
138 184 223 61 16 23 175 107 6 177 81 114 207 109 29 285 160 142 73
138 80 82
313 120 11 98 255 122 291 221 146 209 83 88 315 139 8 151 189 193 235
291 58 20
273 295 51 76 40 257 37 49 161 233 1 266 257 294 64 22 25 243 104 59
111 201 58
277 288 156 159 2 194 10 170 186 197 169 46 2 41 185 106 290 234 92
145 69 155
9 138 243 208 200 224 48 21 38 189 40 45 178 162 57 196 78 167 190 180
289 306
23 269 312 296 132 19 29 226 175 3 181 64 234 151 130 208 96 34 300
174 75 112
188 112 292 302 89 198 52 275 269 159 155 19 123 66 149 274 70 295 204
266 153

104 126 62 166 203 126 225 25 90 230 262 264 238 30 60 39 270 299 34
147 300
278 216 225 35 75 53 203 22 84 228 243 267 185 47 305 95 200 71 223
273 142 226
12 144 294 276 197 228 31 141 298 259 268 172 101 4 109 133 110 273 38
36 215
297 15 199 296 28 145 281 14 212 242 280 131 32 291 162 161 70 142 14
153 257
141 239 274 129 280 27 158 227 315 198 309 290 130 267 81 173 192 248
237 43 6
24 256 50 44 146 268 68 227 256 213 170 33 56 105 150 39 13 284 173 88
58 124
147 43 263 39 117 109 286 300 21 201 48 292 149 215 85 156 312 192 307
222 133
263 98 102 248 178 9 183 132 182 204 113 279 40 104 163 301 265 270
240 49 8 92
204 54 294 217 212 89 139 67 136 12 85 309 137 256 154 185 210 115 31
304 108
146 56 209 24 103 176 247 250 305 307 10 7 11 310 65 9 79 258 69 308
150 202
139 171 277 125 297 272 214 157 87 71 70 83 29 14 67 77 27 311 52 63
94 272 2
31 200 283 33 160 246 263 251 292 45 74 21 260 33 115 90 289 247 38
130 55 222
286 88 211 227 ^
668 0 211 254 226 115 16 41 170 158 305 247 25 183 117 235 218 125 212
30 152
243 195 254 239 61 28 237 120 302 167 130 194 148 96 312 105 65 160
220 53 281
270 226 102 70 57 137 91 26 296 105 78 106 204 40 31 232 177 22 129 68
168 269
74 276 275 208 174 14 126 47 219 218 138 158 14 139 310 203 251 205
166 11 59
52 214 282 20 214 308 275 182 282 46 106 230 249 316 252 311 309 77
271 15 291
136 269 100 168 243 182 308 301 74 250 66 240 200 197 156 99 16 54 116
96 21
314 33 134 37 23 229 156 73 124 132 96 8 51 49 147 287 15 278 190 14
113 101
235 231 71 196 63 219 205 192 220 27 72 31 206 285 100 155 297 244 4
234 66 253
146 181 189 166 24 5 36 201 32 28 211 228 17 193 313 224 200 210 102
83 3 121
78 93 258 56 44 165 215 71 209 9 157 238 259 182 295 38 90 217 316 278
203 277
97 88 302 193 22 116 122 230 236 7 268 7 242 115 3 95 186 125 238 285
74 263 12
224 187 264 164 50 299 159 127 127 153 45 13 33 121 91 39 242 89 111
173 166 37
314 20 188 53 307 162 148 122 204 27 85 294 190 1 167 117 248 164 63
245 97 114

194 161 42 250 92 132 122 217 290 69 281 257 280 118 37 36 175 94 60
191 94 47
245 110 60 178 148 109 258 43 98 227 228 4 247 12 237 133 248 151 117
261 110
47 232 164 76 191 81 101 261 123 310 216 197 143 153 32 67 49 134 24
31 245 123
6 162 135 176 220 40 18 286 239 35 62 106 217 303 15 265 244 30 126 34
273 280
105 91 52 188 27 98 240 174 305 234 79 199 84 168 256 128 292 288 141
212 17
206 259 208 187 277 110 34 286 226 89 124 119 150 24 18 299 185 19 95
173 179
300 252 7 255 61 304 82 253 133 235 205 179 274 43 85 281 244 17 180
50 286 213
143 140 86 83 16 67 62 80 8 64 312 131 3 82 240 187 251 218 112 266 92
119 176
233 303 2 2 306 43 59 26 5 314 46 80 21 10 296 118 24 44 191 107 39
229 143 127
140 99 29 54 129 42 276 301 100 142 34 260 17 167 104 302 180 76 178
135 163
274 56 31 219 231 84 142 1 206 272 154 171 264 177 42 237 146 194 135
150 11 72
44 152 269 87 222 259 195 241 293 77 284 7 229 169 19 108 119 163 287
2 15 252
298 46 93 284 311 283 185 32 41 157 212 4 260 4 221 120 315 113 114
181 215 58
263 25 170 171 251 231 58 250 79 186 138 184 223 61 13 20 175 107 6
175 81 114
207 107 26 283 159 140 73 137 78 80 312 118 11 98 253 120 289 221 146
207 81 88
315 139 6 149 189 192 233 290 56 18 273 293 51 75 39 255 35 49 160 233
316 265
257 293 64 21 23 242 102 57 111 199 58 276 288 154 158 1 193 9 170 184
197 169
45 41 183 104 289 234 92 145 68 155 6 136 243 208 203 ^
656 0 221 48 21 32 186 40 42 178 159 51 196 72 161 187 177 283 306 23
266 309
296 129 10 26 220 172 316 178 61 228 151 130 205 90 34 297 168 69 109
182 112
292 299 83 192 46 269 263 159 149 19 117 63 143 274 67 292 201 260 147
104 313
126 56 163 200 123 225 16 90 230 262 258 232 21 57 39 264 296 31 147
300 278
216 219 35 75 50 197 13 78 225 237 267 185 41 302 95 197 71 220 270
139 223 3
144 288 273 191 228 28 135 295 253 265 172 95 1 106 130 107 267 29 36
212 291 6
196 293 19 142 275 8 209 239 280 131 23 291 162 161 64 142 14 150 254
141 236
268 126 277 21 155 221 309 198 306 284 124 264 75 173 192 242 234 34 3
21 253

44 35 140 262 62 227 250 213 167 30 50 99 150 33 4 281 170 82 55 124
141 40 263
36 114 109 280 294 18 201 39 289 149 215 82 153 306 186 301 219 133
257 95 99
248 175 9 183 126 179 198 110 279 34 101 163 298 265 270 237 46 8 86
204 51 294
214 206 89 133 61 130 9 85 303 134 256 154 182 210 115 25 304 108 143
53 209 18
103 176 244 247 299 304 10 1 8 307 62 6 73 258 69 305 147 202 136 171
277 119
297 266 211 154 84 68 70 83 29 11 67 71 24 305 49 60 91 269 2 22 194
280 33 160
246 260 245 286 39 68 12 260 24 109 84 289 247 38 127 55 222 283 85
205 311 211
252 226 113 12 39 166 154 305 245 25 181 113 233 216 121 212 30 148
241 195 254
239 59 313 28 233 118 298 167 128 192 144 92 308 101 65 156 220 49 281
268 224
100 66 57 137 87 22 292 103 78 102 202 38 29 232 177 22 129 68 168 265
74 272
275 204 174 10 124 43 217 218 134 158 12 137 308 199 247 201 162 5 59
52 210
278 20 214 304 271 178 282 46 106 228 249 312 250 311 309 75 271 15
287 136 269
100 164 239 182 308 297 70 246 64 240 196 195 156 97 14 52 112 96 15
312 29 134
35 17 227 152 71 122 128 94 2 47 47 145 287 13 274 190 8 111 97 235
229 67 194
59 215 205 188 216 23 70 25 206 285 98 151 293 240 230 66 253 142 177
185 164
18 5 34 199 26 24 207 226 15 189 313 224 198 208 102 79 3 119 76 89
256 56 40
165 213 69 207 5 157 234 255 180 295 32 88 217 316 276 203 273 93 86
302 193 20
116 122 226 236 7 268 3 242 111 318 93 184 125 238 281 72 259 10 222
185 262
160 50 295 155 123 127 153 45 9 27 121 89 35 238 85 107 169 166 31 310
16 188
53 307 160 148 118 200 25 83 290 188 314 165 115 246 162 63 241 97 112
194 157
38 250 90 132 120 213 290 65 279 255 278 118 37 30 173 94 60 189 92 47
243 110
58 176 146 105 254 43 94 223 224 2 243 12 235 131 244 149 117 259 108
45 230
164 76 187 79 101 261 109 ^
656 0 310 208 189 143 149 32 59 41 130 20 31 245 123 2 154 135 168 216
32 14
286 231 31 62 102 213 303 7 257 244 22 118 26 269 272 105 91 44 180 27
94 232
166 301 230 79 199 84 164 248 128 288 284 133 208 17 198 259 204 187
273 102 30

282 218 85 116 111 146 16 18 299 177 11 87 169 179 292 252 3 255 53
296 82 249
125 227 201 171 266 39 77 273 240 9 172 46 278 205 139 136 86 79 16 63
54 76 56
312 123 312 78 236 179 247 210 104 266 84 119 172 229 299 315 311 298
31 51 26
314 314 38 80 13 2 292 114 24 44 191 103 27 221 135 123 140 99 25 317
54 121 42
276 301 92 134 22 256 5 159 100 298 180 72 170 131 155 270 52 23 219
231 76 138
1 202 272 150 167 264 169 30 237 142 186 127 142 3 72 32 152 265 87
214 255 191
241 293 77 284 316 229 161 15 100 115 159 283 319 3 244 298 42 93 280
303 283
181 24 33 149 204 4 252 313 213 120 307 113 106 181 207 54 259 21 166
163 251
223 50 246 75 186 138 184 223 61 1 8 175 107 6 167 81 114 207 99 14
275 155 132
73 133 70 72 308 110 11 98 245 112 281 221 146 199 73 88 315 139 319
141 189
188 225 286 48 10 273 285 51 71 35 247 27 49 156 233 312 261 257 289
64 17 15
238 94 49 111 191 58 272 288 146 154 318 189 5 170 176 197 169 41 313
41 175 96
285 234 92 145 64 155 315 128 243 208 200 219 48 21 28 184 40 40 178
157 47 196
68 157 185 175 279 306 23 264 307 296 127 4 24 216 170 314 176 59 224
151 130
203 86 34 295 164 65 107 178 112 292 297 79 188 42 265 259 159 145 19
113 61
139 274 65 290 199 256 143 104 311 126 52 161 198 121 225 10 90 230
262 254 228
15 55 39 260 294 29 147 300 278 216 215 35 75 48 193 7 74 223 233 267
185 37
300 95 195 71 218 268 137 221 318 144 284 271 187 228 26 131 293 249
263 172 91
320 104 128 105 263 23 36 210 287 194 291 13 140 271 4 207 237 280 131
17 291
162 161 60 142 14 148 252 141 234 264 124 275 17 153 217 305 198 304
280 120
262 71 173 192 238 232 28 1 19 251 40 29 136 258 58 227 246 213 165 28
46 95
150 29 319 279 168 78 53 124 137 38 263 34 112 109 276 290 16 201 33
287 149
215 80 151 302 182 297 217 133 253 93 97 248 173 9 183 122 177 194 108
279 30
99 163 296 265 270 235 44 8 82 204 49 294 212 202 89 129 57 126 7 85
299 132
256 154 180 210 115 21 304 108 141 51 209 14 103 176 242 245 295 302
10 318 6
305 60 4 69 258 69 303 145 202 134 171 277 115 297 262 209 152 82 66
70 83 29 9

```
67 67 22 301 47 58 89 267 2 16 190 278 33 160 246 258 241 282 35 64 6
260 18
105 80 289 247 38 125 55 292 ^
<D
```

```
H>SHS Type 3 Strings<H
```

```
D>
```

```
45 0 14 5 3 1 4 16 12 20 1 6 15 11 18 4 17 16 6 10 3 2 9 9 14 6 2 8 6
7 10 17
12 20 6 7 5 16 1 4 2 17 10 15 8 20 1 ^
<D
```

```
File = cancel.auth-script
```

```
## this tests canceling EXTERNAL
```

```
SERVER
```

```
SEND 'AUTH EXTERNAL USERID_HEX'
```

```
EXPECT_COMMAND OK
```

```
EXPECT_STATE WAITING_FOR_INPUT
```

```
SEND 'CANCEL'
```

```
EXPECT_COMMAND REJECTED
```

```
EXPECT_STATE WAITING_FOR_INPUT
```

```
## now start over and see if it works
```

```
SEND 'AUTH EXTERNAL USERID_HEX'
```

```
EXPECT_COMMAND OK
```

```
EXPECT_STATE WAITING_FOR_INPUT
```

```
SEND 'BEGIN'
```

```
EXPECT_STATE AUTHENTICATED
```

```
File = ch01.html
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Introduction</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
```



```

<link rel="prev" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="next" href="ch02.html" title="API Reference">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2"><tr valign="middle">
<td><a accesskey="p" href="index.html"></a></td>
<td>Â </td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="ch02.html"></a></td>
</tr></table>
<div class="chapter">
<div class="titlepage"><div><div><h2 class="title">
<a name="idp744992"></a>Introduction</h2></div></div></div>
<p>D-Bus is a message bus system, a simple way for applications
to talk to one another.</p>
<p>D-Bus supplies both a system daemon (for events such as "new
hardware device added" or "printer queue changed") and a
per-user-login-session daemon (for general IPC needs among user
applications). Also, the message bus is built on top of a general
one-to-one message passing framework, which can be used by any two
apps to communicate directly (without going through the message
bus daemon). Currently the communicating applications are on one
computer, but TCP/IP option is available and remote support
planned.</p>
<p>The D-Bus API isn't finished yet, and the design is by no
means set in stone. One of our main goals is for lots of projects
to use it, so if you wouldn't use it, by all means mail us and say
why - design, licensing, indentation style, we would rather know
than not know.</p>
<p>D-Bus' only <span class="emphasis"><em>required</em></span>
dependency is an
XML parser (either
libxml or expat). D-Bus includes many language bindings to the
D-Bus system, if you want to build those. The list of projects
using D-Bus is growing and they provide a wealth of examples of
using the various APIs to learn from.</p>
<p>Trying out D-Bus in sample applications is encouraged - we
want to get it widely used and tested. It should be working
pretty well these days, and when it isn't bug reports are very
welcome.</p>
<p>For more information on D-Bus, see

```

```
        <a class="ulink"
href="http://www.freedesktop.org/wiki/Software/dbus" target="_top">
        http://www.freedesktop.org/wiki/Software/dbus</a>.
    </p>
</div>
<div class="footer">
<hr>
        Generated by GTK-Doc V1.18</div>
</body>
</html>
```

File = ch02.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>API Reference</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="prev" href="ch01.html" title="Introduction">
<link rel="next" href="dbus-glib-DBusGConnection.html"
title="DBusGConnection">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2"><tr valign="middle">
<td><a accesskey="p" href="ch01.html"></a></td>
<td>Ã </td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBusGConnection.html"></a></td>
</tr></table>
<div class="chapter">
<div class="titlepage"><div><div><h2 class="title">
<a name="idp748720"></a>API Reference</h2></div></div></div>
<div class="toc"><dl>
<dt><span class="sect1"><a href="ch02.html#api-overview">API for using
D-BUS with GLib</a></span></dt>
<dt>
```

```

<span class="refentrytitle"><a href="dbus-glib-
DBusGConnection.html">DBusGConnection</a></span><span
class="refpurpose"> â€” DBus Connection</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-DBus-GObject-related-
functions.html">DBus GObject related functions</a></span><span
class="refpurpose"> â€” Exporting a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
remotely</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-
DBusGMessage.html">DBusGMessage</a></span><span class="refpurpose">
â€” DBus Message</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-
DBusGMethod.html">DBusGMethod</a></span><span class="refpurpose"> â€”
GMethod Info &amp; Invocation</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-
DBusGError.html">DBusGError</a></span><span class="refpurpose"> â€”
DBus GError</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-
DBusGProxy.html">DBusGProxy</a></span><span class="refpurpose"> â€”
DBus Proxy</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-Specializable-GType-
System.html">Specializable GType System</a></span><span
class="refpurpose"> â€” Specialized GTypes</span>
</dt>
<dt>
<span class="refentrytitle"><a href="dbus-glib-DBus-GLib-low-
level.html">DBus GLib low level</a></span><span class="refpurpose">
â€” DBus lower level functions</span>
</dt>
</dl></div>
<div class="sect1">
<div class="titlepage"><div><div><h2 class="title" style="clear:
both">
<a name="api-overview"></a>API for using D-BUS with
GLib</h2></div></div></div>
<p>libdbus proper is a low-level API, these GLib bindings wrap
libdbus with a much higher-level approach. The higher level
approach is possible because GLib defines a main loop, an
object/type system, and an out-of-memory handling policy (it exits

```

```

    the program). See <a class="ulink" href="http://www.gtk.org"
target="_top">http://www.gtk.org</a> for GLib
    information.</p>
<p>To manipulate remote objects, use DBusGProxy. </p>
</div>
</div>
<div class="footer">
<hr>
    Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = ch03.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Tools Reference</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="prev" href="dbus-glib-DBus-GLib-low-level.html" title="DBus
GLib low level">
<link rel="next" href="dbus-binding-tool.html" title="dbus-binding-
tool">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2"><tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBus-GLib-low-level.html"></a></td>
<td>Â </td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-binding-tool.html"></a></td>
</tr></table>
<div class="chapter">
<div class="titlepage"><div><div><h2 class="title">
<a name="idp3754816"></a>Tools Reference</h2></div></div></div>
<div class="toc"><dl><dt>

```

```
<span class="refentrytitle"><a href="dbus-binding-tool.html">dbus-
binding-tool</a></span><span class="refpurpose"> â€” C language GLib
bindings generation utility</span>
</dt></dl></div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>
```

File = ChangeLog

```
commit 02b44f8427ea48463d7b9f2a9f7099be7643b60d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2013-02-25 12:24:35 +0000
```

0.102: respin tarball

```
configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 78f3e953e4a05a1126874dd2ca6cec5777a15b3b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2013-02-15 12:37:04 +0000
```

0.100.1 point release

```
configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 166978a09cf5edff4028e670b6074215a4c75eca
Author: Colin Walters <walters@verbum.org>
Date: 2013-02-14 10:19:34 -0500
```

CVE-2013-0292: dbus-gproxy: Verify sender of NameOwnerChanged signals to be o.f.DBus

Anyone can hop on the bus and emit a signal whose interface is o.f.DBus; it's expected at the moments that clients (and notably DBus libraries) check the sender.

This could previously be used to trick a system service using dbus-glib into thinking a malicious signal came from a privileged source, by claiming that ownership of the privileged source's well-known name had changed from the privileged source's real unique name to the attacker's

unique name.

[altered to be NULL-safe so it won't crash on peer connections - smcv]

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gproxy.c | 7 ++++---

1 file changed, 4 insertions(+), 3 deletions(-)

commit c6cbdf9ed99f82983dd529319475dd02c53ad2aa

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2012-06-25 18:23:30 +0100

Fix builds without tests

test/manual/Makefile.am | 5 +++++

1 file changed, 5 insertions(+)

commit a0dd0c8c240896a5dd205c1dbc51924b9d41f833

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2012-06-25 17:26:48 +0100

Prepare version 0.100

configure.ac | 4 ++--

1 file changed, 2 insertions(+), 2 deletions(-)

commit 8b7e4a1c4ae055864e26db4addbcc2dc38ee6963

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2012-04-16 12:11:38 +0100

Fix two crashes when `dbus_g_proxy_new_for_peer` is used on a bus

The first part of the bug is that when `NameOwnerChanged` is received

with

a `dbus_g_proxy_new_for_peer` (which has no name) alive, checking whether it was affected by the `NameOwnerChanged` caused a NULL dereference and segfault.

The second part of the bug is that if the last proxy in existence is

for a peer, when it was unregistered there would be no `owner_match_rules`, causing a crash.

Both are exercised in the new test added here.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=41126

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
.gitignore | 1 +
dbus/dbus-gproxy.c | 5 +-
test/core/Makefile.am | 3 +
test/core/peer-on-bus.c | 148
+++++
test/core/run-test.sh | 1 +
5 files changed, 156 insertions(+), 2 deletions(-)
```

commit 82d45d15088d206f1e1f024215e95301fa04e92c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2012-04-16 12:10:37 +0100

Add a utility function to tear down a private connection in tests

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=41126
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
configure.ac | 1 +
test/Makefile.am | 4 +-
test/core/Makefile.am | 4 +-
test/interfaces/Makefile.am | 4 +-
test/lib/Makefile.am | 19 ++++++++
test/lib/util.c | 69
+++++
test/lib/util.h | 32 ++++++++
7 files changed, 129 insertions(+), 4 deletions(-)
```

commit 403c5bf4d98fcba83caa94aee96da27128bbf9c8
Author: Matthias Klose <doko@ubuntu.com>
Date: 2012-06-25 17:11:45 +0100

Fix build error with -Werror=format-security

Origin: vendor, Ubuntu
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
dbus/dbus-gobject.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit ae0ed3e435c550c3fae1a86a1c3cb42da319dff2
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 17:42:44 +0100

Move use of 0 as an error domain into the invalid-usage test

I think this is invalid, although others might disagree.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
test/core/error-mapping.c | 26 -----
test/manual/invalid-usage.c | 25 ++++++
2 files changed, 25 insertions(+), 26 deletions(-)
```

commit 92c613b8f429a4fe6c762805f1a4f3588511e6a7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 17:35:42 +0100

Add a manual test for various invalid behaviour

Most of this has been sitting in a branch since fd.o #30171; fixing fd.o #40151, another case of library-user error leading to undefined behaviour and a hard-to-diagnose crash, seems a good time to get this merged.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
.gitignore | 1 +
configure.ac | 1 +
test/Makefile.am | 4 +-
test/manual/Makefile.am | 27 ++++
test/manual/invalid-usage.c | 295
+++++
5 files changed, 326 insertions(+), 2 deletions(-)
```

commit 539bd92b8703134057c039f965ff7a5b5434c3a7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 17:41:35 +0100

Remove tests in test-dbus-glib which basically just test error mapping

Also remove the methods on MyObject that only existed to support these tests; ThrowError is now versatile enough to implement them all.

Leave ThrowUnregisteredError as it is, since it violates GError expectations (see GNOME#660371), but stop using it in test-dbus-glib - it's enough to use it in test-error-mapping.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
test/core/my-object.c | 36 -----
```

-


```
test/core/test-dbus-glib.c      |    45
-----
test/core/test-service-glib.xml |     9 -----
3 files changed, 90 deletions(-)
```

commit b0dc0f419b641f4ab8fdd8eaf69f4ed4c2d61159
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 14:56:45 +0100

Add a new test for error mapping

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
.gitignore                      |    1 +
test/core/Makefile.am          |    7 +
test/core/error-mapping.c     |   327
+++++
test/core/run-test.sh         |    1 +
4 files changed, 336 insertions(+)
```

commit 5aa25d4f3c44e7f46e95b8d8de64bf63f2c7c42e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 17:23:18 +0100

MyObject: make ThrowError, AsyncThrowError throw a caller-
specified
error

This can be used to test arbitrary errors, but only in-process; in
tests with the service out-of-process, like test-dbus-glib,
the initial
error (matching the error they previously threw) will always be
used.

This obsoletes ThrowErrorUnderscore, ThrowErrorMultiWord and
ThrowNotSupported, but not ThrowUnregisteredError due to some
strange
assumptions about the validity of GError domains in that method
(see GNOME#660731).

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
test/core/my-object.c |    49
+++++
test/core/my-object.h |    5 +++++
2 files changed, 40 insertions(+), 14 deletions(-)
```

commit df3949806818ba0eab495fcaad2c7f3ad38007c7

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 16:14:52 +0100

test-dbus-glib.c isn't GTest yet, but add bug numbers anyway

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

test/core/test-dbus-glib.c | 4 +++
1 file changed, 4 insertions(+)

commit ead4ef065f700a4b586e78400789a7a18c856cc4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 14:57:02 +0100

Add copyright/licensing information to test-dbus-glib

I've tried to dig up the copyright holders from git history;
possibly
incomplete, but none of them cared enough to add their own
copyright
notices, so this is the best we'll get.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

test/core/test-dbus-glib.c | 26 +++++
1 file changed, 26 insertions(+)

commit 309bce6d7965037bc6f1827715d2372ae19c98d5
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 13:23:29 +0100

Form a valid D-Bus error name if an unmapped error has a negative
code

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-gobject.c | 3 +-
1 file changed, 2 insertions(+), 1 deletion(-)

commit ffd32e448a76045696d81a82fec9aeac7dd5e64b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-28 12:20:16 +0100

If an error code is out of range for its domain, warn about it

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-gobject.c | 10 +++++++-
1 file changed, 9 insertions(+), 1 deletion(-)

commit a31371cb5852c485b85926125988c0a170a640ca
Author: David Woodhouse <dwmw2@infradead.org>
Date: 2011-09-28 12:16:01 +0100

Don't crash in `gerror_domaincode_to_dbus_error_name` if code is out of range

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=40151
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gobject.c | 3 +-
1 file changed, 2 insertions(+), 1 deletion(-)

commit 86ba4b0b8591b882803bela064df4c213349fba0
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-22 18:25:02 +0100

`dbus_g_value_parse_variant`: fix several `GVariant` ref leaks

`g_variant_get_child_value()` (and hence `g_variant_get_variant()`, which is a simple wrapper) returns a ref to the child; we were assuming it didn't.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=41125
Reviewed-by: Jonny Lamb <jonny.lamb@collabora.co.uk>

dbus/dbus-gvalue-parse-variant.c | 56
+++++-----
1 file changed, 38 insertions(+), 18 deletions(-)

commit 51222609f1af2eda0c223881dda3985435c3ad3f
Author: Colin Walters <walters@verbum.org>
Date: 2012-01-08 11:29:33 -0500

`test-service-glib`: Fix build with `Glib 2.30`

Only `<glib.h>` can be included directly.

test/core/test-service-glib.c | 1 -
1 file changed, 1 deletion(-)

commit 0215b0af4c95c9b9fc79ada5b99882f1261e2644
Author: Colin Walters <walters@verbum.org>
Date: 2012-01-08 11:28:10 -0500

autogen.sh: Honor NOCONFIGURE=1 environment variable

<http://people.gnome.org/~walters/docs/build-api.txt>

autogen.sh | 24 ++++++-----
1 file changed, 14 insertions(+), 10 deletions(-)

commit 6a8daa9569a4a6c2610758b122f46f181ce52448
Author: Derek Foreman <derek.foreman@collabora.co.uk>
Date: 2011-04-20 12:14:23 -0400

Support building on Android

Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=42532

.gitignore | 1 +
Android.mk | 35 ++++++
dbus/Makefile.am | 9 ++++++
3 files changed, 45 insertions(+)

commit dfdb0de78563a01c3d722e1de83d85e90163feaa
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-30 16:12:29 +0100

Micro version

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit f669a4aa6c7caaf6cbf9395ed14a1a357c4f6004
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-30 12:06:50 +0100

Prepare 0.98

configure.ac | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 1803b6cf35876247bb3e577bec9be8eed8b1b825
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-08-17 19:28:04 +0100

invoke_object_method: simplify how we return

Based on review feedback from Cosimo.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767

dbus/dbus-gobject.c | 5 +----

1 file changed, 1 insertion(+), 4 deletions(-)

commit 83bdd09f6124289dd19556da9334906f28064838
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-08-17 19:26:46 +0100

dbus-gobject: centralize death-by-OOM and use oom() instead of goto

You can't recover from OOM (in GLib-land) so there's no point in complicating the code for the reader.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767

dbus/dbus-gobject.c | 37 ++++++-----
1 file changed, 19 insertions(+), 18 deletions(-)

commit 705761ad983392c311c3e66c3ddf7e44e0efbc6f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 18:42:33 +0100

Fix various minor documentation bugs

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

dbus/dbus-glib.c | 2 +-
dbus/dbus-glib.h | 2 +-
dbus/dbus-gmain.c | 10 +++++-----
dbus/dbus-gobject.c | 33
+++++-----
dbus/dbus-gproxy.c | 33
+++++-----
dbus/dbus-gvalue.c | 2 +-
doc/reference/dbus-glib-sections.txt | 6 +++++-
7 files changed, 60 insertions(+), 28 deletions(-)

commit 0d058307cb914fa61e809b9e0500ac139e286ec6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 17:47:03 +0100

specialized types: improve documentation and document more things

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-gtype-specialized.c | 436
+++++-----
dbus/dbus-gtype-specialized.h | 6 +-
doc/reference/dbus-glib-sections.txt | 1 +
3 files changed, 388 insertions(+), 55 deletions(-)
```

commit b3af6fdb643f9293d156c95860270a997b9f6843
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 17:50:07 +0100

dbus_g_type_collection_value_iterate, etc.: check that the type
is suitable

Otherwise we'd probably crash when we cast the vtable to an
inappropriate
type, and call its methods with inappropriate arguments as a
result.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-gtype-specialized.c | 29 +++++-----
1 file changed, 25 insertions(+), 4 deletions(-)
```

commit 3183d79b119d0296924602f19d9fc2a58a7bd597
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 17:47:46 +0100

dbus_g_type_collection_get_fixed: check preconditions on the type
and vtable

Previously, if it wasn't a collection or didn't have the
fixed_accessor,
we'd just segfault. Not ideal.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-gtype-specialized.c | 14 +++++-----
1 file changed, 10 insertions(+), 4 deletions(-)
```

commit 7fc6f6c8c6d0e6ade0e8d5460b5311b564a24d3f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 17:41:36 +0100

dbus-gtype-specialized: warn if vtables have missing callbacks

It's just about conceivable that they wouldn't segfault, if

application
authors carefully avoided the unimplemented functionality... but
still.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

dbus/dbus-gtype-specialized.c | 19 ++++++
1 file changed, 18 insertions(+), 1 deletion(-)

commit 58ed26382bf23fe9fb592d38b5be2c2c7b2e77e2
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:17:10 +0100

Tidy up docs for DBusGError and its pseudo-methods

- * attach the introductory doc-comments to the enum and macro names,
not the internal function
- * remove docs for method parameters which didn't actually exist
- * more cross-references
- * make the domain macro visible in the documentation index

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

dbus/dbus-glib.c | 33
+++++-----
doc/reference/dbus-glib-sections.txt | 1 -
2 files changed, 18 insertions(+), 16 deletions(-)

commit 4b6482d5987095b474c9582e7415bf94a154d9b4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-27 16:25:21 +0100

Make gtk-doc DOC_SOURCE_DIR absolute, fixing out-of-tree docs

I used an absolute path to avoid differing behaviour with older
versions
of gtk-doc, where DOC_SOURCE_DIR was relative to the *source*
directory.

Also resync doc/reference/Makefile.am with gtk-doc's example.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

doc/reference/Makefile.am | 51

```
+++++-----
1 file changed, 38 insertions(+), 13 deletions(-)
```

```
commit 83471e869e464c63063e9e254bc8c40b25de03b7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:04:29 +0100
```

DBusGProxy: be more pedantic about boolean returns

Returning false doesn't strictly imply that the error is set; if you're being pedantic enough, it's really a tri-state where TRUE indicates success, FALSE indicates failure, and calling the function incorrectly results in undefined behaviour (currently a critical warning and returning FALSE).

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: NB#218973

```
dbus/dbus-gproxy.c | 10 +++++-----
1 file changed, 5 insertions(+), 5 deletions(-)
```

```
commit c6aal7dbdf04501495e4a3b4cfe6e7d808af6dd
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 15:57:44 +0100
```

DBusGProxy: misc documentation tidying

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-gproxy.c | 37
+++++-----
doc/reference/dbus-glib-sections.txt | 4 +++
2 files changed, 32 insertions(+), 9 deletions(-)
```

```
commit c568ace5ee6a9c13f748d876d69ad50234b2bc78
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 15:57:15 +0100
```

DBusGProxy: make argument names in declaration, definition and docs consistent

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793

Bug-NB: related to NB#218973

```
dbus/dbus-glib.h      |      8 +++----  
dbus/dbus-gproxy.c   |      59  
+++++-----  
2 files changed, 32 insertions(+), 35 deletions(-)
```

commit dc766e475f3a42872edbd9c3bcc98714dde8bff7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 15:55:47 +0100

DBusGProxy: mark struct contents as private

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-glib.h      |      2 ++  
1 file changed, 2 insertions(+)
```

commit 9d1cce7c64662733f0b48e52e4d5a1e805a3e246
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 15:40:40 +0100

Remove declaration of `dbus_pending_call_get_g_type`, which doesn't exist

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-glib-lowlevel.h      |      8 -----  
doc/reference/dbus-glib-sections.txt |      2 --  
2 files changed, 10 deletions(-)
```

commit 35e049459945a22b87cc61c05bd903790f05cf40
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:17:34 +0100

Add `DBUS_TYPE_CONNECTION`, `DBUS_TYPE_MESSAGE` to the docs, and document them better

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-glib.c          |     15 ++++++-----  
doc/reference/dbus-glib-sections.txt |      2 +-  
2 files changed, 13 insertions(+), 4 deletions(-)
```

2 files changed, 12 insertions(+), 5 deletions(-)

commit 43a4bf8b038c08d6b8c565495cbf69433bc2dcc6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:15:43 +0100

Tidy up docs for DBusGMessage

- * document the instance struct
- * make argument names match the headers
- * document the GType macro, not the function it calls, and mention it

in the documentation (since this is a boxed type, so it's non-obvious)

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-glib.c | 28 ++++++-----  
---  
doc/reference/dbus-glib-sections.txt | 2 +-  
2 files changed, 19 insertions(+), 11 deletions(-)
```

commit 53ac9ee390e173f8e63f276d8c7e211e4fadca4b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 16:34:38 +0100

Clean up docs for DBusGConnection

- * introductory doc-comment for the (pseudo-)class
- * attach the doc-comment for the GType to the macro, not the internal function
- * rename function parameters to match what the header file says

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

```
dbus/dbus-glib.c | 26 ++++++-----  
-  
doc/reference/dbus-glib-sections.txt | 2 +-  
2 files changed, 18 insertions(+), 10 deletions(-)
```

commit be937e037a6c076cf650bb79aba9c1138da39511
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 16:42:19 +0100

Ignore non-API source files correctly

gtk-doc only cares about the basename of the files.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

doc/reference/Makefile.am | 17 ++++++-----
1 file changed, 10 insertions(+), 7 deletions(-)

commit 053d41e56050706b512562f13b9ebde7dac3c60b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 16:42:44 +0100

Remove stray empty subsection

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

doc/reference/dbus-glib-sections.txt | 3 ---
1 file changed, 3 deletions(-)

commit 3d02f22d475286db8160604755ee85d24b133005
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:13:24 +0100

Mention which header to include in the gtk-doc

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

doc/reference/dbus-glib-sections.txt | 8 +++++++
1 file changed, 8 insertions(+)

commit 30b6a106ed590c0fd97a783998a19779002e65d8
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 16:13:07 +0100

Add some missing symbols to the docs

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37793
Bug-NB: related to NB#218973

doc/reference/dbus-glib-sections.txt | 6 +++++
1 file changed, 6 insertions(+)

commit a4bc263721fc5a80a78c6b551f35d281c22694ff
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-21 18:14:46 +0100

Micro version

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 80e37faeac62a4694514410ec17c2ab4a75d50e6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-09-21 17:36:11 +0100

dbus-glib 0.96

configure.ac | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 4bc172fd5f31123ee883c4027e729b38c2777d7d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 17:43:15 +0100

test-dbus-glib: make lose() abort, and make lose_gerror() more
informative

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

test/core/test-dbus-glib.c | 21 ++++-----
1 file changed, 4 insertions(+), 17 deletions(-)

commit 9e97f9744e7ffeb04a105087da6143931195ce77
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 16:25:15 +0100

get_all_object_properties: if _dbus_gvalue_marshal fails, bail out

This isn't necessarily OOM: _dbus_gvalue_marshal can fail due to
programming errors. If so, raise a critical warning, then (if
that wasn't
fatal) return a D-Bus error to be sent to the caller.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 27 ++++++-----
1 file changed, 25 insertions(+), 2 deletions(-)

commit bfd1c05174b86c9dfcflb22a4fa0b2c552dfc902
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 18:02:47 +0100

get_all_object_properties: note which operations can only fail via OOM

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 12 ++++++++
1 file changed, 12 insertions(+)

commit 30198a2368c9cd19db4b8f595bfad7a719ada7e9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 16:21:12 +0100

get_all_object_properties: skip invalidly-named properties, with a critical

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 9 ++++++
1 file changed, 9 insertions(+)

commit a2a3315ac9747dcc7dda860ceb1d1f94ab56d29b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-08-17 19:15:05 +0100

dbus-gobject: add and use connection_send_or_die to check for OOM

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Conflicts:

dbus/dbus-gobject.c

dbus/dbus-gobject.c | 40 ++++++++-----
-
1 file changed, 25 insertions(+), 15 deletions(-)

commit 22262581b0956f69f321b9e2eb702e3f15b49823
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:50:38 +0100

emit_signal_for_registration: assert that signal headers are all valid

This is an assertion, not a return_if_fail, because we already checked the object path in dbus_g_connection_register_g_object and the others in export_signals.

After these checks, dbus_message_new_signal can really only fail

on OOM.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 4 ++++
1 file changed, 4 insertions(+)

commit cccf1e6672fba6b649fe02082c2f087cdf022dcd
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:49:18 +0100

export_signals: check interface, signal names for validity

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 14 ++++++
1 file changed, 14 insertions(+)

commit d09dfcb130be0a17f337bd198a9fb43aab7dfa28
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:43:30 +0100

object_registration_message: check for OOM when a property is absent

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 8 ++++--
1 file changed, 5 insertions(+), 3 deletions(-)

commit 7b92ac1bfd1e50e842549dbb14b845478d114652
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:42:49 +0100

object_registration_message: make logic/assertions slightly clearer

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 13 +++++-----
1 file changed, 6 insertions(+), 7 deletions(-)

commit b5b27e4ca0d8ca250fa7649bcf2613e8d889a722
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:42:29 +0100

object_registration_message: handle remote caller errors better

If the remote process sends us a wrong message, that's its fault, not ours; we should send back a comprehensible D-Bus error, and not spam to stderr.

Previously, in each of these cases libdbus would have sent back NoReply, because we declined to handle the method call.

One case that's still wrong is passing extra arguments to Get, GetAll or Set, like so:

```
Get("com.example.Iface", "MyProperty", "extra")
Set("com.example.Iface", "MyProperty", Variant("foo"), "extra")
GetAll("com.example.Iface", "extra")
```

dbus-glib has historically warned, ignored the extra argument(s) and sent back a reply as if they hadn't been there, whereas a stricter implementation (like telepathy-glib's TpDBusPropertiesMixin) would have sent back an error reply and done nothing.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          20 ++++++++-----
1 file changed, 14 insertions(+), 6 deletions(-)
```

```
commit e5cf2be0e03af4df19618076015cc2a6556ca5d4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:36:09 +0100
```

get_object_property: add brief documentation

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          5 +++++
1 file changed, 5 insertions(+)
```

```
commit d3aa6c7942cce0cf7b602c6b0854488938f0548e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-05 15:25:55 +0100
```

get_object_property: unwind on errors, and avoid returning NULL

Also treat all errors here as programming errors (because this method should never fail), upgrading them from warning to critical; return a

D-Bus error reply anyway, to be nicer to our callers.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          33 ++++++-----  
1 file changed, 23 insertions(+), 10 deletions(-)
```

```
commit 98b3b3f323fd4001be0630c3e0a1dc4fb83cdeab  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-04-05 15:21:56 +0100
```

check_property_access: centralize error handling (and check for OOM)

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          69  
+++++-----  
1 file changed, 43 insertions(+), 26 deletions(-)
```

```
commit 6293849bf0469ce8d6d122b38db5091c21304a30  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-08-17 19:11:14 +0100
```

dbus-gobject: check for NULL when producing messages

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Conflicts:
 dbus/dbus-gobject.c

```
dbus/dbus-gobject.c |          6 +++--  
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
commit 8ec3166b6a40508a91144efedc10dcb32964498e  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-04-05 14:56:16 +0100
```

dbus-gobject: factor out reply_or_die, error_or_die and add error-checking

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35766
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          83  
+++++-----  
1 file changed, 50 insertions(+), 33 deletions(-)
```

```
commit 7a1abf5243fe5b7adf7211fd7d8c82851f448192  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
```


Date: 2011-04-04 16:52:58 +0100

invoke_object_method: if marshalling an out argument fails,
discard
message

We shouldn't report this as OOM, because it probably wasn't (it's
much
more likely to be programming error); we should also continue
through the
arguments, so that we don't leak them.

By abandoning the message as soon as we detect a programming
error,
we can use reply == NULL as an indicator of whether to keep
appending.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          47  
+++++-----  
1 file changed, 36 insertions(+), 11 deletions(-)
```

commit 4ba637e2ae32b2daf224265c22cc8b43cf0a0bbf
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-04 15:45:07 +0100

invoke_object_method, dbus_g_method_return_error: handle sending
failure

I just made it fatal, since it's either programming error or OOM.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          9 ++++++--  
1 file changed, 7 insertions(+), 2 deletions(-)
```

commit cd7e9a1c6dd0eef6e867cac80339a38758d0cd6f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-04 15:44:28 +0100

gerror_to_dbus_error_message: guarantee to return non-NULL

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
dbus/dbus-gobject.c |          6 ++++++  
1 file changed, 6 insertions(+)
```

commit c239a810543bde85dd458012612874cdc81a4151

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-04 15:41:32 +0100

invoke_object_method: if dbus_message_new_method_return fails,
fail hard

There's no point in doing graceful unwinding here, since it really
shouldn't happen.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit 480c480b024a77dd8f4efecdb6639d45de9767ce
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-04 15:40:40 +0100

arg_iterate: document

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35767
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 40
+++++
1 file changed, 40 insertions(+)

commit 36b55c3aac5de873a69c9bd0fbe003200a03ba32
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-16 15:33:01 +0100

dbus_g_proxy_manager_unregister: if GetNameOwner failed, don't
assert

got_name_owner_cb never adds the proxy to the unassociated_proxies
list
if there is a D-Bus error other than NameHasNoOwner.

Based on a patch from Janne Karhunen, for Maemo.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=38408
Bug-NB: NB#116862
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gproxy.c | 10 +++++----
1 file changed, 6 insertions(+), 4 deletions(-)

commit 18705c42ca6b6a583b64451055eec4dad3aaa3b4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-17 14:40:28 +0100

Add a regression test for fd.o #38406

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=38406

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
.gitignore          |    1 +
test/core/Makefile.am      |    7 ++
test/core/proxy-peer.c |   274
+++++
3 files changed, 282 insertions(+)
```

commit 5d800ca9352b40354a0f4a905c1c0b111b81f030

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2011-06-17 14:53:09 +0100

MyObject: register marshallers so individual tests don't have to

These marshallers are actually dual-use: they're used to call certain

methods on MyObject, and also to bind to its signals.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=38406

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

```
test/core/Makefile.am      |    4 +++-
test/core/my-object.c      |   12 ++++++++
test/core/my-object.h      |    1 +
test/core/test-dbus-glib.c |   11 +++-----
4 files changed, 20 insertions(+), 8 deletions(-)
```

commit 70e09208ab13812037b3cf133df312aacd24d036

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2011-06-17 12:18:57 +0100

DBusGProxy: deal with errors in DBUS_G_VALUE_ARRAY_COLLECT_ALL

G_VALUE_COLLECT can fail (admittedly, it's a programming error if it does). Previously, we leaked the resulting g_malloc'd string; now we emit a critical warning, free it, and abort the call.

Based on patches from Kimmo Härmäläinen and Christian Dywan in Maemo's dbus-glib.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=38406

Bug-NB: NB#86280

Bug-NB: NB#180486 (CID-34419 to CID-34424 inclusive)

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gproxy.c | 93

```
+++++-----  
1 file changed, 66 insertions(+), 27 deletions(-)
```

```
commit 2731a2f7f824f9c1cd11dd14e88ff1e0875f280b  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-06-17 12:15:20 +0100
```

```
dbus_g_proxy_call: simplify error handling now that call_id may be  
0
```

```
There's no need to special-case it any more - we can just use  
dbus_g_proxy_end_call_internal and rely on it to cope.
```

```
This reverts the fix for fd.o #12675, since the commit before this  
one  
is more general.
```

```
Bug: https://bugs.freedesktop.org/show\_bug.cgi?id=38406  
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
```

```
dbus/dbus-gproxy.c | 15 +++-----  
1 file changed, 3 insertions(+), 12 deletions(-)
```

```
commit 1b38cecf7bbdb8a99ef13616b8c661dfb0bc060f  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-06-17 12:06:11 +0100
```

```
DBusGProxy: cope gracefully with call_id == 0
```

```
As well as happening on a programming error or OOM (which both  
provoke  
critical warnings), dbus_g_proxy_begin_call_internal() can  
legitimately  
fail if we got disconnected from D-Bus (so can't send any more),  
but are  
still working through our backlog of incoming messages (so we  
haven't  
got round to the Disconnected message yet, so the DBusGProxy  
hasn't  
emitted ::destroy).
```

```
fd.o #12675 fixed dbus_g_proxy_call(), but  
dbus_g_proxy_call_with_timeout()  
and the asynchronous API still need similar treatment.
```

```
Bug: https://bugs.freedesktop.org/show\_bug.cgi?id=38406  
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
```

```
dbus/dbus-gproxy.c | 16 ++++++  
1 file changed, 16 insertions(+)
```

```
commit af31c9c21efec8a38c73de24d82a2038d80f74e
```

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-08-03 18:38:49 +0100

registrations test: fix race condition in adding match rules

If you don't block waiting for the reply to AddMatch, and you're using two connections to the bus, then messages sent from connection 2 might not be matched by connection 1, because they arrive before the AddMatch call from connection 1. This manifests itself as the /registrations/twice test *sometimes* failing (dependent on timing).

Only tests with two parallel bus connections need this, but to be safe, I've added it throughout this test.

Reviewed-by: Vivek Dasmohapatra <vivek@collabora.co.uk>

test/core/registrations.c | 28 ++++++-----
1 file changed, 25 insertions(+), 3 deletions(-)

commit 6d06dd358b22057dca1808c1432e620998202c41
Author: Tim Waugh <twaugh@redhat.com>
Date: 2011-07-22 15:13:42 +0100

dbus-binding-tool: fixed "pretty" mode output for methods.

Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=39475

dbus/dbus-glib-tool.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit e294ba3e579db1c12158ce95ad6ff34391cf571a
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-16 18:12:27 +0100

Show warning if marshalling in dbus_g_method_return fails

Based on a patch from Christian Dywan
<christian.dywan@lanedo.com>.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=29884
Bug-NB: NB#180486
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>

dbus/dbus-gobject.c | 8 ++++++--
1 file changed, 7 insertions(+), 1 deletion(-)

commit cce0887278ae105f010d35636fc5f6d654c2d41f
Author: Christian Dywan <christian.dywan@lanedo.com>
Date: 2010-12-14 15:07:27 +0100

remove minor dead code in dbus-gparser

There's no way retval can be false at this location.

Bug-NB: NB#180486 CID-643
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gparser.c | 3 ---
1 file changed, 3 deletions(-)

commit f446cb80f91b4cf584883b47477295d07dd32eac
Author: Derek Foreman <derek.foreman@collabora.co.uk>
Date: 2011-06-02 17:16:42 +0100

remove remaining inclusions of <glib/glib.h> from library

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36428
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-binding-tool-glib.c | 1 -
dbus/dbus-gproxy.c | 1 -
2 files changed, 2 deletions(-)

commit 2a85bd434e6d8dba9615a53d288a0a72d6b5e0cf
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-02 13:50:20 +0100

Regression test for fd.o #37852

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37852
Reviewed-by: Colin Walters <walters@verbum.org>

test/core/my-object.c | 17 ++++++++
test/core/my-object.h | 2 ++
test/core/registrations.c | 68
++++
test/core/test-service-glib.xml | 4 +++
4 files changed, 91 insertions(+)

commit 3e0828f57c3925ea9b63d22ab82d991a0fea0536
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-02 13:49:51 +0100

Fix regression in marshalling objects as object paths

This regressed while fixing fd.o #36811. NetworkManager apparently uses

this idiom.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37852
Bug-Debian: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=628890>
Tested-by: Michael Biebl <biebl@debian.org>
Reviewed-by: Colin Walters <walters@verbum.org>

dbus/dbus-gobject.c | 8 ++++---
1 file changed, 4 insertions(+), 4 deletions(-)

commit f534276eb20d8035566a206d1e38fd651a1eeab9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 17:03:59 +0100

registrations test: only listen for signals, not all messages

Otherwise, we'd reply with an error to messages not intended for us.

Reviewed-by: Colin Walters <walters@verbum.org>

test/core/registrations.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 867d670daad983f8b4416528e69d3ad2e0596e88
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 19:44:42 +0100

0.95

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 469df67525a9e5363644ac730ac8ec7580a8ebe0
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 19:07:30 +0100

Prepare 0.94 release

configure.ac | 7 ++++---
1 file changed, 4 insertions(+), 3 deletions(-)

commit 5a780af947f0141f4e0524aaf62fdd06abe6cd5f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-04 13:46:11 +0100

dbus-binding-tool: forbid ReturnVal annotation after the first OUT <arg>

It has never actually worked correctly (invoke_object_method would always

treat the ReturnValue as if it had been the first OUT argument),
so let's
only allow the situation that worked in practice.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=35952
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-binding-tool-glib.c | 15 ++++++
1 file changed, 15 insertions(+)

commit a8605cf4c6c9dbc4a933dd744d339744e0102765
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:33:48 +0100

Fix compiler warnings in dbus-binding-tool

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-glib-tool.c | 3 +-
1 file changed, 2 insertions(+), 1 deletion(-)

commit 0c162cc63b073a9f50dd11ea99db72fe95a16064
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:33:07 +0100

Remove all support for Doxygen

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=10890

Doxyfile.in | 180

Makefile.am | 2 -
README | 1 -
configure.ac | 34 +-----
doc/api/README | 3 -
5 files changed, 1 insertion(+), 219 deletions(-)

commit 6e20efb5c4c094915d2b6e86c3cab486a0562a95
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:32:58 +0100

Use single star in doc-comments not intended for gtk-doc

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=10890

dbus/dbus-glib.h | 4 +-
dbus/dbus-gloader-expat.c | 2 +-
dbus/dbus-gproxy.c | 2 +-
dbus/dbus-gtest.c | 2 +-
4 files changed, 5 insertions(+), 5 deletions(-)

commit 1472d195bd201d219aab5bc4b31d80d8340cee5c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:32:06 +0100

Remove Doxygen droppings from source code

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=10890

```
dbus/dbus-gidl.c      |      4 ----  
dbus/dbus-gidl.h     |      4 ----  
dbus/dbus-glib.c     |      3 +--  
dbus/dbus-gmain.c    |     22 +++-----  
dbus/dbus-gobject.c  |      3 +--  
dbus/dbus-gparser.c  |      4 ----  
dbus/dbus-gproxy.c   |     26 +------  
dbus/dbus-gthread.c  |      6 -----  
dbus/dbus-gutils.c   |      4 ----  
dbus/dbus-gutils.h   |      4 ----  
dbus/dbus-gvalue.c   |      5 +----  
11 files changed, 8 insertions(+), 77 deletions(-)
```

commit 4c3a346d296789cf3a81f6cc98f64d84ea892247
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:24:36 +0100

Remove `_dbus_g_proxy_test`, which isn't called and doesn't do anything

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37790

```
dbus/dbus-gproxy.c | 17 -----  
1 file changed, 17 deletions(-)
```

commit cded5996b02ff937a73654788d1f3a7ca46ca0a9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:24:29 +0100

Remove `_dbus_gutils_test`, which didn't test anything

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37790

```
dbus/dbus-gtest.c | 6 +-----  
dbus/dbus-gutils.c | 17 -----  
2 files changed, 1 insertion(+), 22 deletions(-)
```

commit d81722def291666a3c557a18d66be061f8f55e31
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-06-01 17:24:22 +0100

_dbus_gidl_test: remove unused function which does nothing

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37790

dbus/dbus-gidl.c | 17 -----
1 file changed, 17 deletions(-)

commit 1f371d3c75a5c2e9979350380e4243e3c445e093

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2011-06-01 17:23:45 +0100

dbus-glib-tool: remove regression-test boilerplate which does nothing

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37790

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-glib-tool.c | 48

1 file changed, 48 deletions(-)

commit 8abfa07b8acd2d8527785db3711d8a88a213a60e

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2011-06-01 17:18:52 +0100

dbus-binding-tool: clean up temporary file on error

This fixes distcheck, which otherwise fails because temporary files are left over after distclean. We could delete them with the build system, but leaving them behind is a bug, so...

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37812

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

dbus/dbus-glib-tool.c | 78
+++++
1 file changed, 58 insertions(+), 20 deletions(-)

commit d5b9aa101d61bcd2321f0266fa638b644e4cba96

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2011-05-31 14:35:01 +0100

Remove useless HAVE_GLIB conditional - GLib is mandatory

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37789

configure.ac | 1 -

1 file changed, 1 deletion(-)

commit 601b51279e7fbb08ff6ce8c98eaf4bcee21d46a5
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 14:40:12 +0100

Remove unused DBUS_DAEMONDIR

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37789

configure.ac | 10 -----
1 file changed, 10 deletions(-)

commit 1a966d070e71378590fb0b64c5ec7a2e250d397b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 14:39:57 +0100

Remove vestigial dbus-glib-error-*.h and the machinery to generate them

This was the only use of DBUS_INCLUDEDIR, so remove that too.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37789

.gitignore | 4 ----
configure.ac | 2 --
dbus/Makefile.am | 16 ++-----
dbus/make-dbus-glib-error-enum.sh | 2 --
dbus/make-dbus-glib-error-switch.sh | 2 --
5 files changed, 2 insertions(+), 24 deletions(-)

commit 98d7cb8c869b53686a742fb48d1f189c74ff1484
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-31 14:33:18 +0100

Simplify the check for dbus-1

As in commit 0f782b4, omitting the action-if-found and action-if-not-found results in much better error messages if this mandatory dependency isn't found.

Nothing checks HAVE_DBUS (as they shouldn't - it's mandatory!), so there's no point in keeping that either.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37789

```
configure.ac | 12 +++-----  
1 file changed, 3 insertions(+), 9 deletions(-)
```

```
commit b00d53dd647cf456ddb67c1dfdf44a8d56be90b4  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-01-04 18:35:41 +0000
```

Add more auto-generated gtkdoc droppings to .gitignore

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37789

```
.gitignore | 2 ++  
1 file changed, 2 insertions(+)
```

```
commit bd92f266609fef162b6028fd249250b0ab9ae689  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-05-31 14:46:33 +0100
```

Remove support for i18n completely

As far as I can tell, dbus-glib has never actually been translated into another language, or even had the necessary infrastructure so people could do that. The translated strings are all low-level errors in "programmer English" anyway, and most of them only make sense to D-Bus experts.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36428

```
configure.ac | 25 +-----  
dbus/dbus-binding-tool-glib.c | 8 +++---  
dbus/dbus-glib-tool.c | 35 ++++++++-----  
dbus/dbus-gparser.c | 56  
+++++++-----  
dbus/dbus-gproxy.c | 8 +++---  
dbus/dbus-gvalue.c | 41 ++++++++-----  
6 files changed, 68 insertions(+), 105 deletions(-)
```

```
commit c37bd4357d50d1cd683ef04f05ee427c9a0f7e9f  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-03-28 14:43:15 +0100
```

Remove i18n macros from files where they're unused

dbus-glib-tool (dbus-binding-tool) and dbus-gparser are theoretically localizable, but the library doesn't make any use of gettext.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36428

dbus/dbus-glib.c | 5 -----
dbus/dbus-gmain.c | 4 ----
2 files changed, 9 deletions(-)

commit e8a4f0496d0eec10fc9c76e0ef20e761bb2256c6
Author: Dan Williams <dcbw@redhat.com>
Date: 2011-05-31 11:25:59 -0500

Log error message from failed object registration

See https://bugzilla.redhat.com/show_bug.cgi?id=708686 for a case of this; it would be extremely helpful to figure out what the error actually was since it appears not to be a double-registration or other such issues after inspection of nm-applet and libnm-glib.

Reviewed-by: Colin Walters <walters@verbum.org>
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37795

dbus/dbus-gobject.c | 15 ++++++++-----
1 file changed, 10 insertions(+), 5 deletions(-)

commit eeb1c50aa03a95c39487e9331107142f27412958
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-20 16:41:28 +0100

Remind maintainers to update the libtool versioning

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=22854
Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

HACKING | 1 +
1 file changed, 1 insertion(+)

commit 9e2031148fd0ee114ec67c681d2b9bcfd84cdaa4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-20 16:41:00 +0100

add a Makefile rule to upload a tarball and up-to-date docs

Simplified from telepathy-glib, which has more elaborate machinery to automate releases.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=26952

Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

```
HACKING      |      3 ++-
Makefile.am  |     12 ++++++++
2 files changed, 14 insertions(+), 1 deletion(-)
```

commit 61f48c290176c007ff0aef6ca4d752491763e4b0
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-30 15:41:27 +0100

Force a full rebuild for new gcc versions if gcov is enabled

This avoids a lot of "Version mismatch - expected 406p got 405*". Because we're changing the content of each file after preprocessing, it even defeats ccache, which is desirable here - otherwise we'd get old versions of the profiling code being resurrected from the cache.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37060
Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

```
configure.ac |    17 ++++-----
1 file changed, 4 insertions(+), 13 deletions(-)
```

commit 0f320c5224fe39df44648ac75ecce9e0a01079d0
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-30 15:41:01 +0100

Include config.h in generated C files, and regenerate dbus-gmarshal.c

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37060
Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

```
dbus/Makefile.am      |      3 ++-
dbus/dbus-gmarshal.c |      7 +++++--
dbus/examples/statemachine/Makefile.am |      4 +++-
test/core/Makefile.am |      4 +++-
4 files changed, 13 insertions(+), 5 deletions(-)
```

commit f91828abb160e44b53a32823516d55d8a5d634bc
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-30 15:39:28 +0100

Consistently include config.h in all non-generated .c files

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37060

Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

```
dbus/dbus-gidl.c | 2 ++
dbus/dbus-gloader-expat.c | 2 ++
dbus/dbus-gparser.c | 3 +++
dbus/dbus-gproxy.c | 3 +++
dbus/dbus-gthread.c | 2 ++
dbus/dbus-gtype-specialized.c | 2 ++
dbus/dbus-gvalue-parse-variant.c | 2 ++
dbus/examples/example-client.c | 2 ++
dbus/examples/example-service.c | 2 ++
dbus/examples/example-signal-emitter.c | 2 ++
dbus/examples/example-signal-recipient.c | 2 ++
dbus/examples/statemachine/statemachine-client.c | 2 ++
dbus/examples/statemachine/statemachine-server.c | 2 ++
dbus/examples/statemachine/statemachine.c | 2 ++
test/core/30574.c | 2 ++
test/core/peer-client.c | 2 ++
test/core/peer-server.c | 2 ++
test/core/test-dbus-glib.c | 2 ++
test/core/test-gvariant.c | 2 ++
test/core/test-service-glib.c | 2 ++
test/core/test-thread-client.c | 2 ++
test/core/test-thread-server.c | 2 ++
test/core/test-types.c | 2 ++
test/core/test-variant-recursion.c | 2 ++
test/interfaces/test-client.c | 2 ++
test/interfaces/test-objects.c | 2 ++
test/interfaces/test-server.c | 2 ++
test/test-service.c | 2 ++
28 files changed, 58 insertions(+)
```

commit 6d6d9a61e47f5bc28b5d61ae0d6cb072a81b5569
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-13 19:00:41 +0100

dbus_g_proxy_add_signal: document error cases

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36216
Reviewed-by: Guillaume Desmottes
<guillaume.desmottes@collabora.co.uk>

```
dbus/dbus-gproxy.c | 5 ++++
1 file changed, 4 insertions(+), 1 deletion(-)
```

commit 52f4e09d49551f3a8ece6578a6cb6e016c03a9d8
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-10 12:00:15 +0100

dbus-gobject: check various preconditions for methods

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
dbus/dbus-gobject.c |          29 ++++++-----  
1 file changed, 26 insertions(+), 3 deletions(-)
```

commit 6a61c922cedad9f5af5c32a0d783e9fbd53f3487
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 12:52:25 +0100

dbus-gobject: move weakref to ObjectExport, with
ObjectRegistration
pointing there

This has the following results:

- * each exported object only needs one weak ref to itself, however many times it is registered
- * because the ObjectRegistration now points to the ObjectExport, it can always remove itself from the list of registrations even if the actual object has been disposed, avoiding a leak (fd.o #36811)

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
dbus/dbus-gobject.c |          89  
+++++-----  
1 file changed, 59 insertions(+), 30 deletions(-)
```

commit f07cf89fd36aeb5bd32f9e7cff13b5697365917
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 12:25:18 +0100

Add a regression test for removing all registrations and starting again

This has been confirmed to fail when the previous commit is reverted.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
test/core/registrations.c |    64  
+++++-----  
1 file changed, 64 insertions(+)
```

commit c3c6f8365f5074f3f59b66c5b4ecelfeb214be23
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 12:24:35 +0100

dbus_g_connection_register_g_object: only hook onto signals on first use

This fixes a bug in which an empty list of registrations was considered to be synonymous with the object never having been exported, resulting in this failure mode:

- * register object at n locations
 - the first time, export_signals() is called
- * unregister all of those locations
- * register object at a new location
 - export_signals() is wrongly called again
- * emit a signal
 - the D-Bus signal gets emitted twice

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
dbus/dbus-gobject.c |          48  
+++++-----  
1 file changed, 16 insertions(+), 32 deletions(-)
```

commit 323d5adb48f8630024f2b22172f10684508bd2a8
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 10:47:56 +0100

dbus_g_connection_register_g_object: attach ObjectExport first

This means we no longer need the unnecessarily subtle "steal and put back" pattern.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
dbus/dbus-gobject.c |          20 +++++-----  
1 file changed, 9 insertions(+), 11 deletions(-)
```

commit c9961279c5ac667d4ddca98cbd3a3dfb4f1d8705
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 10:45:58 +0100

dbus-gobject: wrap the GSList of registrations in a simple struct

For simplicity, the ObjectExport struct isn't freed until the object is actually destroyed, even if there are no more registrations.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36811

```
dbus/dbus-gobject.c |          77
+++++-----
1 file changed, 51 insertions(+), 26 deletions(-)
```

```
commit 357fef2267d4243435e0d26ae659d21bd6bcde96
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-01-04 16:12:03 +0000
```

test/interfaces: test annotated arguments

The D-Bus Specification doesn't actually allow this, but dbus-glib has always supported it, and indeed understands a couple of <arg> annotations itself - org.freedesktop.DBus.GLib.ReturnVal and org.freedesktop.DBus.GLib.Const.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
test/interfaces/valid-annotations.xml | 10 ++++++--
1 file changed, 8 insertions(+), 2 deletions(-)
```

```
commit b04f6b33224973cefec959a9fe0f7de655b93d0e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-01-04 16:07:39 +0000
```

test/interfaces: check that some invalid annotations aren't allowed

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
.gitignore | 2 ++
test/interfaces/Makefile.am | 7 +++++-
test/interfaces/invalid-annotated-node.xml | 8 ++++++
test/interfaces/invalid-nested-annotation.xml | 8 ++++++
test/interfaces/run-test.sh | 20
+++++-----
5 files changed, 43 insertions(+), 2 deletions(-)
```

```
commit e4a25a148703ef7ff13e5d87276b5d6fc395a57d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-01-04 16:01:39 +0000
```

Test that interfaces, methods, properties, signals can be annotated

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
test/interfaces/Makefile.am | 20 ++++++++-----  
test/interfaces/valid-annotations.xml | 21 ++++++++-----  
2 files changed, 39 insertions(+), 2 deletions(-)
```

commit ac53fc025df08f12532fa7e3e69208db16151935
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-01-04 15:51:05 +0000

Don't .gitignore everything in test/interfaces, and remove some duplication

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
.gitignore | 19 +++++-----  
1 file changed, 6 insertions(+), 13 deletions(-)
```

commit b1338486f0ce5df460e9af67f3738f6a0f829840
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-01-04 15:45:57 +0000

Allow <annotation> inside <property> too

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
dbus/dbus-gparser.c | 4 ++--  
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 160842c85f808a4815d005dd39a0d9419aa06788
Author: Christian Dywan <christian@lanedo.com>
Date: 2011-01-04 15:43:05 +0000

Allow signal annotations in dbus-binding-tool

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27598
Bug-NB: NB#163566
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Robert Ancell <robert.ancell@canonical.com>

```
dbus/dbus-gparser.c | 1 -  
1 file changed, 1 deletion(-)
```

commit dcbebd2ee2941136a2e04868da08777c985e0117
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 18:49:19 +0100

Add a regression test for removing an object from only one connection

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32087

test/core/registrations.c | 9 ++++----
1 file changed, 5 insertions(+), 4 deletions(-)

commit 818b90092102e4fe6d2addfe77ee5a86d50e31c5
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 18:49:06 +0100

dbus_g_connection_unregister_g_object: only unregister from
@connection

Previously, we unregistered the object from all connections, if it
was
present on more than one.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32087

dbus/dbus-gobject.c | 8 +++++--
1 file changed, 6 insertions(+), 2 deletions(-)

commit 6b29f7f317df7d96a94ac910306508452be9130c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 18:43:29 +0100

Add a regression test for registering an object on two connections

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32087

test/core/registrations.c | 101
++++
1 file changed, 101 insertions(+)

commit 17d5dcf86c766075400aff5d2aa3fe8e599fb715
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 18:19:44 +0100

fd.o #32087: emit signals on the appropriate connections

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32087

dbus/dbus-gobject.c | 23 +++++-----
1 file changed, 5 insertions(+), 18 deletions(-)

commit 7f041871c3e5d023ee9cb77609bf0b69ebdbde34
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 16:06:25 +0100

Add a regression test for re-registering an object at the same
path

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793

test/core/registrations.c | 24 ++++++
1 file changed, 24 insertions(+)

commit 43f9dda46e9279b4489b6bd1efa0e5f9eb825a48
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 16:04:30 +0100

dbus_g_connection_register_g_object: don't destroy state on early return

At the beginning of the function we steal the object's state so we can add to the list. After doing that, we must make sure we give it back!

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Reviewed-by: Sjoerd Simons <sjoerd.simons@collabora.co.uk>

dbus/dbus-gobject.c | 12 ++++++----
1 file changed, 8 insertions(+), 4 deletions(-)

commit 4da28a6c49cecb902544d9152a693105d320cdac
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 11:59:44 +0100

dbus_g_connection_unregister_g_object: fix out-of-bounds reading

The list of registrations is singly linked; we only avoid a crash here by luck.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793
Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Reviewed-by: Sjoerd Simons <sjoerd.simons@collabora.co.uk>

dbus/dbus-gobject.c | 6 +++--
1 file changed, 3 insertions(+), 3 deletions(-)

commit da740bfd88850c858dlcee079c8d261a79e2c75e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 15:25:08 +0100

Subsume the test for #5688 into the more general registrations test

Also test that the object is unregistered by the last unref or by forced disposal, without crashing.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793

```
test/core/5688.c          | 81
-----
test/core/Makefile.am     | 9 ----
test/core/registrations.c | 73
+++++
test/core/run-test.sh     | 1 -
4 files changed, 67 insertions(+), 97 deletions(-)
```

commit 0692c5995ea4a7013b4ab605cd76aca49239168d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 15:08:27 +0100

Convert registrations test to use GTest

Also use a private bus connection, for potentially better leak-detection.

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793

```
test/core/registrations.c | 74
+++++
1 file changed, 53 insertions(+), 21 deletions(-)
```

commit 002feb57cfcfeb439belace875a67ad540fed950
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-03 14:49:29 +0100

Rename test/core/unregister to registrations (no source changes)

Reviewed-by: Cosimo Alfarano <cosimo.alfarano@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36793

```
.gitignore                | 2 +-
test/core/Makefile.am     | 6 ++--
test/core/registrations.c | 71
+++++
test/core/run-test.sh     | 2 +-
test/core/unregister.c    | 71
-----
5 files changed, 76 insertions(+), 76 deletions(-)
```

commit 15d9f31812b07439558e615d4d9bdfccea33dbd5
Author: Mike Gorse <mgorse@novell.com>
Date: 2011-03-11 09:26:01 -0600

Fix removal of old io and timeout handlers when switching GMainContexts

Bug: http://bugs.freedesktop.org/show_bug.cgi?id=35115
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gmain.c | 16 ++++-----
1 file changed, 5 insertions(+), 11 deletions(-)

commit [ala9a6f973f64e543c3d43b01bd6ef0963f8bf16](#)
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-10 11:17:53 +0100

test-service: ignore result of dbus_bus_request_name since we look
at the error instead

gcc 4.6 warns for this usage.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

test/test-service.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)

commit [6c1e9559094be37bf66731b9f09fffcbl1a19e8e5](#)
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-10 11:16:45 +0100

marshal_collection_array: statically assert that size conversion
is
not needed

Previously, the function blindly assumed this without a check.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

dbus/dbus-gvalue.c | 10 ++++++++
1 file changed, 10 insertions(+)

commit [324122a9f0bc859bb424d9026019e9077cda4015](#)
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-10 11:15:32 +0100

marshal_collection_array: remove set-but-unused variable elt_size

gcc 4.6 warns about these.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

dbus/dbus-gvalue.c | 3 ---
1 file changed, 3 deletions(-)

commit 341043e2d776cbd6faaf5b8d6a54a46920bb094c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 15:18:48 +0100

marshal_map, marshal_struct: remove unused variable ret

gcc 4.6 warns about these.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

dbus/dbus-gvalue.c | 6 -----
1 file changed, 6 deletions(-)

commit 1ea2eee4623cace2e852f1c16b1b1ce23035d037
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 15:18:18 +0100

demarshal_static_variant: remove useless call to
dbus_message_iter_get_arg_type

gcc 4.6 warns about this.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

dbus/dbus-gvalue.c | 2 --
1 file changed, 2 deletions(-)

commit f8a7ccf024a19caa54d6f79cd9d62a63a9b36c5b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-05-04 15:17:51 +0100

bash completion helper: remove unused variable 'prev', and
simplify

gcc 4.6 warns about this.

Reviewed-by: Colin Walters <walters@verbum.org>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=37062

dbus/dbus-bash-completion-helper.c | 8 +-----
1 file changed, 1 insertion(+), 7 deletions(-)

commit 09d85a945760d06447aea09a38ea99d0dc54de2a
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 17:05:04 +0100

DBusGProxy: link against GIO and use GDBus to check names' syntax

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=23616


```
configure.ac          | 2 +-
dbus/dbus-gproxy.c   | 29 ++++++-----
2 files changed, 21 insertions(+), 10 deletions(-)
```

```
commit f2b445d0d931f3f680cefe0141e62abeafda9742
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 15:08:36 +0100
```

dbus_g_proxy_end_call: check that it's a proxy

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=23616

```
dbus/dbus-gproxy.c   | 2 ++
1 file changed, 2 insertions(+)
```

```
commit 811089dae49c2f47d0f15fd7e9c180674f484391
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 15:08:05 +0100
```

dbus_g_proxy_set_interface: check that it's a proxy and not destroyed

If it has emitted destroy, our use of priv->manager will be a NULL pointer dereference.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=23616

```
dbus/dbus-gproxy.c   | 5 +++++
1 file changed, 5 insertions(+)
```

```
commit af3d48b10042616ac7ecceed5af8ddaeccc093f5
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 15:07:23 +0100
```

Document that most DBusGProxy methods stop working on ::destroy

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=23616

```
dbus/dbus-gproxy.c   | 42 ++++++
1 file changed, 42 insertions(+)
```

```
commit 63bd85823fd3039e88176105434c7bbdeae71e84
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 15:06:18 +0100
```

dbus_g_proxy_new_from_proxy: check that the old proxy is in fact a proxy

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=23616

dbus/dbus-gproxy.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 97de5501c2ebb29e4e32822bae0e45c88140bffd
Author: Alban Crequy <alban.crequy@collabora.co.uk>
Date: 2011-04-14 14:48:56 +0100

DBusGProxy: remove duplicate owner match rules.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=33646
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gproxy.c | 109
+++++-----
1 file changed, 82 insertions(+), 27 deletions(-)

commit bc0497ea0c5ac65abc3658a31c95f9b0e15680f3
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-04-13 19:00:23 +0100

dbus_g_proxy_add_signal: stop falsely claiming that we read introspection

If we believed the introspection, services could change their introspection and remote-crash us, so it's good that we don't. We shouldn't claim that we do, though.

The second sentence is subtle: for D-Bus types that dbus-glib can map

into more than one GLib type, you must currently use the one that dbus-glib would "naturally" produce. The only example I can find is that

object paths must be DBUS_TYPE_G_OBJECT_PATH, even though dbus-glib

can also (in principle) unmarshal object-paths as DBUS_TYPE_G_PROXY.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=36216

dbus/dbus-gproxy.c | 6 +++--
1 file changed, 3 insertions(+), 3 deletions(-)

commit f5008cc76e0b466c4730e331857bc041aced5e94
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 18:52:27 +0100

improve documentation of proxy timeouts to mention that -1 is special

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 10 ++++++----
1 file changed, 6 insertions(+), 4 deletions(-)

commit 25c4b24a0d8490a7ba667b9944ce9b1c0808878a
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 18:52:02 +0100

Don't allow proxy timeouts to be set negative, except for -1

libdbus checks this as a precondition, but to avoid astonishing
error
behaviour (interpreted as OOM), so should we.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 3 +++
1 file changed, 3 insertions(+)

commit 35d1d8479031393f86fe668d2a1f64325b5dd1be
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:35:51 +0100

DBusGProxy: share OOM handler

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 19 ++++++++-----
1 file changed, 10 insertions(+), 9 deletions(-)

commit 3a0e477ed6648bcb99711976fd51f4c0394000c6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:32:11 +0100

dbus_g_proxy_call_no_reply: don't assume NULL message means OOM

It might either be OOM or bad arguments, probably the latter.
That's
programming error, but we already raised a critical warning, so
be silent.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit d0332f9862c3af18161f58444f35c7d03d9fb067

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:32:00 +0100

dbus_g_proxy_begin_call_internal: don't assume NULL message means OOM

It might either be OOM or bad arguments, probably the latter. That's programming error, but we already raised a critical warning, so be silent.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit cd0da3f56873c5399b6247791df16acd3f4a73b1
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:30:29 +0100

dbus_g_proxy_marshall_args_to_message: diagnose inability to marshal args

Failure to marshal arguments is entirely reachable, albeit only via programming error.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gproxy.c | 16 ++++++++-----
1 file changed, 12 insertions(+), 4 deletions(-)

commit 8b75efc7941ac789605128d48f82168da18bb9fd
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:27:18 +0100

marshal_collection_array: simplify exit path

Also don't assume that failure of dbus_message_iter_close_container is always OOM.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 7 +-----
1 file changed, 1 insertion(+), 6 deletions(-)

commit 7729c5a4ee9c1bba0fe3f35d840c542303e26a1a

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:26:56 +0100

marshal_collection_array: if appending fails, don't assume OOM

Appending can also fail by appending ((gboolean) 23) or something;
admittedly, that's a programming error.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 9 ++++++--
1 file changed, 8 insertions(+), 1 deletion(-)

commit 73c1dc8031df7c9de0ab321f02cc5bedc49be7c1
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:25:50 +0100

marshal_collection_array: assert validity of signature

This means we can recognise OOM reliably; do so.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit ca88a4f4adffccc1be8ddc0835d640f36d42f9fc
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:25:25 +0100

marshal_collection_array: fail early if the array is NULL

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 7 ++++--
1 file changed, 4 insertions(+), 3 deletions(-)

commit 7228321ddaafc88e12406ca49406dfaa80240be9d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:24:53 +0100

marshal_collection_ptrarray: abandon broken containers, simplify
exit path

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 14 ++++++-----
1 file changed, 7 insertions(+), 7 deletions(-)

commit 9e8bed2ec229e48df0b1c63560b183f46d33bffd
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:24:18 +0100

marshal_collection_ptrarray: assert that the signature is valid

This means that dbus_message_iter_open_container can't fail except
by OOM,
so use the shared handler.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit 385180d3b1975afce823d4aa9aa5daf82fa09b5a
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:23:46 +0100

marshal_variant: abandon broken containers rather than closing
them

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)

commit 72c21e4e4cac8092c1dcf8348fdf3955cb01229c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:23:34 +0100

marshal_struct: simplify teardown, and abandon broken containers

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 15 ++++++-----
1 file changed, 8 insertions(+), 7 deletions(-)

commit 48cf1b8b147d80265e2ab23b257d5c96ca260db9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:23:08 +0100

marshal_map: on error, abandon the container instead of closing it

libdbus considers it to be a programming error if you close a
container
gracefully when its contents are incomplete.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 11 ++++++---
1 file changed, 9 insertions(+), 2 deletions(-)

commit c827cd5b160b59246305526eac6154006f8bb8d5
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:21:18 +0100

marshal_signature: check validity of the signature

Also remove a spurious cast, and comment why we're **not** assuming that failure here is necessarily OOM.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)

commit 98e5da6a6363b68042503e8e01819dc86ec7781e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:20:51 +0100

marshal_object: check that it's really an object and its path is valid

Again, we can now assume that failure is OOM.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 10 +++++-----
1 file changed, 5 insertions(+), 5 deletions(-)

commit 5d90b996ad4e9ee927d25fc692c8b08baa2a36b4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:20:23 +0100

marshal_object_path: check that it's really a valid object path

Again, this means failure can only be OOM, so use the shared handler.

Also remove an unnecessary cast.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 8 ++++---
1 file changed, 5 insertions(+), 3 deletions(-)

commit f57e4e33838062e98d39764202fdcd60b709362d
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:19:45 +0100

marshal_proxy: check that it's really a proxy and its path is valid

This means that dbus_message_iter_append_basic can't fail other than by OOM, so use the shared OOM handler.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 7 ++++--
1 file changed, 5 insertions(+), 2 deletions(-)

commit 15bf1a2ef4d8b3e37b056cc893a844706fc714dc
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:18:43 +0100

marshal_valuearray: simplify teardown

If close_container fails, it might be an assertion failure in libdbus, so don't assume it's OOM. We have to handle programming errors somewhat gracefully during marshalling anyway, so there's no point in asserting.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 8 +-----
1 file changed, 1 insertion(+), 7 deletions(-)

commit d1093f2c3413285c839cdbbca0c848b0d071cb62
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:16:29 +0100

marshal_valuearray: abandon the container if we fail

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 5 ++++
1 file changed, 4 insertions(+), 1 deletion(-)

commit 782412b65a1364f8736e15217aa853083c962935
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:15:27 +0100

marshal_basic: if marshalling a string fails, critical and return FALSE

This could be OOM, but it could also be non-UTF-8.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 18 ++++++++-----
1 file changed, 12 insertions(+), 6 deletions(-)

commit a2c276a49fa9efb8336db74db60dclbcd2f0e8d6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:14:25 +0100

marshal_basic: if a boolean has a non-boolean value, diagnose the error

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)

commit 94a501e0e69cf912c7f6a5250fdea0ec5fe673c7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:13:50 +0100

dbus-gvalue: replace trivial cases of OOM with a call to a noreturn function

In these cases there's no invalid value we could be supplying to libdbus, so it's definitely OOM, not a programming error (e.g. non-UTF-8) masquerading as OOM.

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

dbus/dbus-gvalue.c | 31 ++++++++-----
1 file changed, 19 insertions(+), 12 deletions(-)

commit 41b2fd28a833995c8ccab1a690c0e3925572488b
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 17:54:12 +0100

When given an object path, use GVariant to check syntactic validity

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

```
dbus/dbus-gobject.c |      2 +-  
dbus/dbus-gproxy.c |      6 +++---  
2 files changed, 4 insertions(+), 4 deletions(-)
```

commit 367a58bc081d451ff90262ae96b7d78ea74f3d11
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-28 19:05:44 +0100

Depend on dbus 1.2.16 for dbus_message_iter_abandon_container

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30171

```
configure.ac |      2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 0f782b478a784ac890741e5d84fd741a26d49510
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2011-03-29 14:45:34 +0100

Stop overriding pkg-config's nice informative message if GLib is too old

This means that instead of this rather ambiguous text:

```
checking for DBUS_GLIB... no  
checking for DBUS_GLIB_THREADS... yes  
configure: error: GLib development libraries not found
```

platforms whose GLib is too old will get the conventional output:

```
checking for DBUS_GLIB... no  
configure: error: Package requirements (gobject-2.0 >= 2.26)  
were not met:
```

Requested 'gobject-2.0 >= 2.26' but version of GObject is 2.24.0

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=33145
Reviewed-by: Colin Walters <walters@verbum.org>

```
configure.ac |      6 +-----  
1 file changed, 1 insertion(+), 5 deletions(-)
```

commit ab2a98bd3ae0c59f198e82cba2c81095a47fa58d
Author: Robert Ancell <robert.ancell@canonical.com>
Date: 2011-02-15 14:10:34 +1100

Fix linking order

Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=34282

```
dbus/Makefile.am | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit af0c64037673d598a6842fb344b5f3dad80ab821  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-06-02 16:31:54 +0100
```

docs: correct links in introduction

Fixes https://bugs.freedesktop.org/show_bug.cgi?id=22667 â€”
thanks to

Daniel Macks for the bug report!

```
doc/reference/dbus-glib-docs.sgml | 13 ++++++-----  
1 file changed, 8 insertions(+), 5 deletions(-)
```

```
commit 04e11da48a6c2a75bebb5ece31cf86508203dd02  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-01-04 16:33:27 +0000
```

Add a regression test for errors containing underscores

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30274
Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
test/core/my-object.c | 15 ++++++  
test/core/my-object.h | 3 +-  
test/core/test-dbus-glib.c | 11 ++++++  
test/core/test-service-glib.xml | 3 +++  
4 files changed, 31 insertions(+), 1 deletion(-)
```

```
commit 2d987890c3ba2fd8efc92f8b75da88904ce425f9  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2011-01-04 18:04:11 +0000
```

Run `run-with-tmp-session-bus.sh` directly rather than via `${SHELL}`

It has a correct `#!/bin/sh` line and is executable, so just running it is fine. This avoids running it under a user's chosen interactive shell,

which might be non-POSIX (zsh, csh). Gentoo Portage apparently sets

`SHELL` to `/bin/false`, presumably to trap this sort of thing.

(I've tested this on Debian unstable with `/bin/sh -> /bin/dash`, to check that weren't accidentally relying on `$SHELL` being bash or similar.)

Based on a workaround from Myckel Habets on the Gentoo bug.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=27193
Bug-Gentoo: https://bugs.gentoo.org/show_bug.cgi?id=254192
Reviewed-by: Colin Walters <walters@verbum.org>

```
test/core/run-test.sh          |          2 +-  
test/interfaces/run-test.sh |          2 +-  
2 files changed, 2 insertions(+), 2 deletions(-)
```

commit 6708c68cc9c312ffea600d0020425412ba242ddb
Author: Christian Dywan <christian@lanedo.com>
Date: 2011-01-04 18:48:31 +0000

Cleanup indentation in dbus-glib-binding-tool

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32351
Bug-NB: NB#180486
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
dbus/dbus-binding-tool-glib.c | 134  
+++++-----  
1 file changed, 67 insertions(+), 67 deletions(-)
```

commit flb8944a93a22d342f744526f5c47e80e394579f
Author: Christian Dywan <christian@lanedo.com>
Date: 2011-01-04 18:48:27 +0000

Plug leaks in dbus-glib-binding-tool

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=32351
Bug-NB: NB#180486
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
dbus/dbus-binding-tool-glib.c | 27 +++++-----  
1 file changed, 13 insertions(+), 14 deletions(-)
```

commit 4cd1f54f4db5759db3781da4645cebcd4143e9a4
Author: Christian Dywan <christian@twotoasts.de>
Date: 2010-09-20 11:57:08 +0200

Do not squash underscores in error names

Underscores are used and valid according to the specification.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30274
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
dbus/dbus-gobject.c |          9 +++++----  
1 file changed, 5 insertions(+), 4 deletions(-)
```

commit d4780b527f5112acf015c0e57c9911e939b16f99

Author: Christian Dywan <christian.dywan@lanedo.com>
Date: 2010-07-21 16:32:22 +0200

Watch server_set_data failure in dbus__server_setup_with_g_main

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=29884
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

dbus/dbus-gmain.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit c7b83e5b3bf7291b75399eed3f01ecea0c98702f
Author: Christian Dywan <christian.dywan@lanedo.com>
Date: 2010-12-13 10:14:57 -0800

Verify Increment call in dbus-glib-test

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=29884
Origin: vendor, Maemo
Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

test/core/test-dbus-glib.c | 3 +++
1 file changed, 3 insertions(+)

commit 7e157ee379de430fce9874291701c0c6666d2c2d
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2010-11-10 10:44:38 +0000

Bump version to 0.93 for development

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit a12d06317591f954542d5ddbela1d3cad10a8156
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2010-11-10 10:25:19 +0000

Release version 0.92

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit cb7a627ea838ae453d1401679a2b97acd95ba327
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2010-11-10 10:22:40 +0000

Bump GLib dependency to 2.26.

G_VARIANT_TYPE_STRING_ARRAY, as used by the GVariant<->GValue
conversion
functions, was added in 2.26. The previous release should have
had this

dependency, but I didn't notice.

```
configure.ac |    2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit c12efeeb7d9eb76eeae049528cee09c331551a5e  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-11-08 00:20:20 -0500
```

Bump version to 0.91 for development

```
configure.ac |    2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 4d4a1e266727de9ed1cba2707006435a389cac90  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-11-08 00:16:46 -0500
```

HACKING: Fix release URL; be honest about NEWS

```
HACKING |    8 ++++----  
1 file changed, 4 insertions(+), 4 deletions(-)
```

```
commit b3c18c486dbb1fd2c6221d26fd64719d73ff4200  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-11-08 00:12:00 -0500
```

Makefile.am: Remove ChangeLog's spurious FORCE dependency

This seems to be some kind of leftover from the now-deleted lcov stuff.

```
Makefile.am |    2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit ccd31bb39e496ba511e888d793b15bfaa22894f0  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-11-08 00:08:15 -0500
```

Release version 0.90

```
configure.ac |    2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 65fb5802295ff28970acle3f0d5adaa6927a5773  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2010-11-08 00:04:31 -0500
```

Build test/ before its subdirs.

This fixes running `make check` without first running `make`, since

subdirs of test/ depend on the binaries in that directory.

```
test/Makefile.am |      2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 22df79e0c1373b696adbbaf2791195f0a8b5fb0f  
Author: Mike Gorse <mgorse@novell.com>  
Date: 2010-10-22 04:33:53 -0400
```

Fix switching a connection's GMainContext

```
dbus_connection_setup_with_g_main and  
dbus_server_setup_with_g_main  
are  
intended to switch the setup to a new main loop context if they  
are  
called with a context other than the one with which they were  
initially  
called. However, this does not work in practice. The problem has  
to do  
with libdbus maintaining a list of watches associated with the  
connection. When dbus_connection_set_watch_functions is called,  
it  
first adds the existing watches using the new function and data,  
then  
removes the watches using the old function and data. Remove_watch  
calls  
connection_setup_remove_watch, and the latter does not check that  
the  
watch's connection setup matches the connection setup passed to  
it, so  
it winds up removing the watch from the new connection setup.  
(The  
watch has already been removed from the old setup at this point,  
as a  
side effect of dbus_watch_set_data).
```

Fixes: <https://bugs.freedesktop.org/show_bug.cgi?id=30574>
Signed-off-by: Will Thompson <will.thompson@collabora.co.uk>

```
.gitignore          |      1 +  
dbus/dbus-gmain.c   |      7 ++--  
test/core/30574.c   |     96  
+++++  
test/core/Makefile.am |      4 +++  
test/core/run-test.sh |      1 +  
5 files changed, 106 insertions(+), 3 deletions(-)
```

```
commit 8f0936de52b62b025385182406e5ea5795503d76  
Merge: 8aca055 afd0987  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2010-10-14 13:06:47 +0100
```

Merge branch 'type-names'

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30431

commit 8aca05563f86bd24fedb4322363ae844affc3c61

Merge: 0ab27bb ad4b66d

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2010-10-14 12:37:58 +0100

Merge branch 'gvariant'

Reviewed-by: Danielle Madeley <danielle.madeley@collabora.co.uk>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=30428

commit ad4b66db0e3a6ecab31bed214f341c4b07691224

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2010-10-14 12:15:08 +0100

copy arrays of 'o', 'g' from GVariant without constructing a
format

string programmatically

dbus/dbus-gvalue-parse-variant.c | 5 ++---

1 file changed, 2 insertions(+), 3 deletions(-)

commit 869c180e12c68ed06fe973c58f26cffbf4a3cec9

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2010-10-14 12:13:19 +0100

output unhandled GVariantClass as ASCII if possible

dbus/dbus-gvalue-parse-variant.c | 5 +++--

1 file changed, 3 insertions(+), 2 deletions(-)

commit 42192c5bc0517d747aa83e9ce82e0208c3426aab

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2010-10-14 12:06:36 +0100

Add dbus_g_value_parse_g_variant to gtkdoc

doc/reference/dbus-glib-sections.txt | 1 +

1 file changed, 1 insertion(+)

commit f3b4a2d82994f0013f14fffb3fedd8cd45cf7da

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>

Date: 2010-10-14 12:06:07 +0100

dbus_g_value_parse_variant_by_type: talk about GVariant, not
GDBus,

in docs


```
dbus/dbus-gvalue-parse-variant.c |    4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
commit bc207d10cace8051fff7b2ddf51176030c0d40d17
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-14 12:05:48 +0100
```

Adjust syntax to avoid relying on array/pointer duality

```
dbus/dbus-gvalue-parse-variant.c |    4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
commit a66176ecd5eab86d62e23c393c90dfda4d9ca354
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-14 12:04:29 +0100
```

dbus_g_value_basic_array_parse_variant: allow the fast path to be taken

```
dbus/dbus-gvalue-parse-variant.c |    2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit b9c4c4bf00321ab1fb092514c6b1312d82a098c3
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-13 13:27:33 +0100
```

Test dbus_g_value_parse_g_variant

Also test dbus_g_value_build_g_variant more thoroughly, by carrying out various round-trip conversions.

```
test/core/test-gvariant.c | 283
+++++
1 file changed, 283 insertions(+)
```

```
commit 817e1fc7c50ccde3643c20ed837dce00bc16a885
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-13 12:57:35 +0100
```

dbus_g_value_build_g_variant: cope with empty arrays, maps

We can't just derive the type from the first entry if there's no first entry, so for the empty case we need a function whose structure mirrors dbus_g_value_build_g_variant itself, which just returns the type.

```
dbus/dbus-gtype-specialized.c | 114
+++++
```

1 file changed, 112 insertions(+), 2 deletions(-)

commit 53d4457fb5bd71467acb1fbca7ecd80d8e39e189
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-13 12:52:39 +0100

dbus_g_type_specialized_map_append: document that the value contents are stolen

This is highly non-obvious, and tripped me up while writing dbus_g_value_parse_g_variant.

dbus/dbus-gtype-specialized.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 26e1c1d86b6e583ac326674ab8b886673f7c1f90
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-13 12:52:07 +0100

fd.o #30428: add dbus_g_value_parse_g_variant

This is the inverse of dbus_g_value_build_g_variant, and can be used to present an API with GVariant 'in' arguments to library users, then translate them into GValue for transmission over dbus-glib, as a transition mechanism.

dbus/Makefile.am | 2 +
dbus/dbus-glib.h | 2 +
dbus/dbus-gvalue-parse-variant.c | 525
++++
dbus/dbus-gvalue-parse-variant.h | 39 +++
4 files changed, 568 insertions(+)

commit abe6ee06e30f71187aeaed2c8804cbcc6692d8cc
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-13 12:48:51 +0100

replace remnants of gcov support with lcov.am from telepathy-glib

.gitignore | 4 ++++
Makefile.am | 27 +-----
tools/lcov.am | 47 +++++
3 files changed, 52 insertions(+), 26 deletions(-)

commit b4362df4d4b9ccd3246f102f7990dbda426f1b8c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-12 15:27:52 +0100

Remove gcov decoder, which hasn't worked since dbus-glib left libdbus

```
Makefile.am          | 16 +-
test/Makefile.am     | 12 +-
test/decode-gcov.c   | 2652
```

3 files changed, 2 insertions(+), 2678 deletions(-)

```
commit c42b86fecc793edaf82055a3552f1490be8ad29c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-12 15:19:24 +0100
```

Test dbus_g_value_build_g_variant for various fixed arrays

```
test/core/test-gvariant.c | 217
+++++
1 file changed, 217 insertions(+)
```

```
commit 75161c695e8d9729ead8543b1297202eee1894f9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-12 15:19:05 +0100
```

Actually run test/core/test-gvariant

```
test/core/run-test.sh | 1 +
1 file changed, 1 insertion(+)
```

```
commit 48b2b580c4d6fdcd9db83db7acebb0ea4be6fff9
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-10-12 14:41:35 +0100
```

Give specialized GArrays iteration/append support

This is necessary to have dbus_g_value_build_g_variant work on them.

```
dbus/dbus-gvalue-utils.c | 162
+++++
1 file changed, 160 insertions(+), 2 deletions(-)
```

```
commit 0ab27bb2b7c3f944348e6368069ffbfccfc2bcb5a
Author: Christian Dywan <christian.dywan@lanedo.com>
Date: 2010-07-21 17:28:09 +0200
```

Free looked up function name in dbus binding tool

```
dbus/dbus-binding-tool-glib.c | 11 ++++++----
1 file changed, 7 insertions(+), 4 deletions(-)
```

```
commit 5784ee8f24334e634e91b8761dfa956f97a0d2ba
Author: Christian Dywan <christian.dywan@lanedo.com>
Date: 2010-07-21 17:18:18 +0200
```

Always free method_c_name in dbus binding tool

```
dbus/dbus-binding-tool-glib.c | 11 ++++++---  
1 file changed, 9 insertions(+), 2 deletions(-)
```

```
commit 4acce76d97825721c1792417994e592542f4d3d9  
Author: Christian Dywan <christian.dywan@lanedo.com>  
Date: 2010-07-21 17:12:34 +0200
```

Free path string after emission in in statemachine server example

```
dbus/examples/statemachine/statemachine-server.c | 1 +  
1 file changed, 1 insertion(+)
```

```
commit ddb48ae24e40c8636b85555e8d288e2a33cadd44  
Author: Christian Dywan <christian.dywan@lanedo.com>  
Date: 2010-07-21 17:10:26 +0200
```

Plug leak of expected_str in threaded server test

```
test/core/test-thread-server.c | 1 +  
1 file changed, 1 insertion(+)
```

```
commit a07afee00e091a11dc9eefd775288a32f0e49e2d  
Author: Christian Dywan <christian.dywan@lanedo.com>  
Date: 2010-07-21 18:41:06 +0200
```

Dereference main loop once variant recursion test is done

```
test/core/test-variant-recursion.c | 2 ++  
1 file changed, 2 insertions(+)
```

```
commit ffe66eb93737eb5ac699b8acba7781a5f77bdab6  
Author: Christian Dywan <christian.dywan@lanedo.com>  
Date: 2010-07-22 16:26:15 +0200
```

Remove unused method attribute variables in introspect_interfaces

```
dbus/dbus-gobject.c | 4 ----  
1 file changed, 4 deletions(-)
```

```
commit afd098718e36b8ce51f565447f429d807bab3785  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2010-09-28 17:19:19 +0100
```

Add DbusGObjectPath, DbusGSignature typedefs

```
dbus/dbus-glib.h | 2 ++  
dbus/dbus-gvalue.c | 44  
+++++-----  
doc/reference/dbus-glib-sections.txt | 4 +++-  
3 files changed, 45 insertions(+), 5 deletions(-)
```

commit 8318613e6b186bd9e5c8a844c58beebf47426116
Author: Colin Walters <walters@verbum.org>
Date: 2010-08-12 10:19:35 -0400

configure: Release 0.88

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 510bdcd63ae4e588a5cb72727696d5ad7fd5298b
Author: Colin Walters <walters@verbum.org>
Date: 2010-04-19 16:47:11 -0400

Respect property access flags for writing, allow disabling for reads

Because DBus-GLib originally was designed as a generic "object mapping" binding, the handler for org.freedesktop.DBus.Properties simply allowed access (read or write) to any GObject property that was exported.

Later, the (compile time) introspection XML was added, and while we only listed "exported" properties in the dynamic introspection XML, we still allowed Get or Set calls to any property that was valid.

With this patch, we deny writes to properties which aren't listed in the XML, or are listed as read-only.

For backwards compatibility however, we still allow reads. A service may disable this by calling `dbus_glib_global_set_disable_legacy_property_access()`.

```
dbus/dbus-binding-tool-glib.c | 53 +++++--
dbus/dbus-glib.h              |  2 +
dbus/dbus-gobject.c          | 293
+++++-----
test/core/my-object.c        | 63 ++++++--
test/core/my-object.h        |  5 +
test/core/test-dbus-glib.c   | 252 ++++++-----
test/core/test-service-glib.xml |  5 +
7 files changed, 601 insertions(+), 72 deletions(-)
```

commit f0668d71d63f97c1c429dad165a30d3aadbdafa0f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-08-09 11:25:19 +0100

Consolidate LDADD variables in tests/examples and make them more complete

This fixes compilation with LDFLAGS=-Wl,--no-add-needed, which is the default behaviour of GNU gold, and of Fedora's patched GNU ld. See:

<http://fedoraproject.org/wiki/Features/ChangeInImplicitDSOLinking>

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=29274
Reviewed-by: Colin Walters <walters@verbum.org>

```
dbus/examples/Makefile.am      |    8 ++++----
dbus/examples/statemachine/Makefile.am |    6 ++++--
test/core/Makefile.am          |   32
+++++++-----
test/interfaces/Makefile.am    |    8 ++++----
4 files changed, 25 insertions(+), 29 deletions(-)
```

commit ccf4a4d8c82b7309af7f3fb23144690acbf8e38c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-08-09 11:25:55 +0100

Add various tests etc. to .gitignore

```
.gitignore |    7 ++++++
1 file changed, 7 insertions(+)
```

commit dcc9c8f2dede482614b885eace06f2dedaf2a736
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-07-27 17:16:33 +0100

Add Libtool m4 to .gitignore

```
.gitignore |    5 +++++
1 file changed, 5 insertions(+)
```

commit 9440209e203cccef158904800ea623568637a71e
Merge: 0eec114 077d445
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2010-08-05 10:09:08 +0100

Merge remote branch 'danni/gvalue-to-gvariant'

Fixes: https://bugs.freedesktop.org/show_bug.cgi?id=28715

commit 0eec114bb510a3bb30888b0020ca0dc11dc3497d
Author: Christian Dywan <christian@twotoasts.de>
Date: 2010-07-22 13:38:01 +0200

Refer to dbus_g_connection_flush rather than the plain dbus call

Fixes: https://bugs.freedesktop.org/show_bug.cgi?id=29214

```
dbus/dbus-gproxy.c |    6 +++---
```

1 file changed, 3 insertions(+), 3 deletions(-)

commit 077d445078663e3ee6f93beacc14d866504148bf
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-07-14 15:43:11 +1000

Support DBUS_TYPE_G_SIGNATURE

dbus/dbus-gtype-specialized.c | 2 ++
test/core/test-gvariant.c | 21 +++++
2 files changed, 23 insertions(+)

commit 13f7c51344111ca5266e790d2f4863fab770bfd8
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-06-25 14:35:17 +1000

Test test_g_variant_equivalent itself

test/core/test-gvariant.c | 144
++++
1 file changed, 144 insertions(+)

commit 90b5e06797f2c9c5ed8918d22de35d5093bc3bd1
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-06-25 14:11:40 +1000

Write a recursive equivalence function for testing the equivalence
of GVariants

Makes up for the limitations of g_variant_equal()

test/core/test-gvariant.c | 114
++++-----
1 file changed, 106 insertions(+), 8 deletions(-)

commit 5456dc1f10544505a61f2608f646256e28b0047a
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-06-24 22:10:48 +1000

Tests for GValue-to-GVariant

test/core/Makefile.am | 7 +-
test/core/test-gvariant.c | 191
++++
2 files changed, 197 insertions(+), 1 deletion(-)

commit 5cc0676c83ee678069886daad5fc5487f098686e
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-06-24 20:52:26 +1000

fd.o #28715: Add dbus_g_value_build_g_variant()

```
dbus/dbus-gtype-specialized.c | 127
+++++
dbus/dbus-gtype-specialized.h | 2 +
doc/reference/dbus-glib-sections.txt | 1 +
3 files changed, 130 insertions(+)
```

commit 5ed147db0808e5f4ef50dfd6011419c44255ea6c
Author: Danielle Madeley <danielle.madeley@collabora.co.uk>
Date: 2010-06-24 20:59:05 +1000

Bumping required GLib to 2.24 for GVariant

```
configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 7f054d7bce4b2ea74e5268f2cf65c467773ee14f
Author: Dan Williams <dcbw@redhat.com>
Date: 2010-06-29 21:19:25 -0700

Fix lookup of regular properties when shadow properties are used

Only free the uscore converted name if there's actually a shadow property registered for this property; otherwise if there is no shadow property we free the uscore converted one and then return it immediately after.

```
dbus/dbus-gobject.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

commit e6b6e27dc2d6e3abae88d9a773a1e4c016e58aa3
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-05-04 12:39:51 +0100

fd.o #27958: dbus_g_error_domain_register: rewrite the documentation

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

```
dbus/dbus-gobject.c | 30 ++++++-----
1 file changed, 25 insertions(+), 5 deletions(-)
```

commit 451183c140f7d6b42fdff3a6c190ef93d860c5fc
Merge: 7db074f 499beb6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2010-05-04 12:18:31 +0100

Merge branch '14579-remove-before-cancel'

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=14579

commit 7db074fead497c31041ebf417db26ae9a7ab565c

Author: Fridrich Å trba <fridrich.strba@bluewin.ch>
Date: 2010-04-26 18:51:49 +0200

Put the G_OS_WIN32 check where it can be defined.

```
test/core/test-profile.c | 5 +++--  
1 file changed, 3 insertions(+), 2 deletions(-)
```

commit fb7cb9f10e08eaf3887b2a63a7fa59510e40b5d2
Author: Fridrich Å trba <fridrich.strba@bluewin.ch>
Date: 2010-04-26 18:50:31 +0200

Fix linking of tests. On windows, undefined symbols are not allowed
and symbol lookup is sequential

```
test/core/Makefile.am | 2 +-  
test/interfaces/Makefile.am | 2 +-  
2 files changed, 2 insertions(+), 2 deletions(-)
```

commit fadbedfbbf512d69611a80a0cd4698e3b5d579ea
Author: Fridrich Å trba <fridrich.strba@bluewin.ch>
Date: 2010-04-26 18:44:21 +0200

Use EXEEXT so that we satisfy dependencies when cross-compiling

```
test/core/Makefile.am | 6 +++---  
test/interfaces/Makefile.am | 20 ++++++++-----  
2 files changed, 13 insertions(+), 13 deletions(-)
```

commit f0abb1f78d4e4764119b57d2b6f3b5fa57d8fdfb
Author: Dan Williams <dcbw@redhat.com>
Date: 2010-04-22 13:26:44 -0700

core: don't pass malformed error interface to dbus (rh#581794)

While clients should really register their errors, dbus-glib shouldn't be passing a malformed error interface to dbus either. It's just not nice and libdbus will call abort().

See https://bugzilla.redhat.com/show_bug.cgi?id=581794

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-gobject.c | 17 ++++++++-----  
test/core/my-object.c | 12 ++++++++  
test/core/my-object.h | 1 +  
test/core/test-dbus-glib.c | 8 ++++++  
test/core/test-service-glib.xml | 3 +++  
5 files changed, 35 insertions(+), 6 deletions(-)
```

commit 47adc3b17f20a78230c5c34e635fc30bb7afbde2

Author: Colin Walters <walters@verbum.org>
Date: 2010-04-26 10:51:39 -0400

Disable test-profile on win32 for now

It uses Unix sockets, and overall isn't very interesting anyways.

```
test/core/test-profile.c | 13 ++++++++--  
1 file changed, 11 insertions(+), 2 deletions(-)
```

commit 16deb341b52421695b81ebe8c91c81f47eb173f3
Author: Fridrich Strba <fridrich.strba@bluewin.ch>
Date: 2010-04-26 10:31:23 -0400

Don't use the identifier "interface" in public headers

This causes problems on Windows.

```
dbus/dbus-gidl.h | 2 +-  
dbus/dbus-glib.h | 6 +++---  
2 files changed, 4 insertions(+), 4 deletions(-)
```

commit 354ed0897ae0e815e95a7eb931de6ba35325121c
Author: Fridrich Strba <fridrich.strba@bluewin.ch>
Date: 2010-04-26 10:29:41 -0400

Use AC_CANONICAL_HOST, not _TARGET

This is what GLib uses; see the Autoconf manual for the full explanation of the train wreck.

```
configure.ac | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit af0ec6795ea64d6498b83f30ecd94adc4e935e7e
Author: Astone Lin <astone.lin@mediatek.com>
Date: 2010-04-19 06:56:43 -0400

Allow duplicate object path registrations for different connections

We clearly need to respect the connection when comparing registrations, since it's perfectly valid to have the same one on two different connections.

```
dbus/dbus-gobject.c | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit fc7114a536571688d48ede025f1357c0e467c879
Author: Guillaume Desmottes <guillaume.desmottes@collabora.co.uk>
Date: 2010-04-07 12:08:59 +0200

add missing DBUS_TYPE_G_* to the doc

Reviewed-by: Will Thompson <will.thompson@collabora.co.uk>

Reviewed-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

doc/reference/dbus-glib-sections.txt | 9 ++++++++
1 file changed, 9 insertions(+)

commit 8df3281e536557cfd6b4056a9c95102d57596179
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-24 20:17:54 -0400

[configure] Release 0.86

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 551d86f7fe0e7f2304f57baa7a406c935182b06a
Author: Dan Williams <dcbw@redhat.com>
Date: 2010-03-24 14:40:09 -0700

core: performance optimization for object info lookup

dbus/dbus-gobject.c | 31 ++++++-----
1 file changed, 18 insertions(+), 13 deletions(-)

commit ed73cb2f32c411a3306c6dfe2ae541aea249f200
Author: Dan Williams <dcbw@redhat.com>
Date: 2010-03-24 12:20:13 -0700

core: allow duplicate property names on GInterfaces

Allows dbus-glib clients to handle objects that implement more than one interface with the properties of the same name. Normally this would not be allowed since with GObject all properties are in the same namespace. This patch allows the interface to register "shadow" properties on a per-interface basis which redirect the lookup of the property name.

dbus/dbus-glib.h | 4 +
dbus/dbus-gobject.c | 160 ++++++-----

test/interfaces/Makefile.am | 24 +++-
test/interfaces/test-client.c | 186
+++++
test/interfaces/test-dup-prop-a.xml | 6 ++
test/interfaces/test-dup-prop-b.xml | 6 ++
test/interfaces/test-dup-prop.c | 192
+++++
test/interfaces/test-dup-prop.h | 60 ++++++
test/interfaces/test-server.c | 8 ++

9 files changed, 627 insertions(+), 19 deletions(-)

commit c1bbf6fb1b7c6e2a743c8a7353cddea52d42fa2d
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-18 21:39:24 -0400

[dbus-gobject.c] Trivial compiler warning fixes

dbus/dbus-gobject.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)

commit b52c16b495632bd945328ca858533d5660106bdf
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-18 21:37:38 -0400

Remove dbus-glib-undeclared.txt from git; it's a generated file

0 files changed

commit 7b029451daclfdd14da4e558cc995fb7f5263d04
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-18 21:37:05 -0400

[configure.ac] Use AM_SILENT_RULES if available

configure.ac | 2 ++
1 file changed, 2 insertions(+)

commit db026859406ce768d71d4ccf850e8be56b52998c
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2010-03-09 17:18:10 +0000

Free errors returned by method implementations

https://bugs.freedesktop.org/show_bug.cgi?id=26981

dbus/dbus-gobject.c | 4 ++++
1 file changed, 4 insertions(+)

commit 3902ac5a13a7510b8411b1147d0cce5711cf4168
Author: Sven Herzberg <herzi@gnome-de.org>
Date: 2010-03-08 16:50:28 +0100

turn the gtk-doc documentation into buildable shape

* dbus/dbus-gobject.c: the < and > symbols broke the sgml
generation
in
gtk-doc

dbus/dbus-gobject.c | 6 +++---
1 file changed, 3 insertions(+), 3 deletions(-)

commit 3d69cfeab177e79b4ecfelb4284a5bd840fd11e7
Author: Colin Walters <walters@verbum.org>
Date: 2009-04-02 14:01:05 +0100

Fix hyphenated error codes correctly

The error code names generated by glib-mkenums separate the words by hyphens which are invalid D-BUS error names. This patch converts them back to wincaps, but we can't uppercase the first letter.

Based on an original patch from Neil Roberts <neil@linux.intel.com>

```
dbus/dbus-gobject.c | 21 ++++++-----  
test/core/my-object.c | 13 ++++++  
test/core/my-object.h | 4 +++-  
test/core/test-dbus-glib.c | 7 ++++++  
test/core/test-service-glib.xml | 3 +++  
5 files changed, 44 insertions(+), 4 deletions(-)
```

commit 4538adc1fe903d1ce71ca24590df9bdac7c4a50e
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-02 12:12:18 -0500

Revert "Squash underscores and dashes in errors when converting to DBus errors"

This reverts commit 9637ed9f0c66982a06048b18ccf983881643e456.

This incorrectly uppercased the first character in error names:
https://bugzilla.redhat.com/show_bug.cgi?id=569631

```
dbus/dbus-gobject.c | 13 +-  
1 file changed, 2 insertions(+), 11 deletions(-)
```

commit 2d85761286e96df3490426af1e20e7b553448092
Author: Colin Walters <walters@verbum.org>
Date: 2010-03-02 13:52:36 -0500

Remove dbus-glib-undocumented.txt from git

It's autogenerated by gtk-doc, so shouldn't be in git.

```
doc/reference/dbus-glib-undocumented.txt | 60  
-----  
1 file changed, 60 deletions(-)
```

commit 9637ed9f0c66982a06048b18ccf983881643e456
Author: Colin Walters <walters@verbum.org>

Date: 2010-02-01 10:09:53 -0500

Squash underscores and dashes in errors when converting to DBus errors

We were just taking the enumeration nick and appending to the DBus error name, but since these can contain '-' we need to squash.

```
dbus/dbus-gobject.c |      13 ++++++++--  
1 file changed, 11 insertions(+), 2 deletions(-)
```

commit d039ca66f85955788e2d58294c249579bed9cc03
Author: Colin Walters <walters@verbum.org>
Date: 2010-01-27 13:56:48 -0500

Release 0.84

```
configure.ac |      2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit f1c3f7bf585d6010c84491372b0a6f0ae3b1432a
Author: Colin Walters <walters@verbum.org>
Date: 2010-01-27 13:01:29 -0500

Add GMainContext to dbus_g_bus_get_private, add a test case

To even sort of work with threads right now, a common workaround is to open a private connection. This patch more explicitly supports creating a private connection, associating it with the GMainContext which will be used for a thread.

Also, add a (very simple) test case which just uses a private connection for the default main context.

```
dbus/dbus-glib.h      |  1 +  
dbus/dbus-gmain.c    |  4 +++-  
test/core/test-dbus-glib.c | 21 ++++++++  
3 files changed, 25 insertions(+), 1 deletion(-)
```

commit 87c645ed17d6fef350e8c26e322ecde566a27d42
Author: Colin Walters <walters@verbum.org>
Date: 2010-01-27 13:06:34 -0500

Fix compilation of dbus-gvalue.c

commit c4a5653e0402f876a1824c33a19e69991ee46a27 was missing a "break;"

```
dbus/dbus-gvalue.c |  1 +  
1 file changed, 1 insertion(+)
```

commit e6cd65545ee5cd2eeb5ccc21d5bfe8e8ac286d95

Author: Colin Walters <walters@verbum.org>
Date: 2010-01-27 11:40:07 -0500

Add note about non-maintenance of NEWS

NEWS | 7 +++++++
1 file changed, 7 insertions(+)

commit c4a5653e0402f876a1824c33a19e69991ee46a27
Author: Andres Salomon <dilinger@collabora.co.uk>
Date: 2009-08-19 08:57:21 -0400

dbus-gvalue: set an error when demarshal_basic doesn't recognize type

By passing `dbus_g_proxy_call` an incorrect signature, we can cause the function to fail but not provide any error message (if `G_DISABLE_ASSERT` is defined). As `smvc` pointed out, this can also become a runtime error when a telepathy CM changes signature, but the client hasn't been updated. As such, `g_assert` isn't an appropriate action. See <http://bugs.debian.org/541632> for more information.

This patch causes it to set an error when demarshalling a type that it's not expecting. Instead of a NULL error, one instead sees something like the following when the method returns a path object but the client expected a string:

```
"modem Create() failed: Expected type gchararray, got type code 'o'"
```

Signed-off-by: Andres Salomon <dilinger@collabora.co.uk>

dbus/dbus-gvalue.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)

commit b976413f896f2d5a0be9449fc77a716f2e67cc96
Author: Jiří Klimeš <jklimes@redhat.com>
Date: 2010-01-27 10:36:57 -0500

Fix bad NAME in dbus-binding-tool man page

doc/dbus-binding-tool.1 | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 34636b12c62523b9db789b0fbef3a86782debf10
Author: Colin Walters <walters@verbum.org>
Date: 2010-01-15 12:23:30 -0500

Import `dbus-bus-introspect.xml` upstream

Rather than having consumers pass an externally-generated XML file,
just include one here. In practice our target audience is OS builders,
who if they have the capability to update dbus, also have the capability
to update dbus-glib.

```
Makefile.am          |    1 +  
configure.ac         |   12 -----  
dbus-bus-introspect.xml |    77  
+++++  
tools/Makefile.am    |   15 +------  
4 files changed, 81 insertions(+), 24 deletions(-)
```

```
commit 9cefa4bae8d20beddf695380af00945790da0206  
Merge: 5e5a1c1 15f4533  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2009-11-25 14:55:32 +0000
```

Merge remote branch 'upstream/master'

```
commit 15f4533742ad6aeab85ebb723ac8fca99c00e939  
Author: Will Thompson <will.thompson@collabora.co.uk>  
Date: 2009-11-16 13:32:40 +0000
```

Don't leak DbusGMethodInvocation for no-reply calls

https://bugs.freedesktop.org/show_bug.cgi?id=25119

```
dbus/dbus-gobject.c |   17 ++++++-----  
1 file changed, 10 insertions(+), 7 deletions(-)
```

```
commit 5e5a1c1930173dac0e331c2cb78a164c3565b0eb  
Merge: 2a4396b 90e2199  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2009-09-29 14:22:07 +0100
```

Merge remote branch 'wjt/duplicate-registrations'

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
commit 2a4396b523912c3bf4d2a1ef6d6fa930c0871c67  
Merge: 980b468 e2007cf  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2009-09-29 14:07:11 +0100
```

Merge branch '20936-fsf-address'

Reviewed-by: Colin Walters <walters@verbum.org>

```
commit 980b46870aa0e44a04ad1191d7af1d0601577501
```


Author: Stian Skjelstad <stian@nixia.no>
Date: 2009-09-18 14:01:00 -0400

Bug 19623 - Add dbus_g_bus_get_private

Useful for cases where you have to get a private connection, among them to work around threading issues.

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-glib.h | 3 +++  
dbus/dbus-gmain.c | 41 +++++++++++++++++++++++++++++++++++++++  
2 files changed, 44 insertions(+)
```

commit 90e2199ac99f5b8ab0cd5f45dcb398ecf9af03d9
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2009-09-12 11:58:22 +0100

Copy object registration list when unregistering.

Since the list of registrations on the object is modified when each path is removed, iterating it directly is wrong: after the first pass of the loop, 'iter' would point to a link which has been freed.

```
dbus/dbus-gobject.c | 8 ++++++-  
1 file changed, 7 insertions(+), 1 deletion(-)
```

commit 39e2642e41b2293de7556fa15c57872f78ffcdc8
Author: Will Thompson <will.thompson@collabora.co.uk>
Date: 2009-09-12 11:28:25 +0100

Only re-set registration list if it's non-empty

```
dbus/dbus-gobject.c | 4 +---  
1 file changed, 3 insertions(+), 1 deletion(-)
```

commit 0263b72669de710e0adda419a520ae9f123a1be9
Author: Colin Walters <walters@verbum.org>
Date: 2009-08-19 13:27:53 -0400

Support duplicate object registrations

Before commit e869fda4, we semi-supported registering the same object multiple times. We'd accept messages for both paths, however when signals were emitted, they'd both use the first object path.

That commit simply disallowed multiple registrations, which broke backwards compatibility with some projects like PolicyKit which

had the same object registered with different paths.

With this commit, explicitly allow and support multiple registrations.

The primary change is that signals are now emitted once for each registration path of an object, using the correct path.

```
dbus/dbus-gobject.c      | 159
+++++-----
test/core/test-dbus-glib.c | 41 +++++
test/core/test-service-glib.c | 4 ++
3 files changed, 154 insertions(+), 50 deletions(-)
```

commit c993494ad463ffe55b8603e58aa5ecb6dd731144
Author: Colin Walters <walters@verbum.org>
Date: 2009-07-16 14:09:18 -0400

Release 0.82

```
configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 65e03ccf0f2417a83fd187035b80d680e376fd32
Author: Colin Walters <walters@verbum.org>
Date: 2009-07-16 13:51:57 -0400

Test for git-log in Makefile.am should just be for git now

The combined tools are no longer in \$PATH for modern git.

```
Makefile.am | 8 ++++----
1 file changed, 4 insertions(+), 4 deletions(-)
```

commit f5edb761202d3aff8f8bd0bc68dd57d3d32d2bd1
Author: Alban Crequy <alban.crequy@collabora.co.uk>
Date: 2009-07-16 13:39:08 -0400

Bug 18294 - Be defensive about a possibly NULL property string

As far as I can tell we should always be writing one, but it doesn't hurt to guard against NULL here.

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-gobject.c | 4 +++-
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 2508e687e7d7792a656c7af0aad1be851faa33bb
Author: Brian Cameron <brian.cameron@sun.com>
Date: 2009-07-16 13:15:38 -0400

Bug 20343 - Add a man page for dbus-binding-tool

Signed-off-by: Colin Walters <walters@verbum.org>

```
doc/Makefile.am |      6 +-
doc/dbus-binding-tool.1 | 294
+++++
2 files changed, 299 insertions(+), 1 deletion(-)
```

commit 29513843b22dbd2aba1d3ba2930bb38e2c6631e5
Author: Marc-Andr   Lureau <marcandre.lureau@gmail.com>
Date: 2009-05-09 18:13:49 +0300

build: fix undefined macro: AM_PROG_LIBTOOL with recent libtool

Without this patch, I get:

```
configure.ac:206: warning: macro `AM_PROG_LIBTOOL' not found
in library
configure.ac:213: error: possibly undefined macro:
AM_PROG_LIBTOOL
    If this token and others are legitimate, please use
    m4_pattern_allow.
    See the Autoconf documentation.
autoreconf2.50: /usr/bin/autoreconf failed with exit status: 1
```

(I am using libltdl-dev 2.2.6a-4 from Debian)

```
Makefile.am |      1 +
autogen.sh  |      1 +
2 files changed, 2 insertions(+)
```

commit d66337bdf9fc8f5372d84b05ae68c818bb9c881b
Author: Luis Menina <liberforce@freeside.fr>
Date: 2009-07-16 10:55:39 -0400

Bugg 22244 - Only include <glib.h>, not individual headers

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-glib-tool.h |      2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 67c289a60a4ac2ee9660f38a7168a238c1b3cffb
Author: Colin Walters <walters@verbum.org>
Date: 2009-07-15 15:33:31 -0400

Clean up some compilation warnings

```
dbus/dbus-gloader-expat.c |      6 +++---
dbus/dbus-gobject.c       |      4 ++--
test/core/test-dbus-glib.c |      2 +-

```

3 files changed, 6 insertions(+), 6 deletions(-)

commit 5eea6af0d24780109dbbc15b8bfa4078df3a80a2
Author: Colin Walters <walters@verbum.org>
Date: 2009-07-15 15:28:06 -0400

Add missing prototype for `dbus_g_connection_unregister_g_object`

The function was introduced in commit `b4911558384de` and intended to be public, make it so.

dbus/dbus-glib.h | 2 ++
1 file changed, 2 insertions(+)

commit de68d2b2fc3012ae733212a704aa29a748f22f80
Author: Brian Tarricone <bjt23@cornell.edu>
Date: 2009-07-15 15:03:36 -0400

Bug 21753 - Correctly initialize GValues in `dbus-binding-tool` generated code

GValues should have their type set to 0.

Signed-off-by: Colin Walters <walters@verbum.org>

dbus/dbus-binding-tool-glib.c | 12 ++++++-----
1 file changed, 8 insertions(+), 4 deletions(-)

commit af07e13e6afe09456588ddba4b72cba215bc7c93
Author: Jason Leach <jason@archlinux.us>
Date: 2009-07-01 15:41:04 -0400

Bug 21362 - Remove use of deprecated symbols

The `_set*_take_ownership` are replaced by `_take`, and use `g_main_loop_run`.

dbus/dbus-binding-tool-glib.c | 6 +++---
dbus/dbus-gvalue-utils.c | 14 ++++++-----
dbus/dbus-gvalue.c | 16 ++++++-----
m4/gtk-doc.m4 | 2 +-
test/core/test-dbus-glib.c | 2 +-
test/core/test-thread-client.c | 2 +-
test/core/test-thread-server.c | 2 +-
7 files changed, 22 insertions(+), 22 deletions(-)

commit e2007cfc5822f057b975660bd369fee56a7f2eb6
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-06-09 16:22:46 +0100

fd.o #20936: Fix another couple of instances of the old FSF address

`git grep 'Temple Place'` now returns nothing.

```
dbus/dbus-bash-completion-helper.c | 2 +-
dbus/dbus-gtool-test.h              | 2 +-
dbus/dbus-gtype-specialized-priv.h | 2 +-
3 files changed, 3 insertions(+), 3 deletions(-)
```

commit cbe5250e553112221e1230301767ad1052790b0a
Author: Tobias Mueller <fdo-bugs@cryptobitch.de>
Date: 2009-04-15 21:37:05 +0100

fd.o #20936: Update FSF address

The glib bindings contain an outdated address of the FSF:
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-
1307

The new address is:
<http://www.fsf.org/about/contact.html> writes:
> Free Software Foundation
> 51 Franklin Street, Fifth Floor
> Boston, MA 02110-1301
> USA

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
COPYING          | 4 ++--
dbus/dbus-binding-tool-glib.c | 2 +-
dbus/dbus-binding-tool-glib.h | 2 +-
dbus/dbus-gidl.c      | 2 +-
dbus/dbus-gidl.h     | 2 +-
dbus/dbus-glib-lowlevel.h | 2 +-
dbus/dbus-glib-tool.c | 2 +-
dbus/dbus-glib-tool.h | 2 +-
dbus/dbus-glib.c     | 2 +-
dbus/dbus-glib.h     | 2 +-
dbus/dbus-gloader-expat.c | 2 +-
dbus/dbus-gmain.c    | 2 +-
dbus/dbus-gobject.c  | 2 +-
dbus/dbus-gobject.h  | 2 +-
dbus/dbus-gparser.c  | 2 +-
dbus/dbus-gparser.h  | 2 +-
dbus/dbus-gproxy.c   | 2 +-
dbus/dbus-gsignature.c | 2 +-
dbus/dbus-gtest-main.c | 2 +-
dbus/dbus-gtest.c    | 2 +-
dbus/dbus-gtest.h    | 2 +-
dbus/dbus-gthread.c  | 2 +-
dbus/dbus-gtype-specialized.c | 2 +-
dbus/dbus-gtype-specialized.h | 2 +-
dbus/dbus-gutils.c   | 2 +-

```

```
dbus/dbus-gutils.h          |    2 +-
dbus/dbus-gvalue-utils.c   |    2 +-
dbus/dbus-gvalue-utils.h   |    2 +-
dbus/dbus-gvalue.c         |    2 +-
test/core/test-profile.c   |    2 +-
test/decode-gcov.c         |    2 +-
31 files changed, 32 insertions(+), 32 deletions(-)
```

```
commit 499beb64a12cead99ccb09d7de01890aaa800ef2
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 16:04:57 +0000
```

Bug 14579: remove pending call from hash table before cancelling it

Previously, the code implicitly assumed that cancelling the pending call would not cause the DBusGProxy to be freed. This can fail if `user_data` for the pending call holds the last reference to the DBusGProxy - in this case, it is unsafe to be manipulating the contents of "priv" after the call is cancelled.

```
dbus/dbus-gproxy.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)
```

```
commit b615918eef9723c00f03d30903181d9989dfd261
Merge: cb8a113 f550c49
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-06-09 16:15:49 +0100
```

Merge branch '13908-autoinit'

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Reviewed-by: Colin Walters <walters@verbum.org>
Bug: http://bugs.freedesktop.org/show_bug.cgi?id=13908

```
commit f550c492a10922795099166db34d8ccf5dbe63a7
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-05-28 18:21:27 +0100
```

fd.o #13908: silently initialize specialized types whenever required

Colin Walters asks: "Is there a reason not to just do the initialization at the time someone calls one of the public API entry points that depends on it?". This turns out to be non-trivial because those public

API entry
points are themselves used inside the initialization.

```
dbus/Makefile.am          | 1 +
dbus/dbus-gtype-specialized-priv.h | 48 ++++++
dbus/dbus-gtype-specialized.c      | 64
+++++-----
dbus/dbus-gvalue-utils.c          | 11 +++++-
4 files changed, 101 insertions(+), 23 deletions(-)
```

commit cb8a113be8808fe2c88d3e7851e79115235f9e8c
Merge: 0d64d31 725e98f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-05-28 18:04:20 +0100

Merge branch '20716-with-dbus-binding-tool'

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>
Bug: https://bugs.freedesktop.org/show_bug.cgi?id=20716
Reviewed-by: Colin Walters <walters@verbum.org>

commit 0d64d318397da26f224abd2d7ef1f27d14a544fa
Merge: 48abffb b93c2a1
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-05-28 18:02:44 +0100

Merge branch '5688-survive-disconnection'

This is unsuitable for cherry-picking by distros, since it adds
API.

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=21219
Reviewed-by: Colin Walters <walters@verbum.org>

commit 48abffba4cd81c182901dd7c80987251ba05348e
Merge: 9a0c188 e869fda
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-05-28 18:01:54 +0100

Merge part of branch '5688-survive-disconnection'

Bug: https://bugs.freedesktop.org/show_bug.cgi?id=5688
Reviewed-by: Colin Walters <walters@verbum.org>

commit 9a0c188e2662809950738e92b963e5602dfbd8eb
Merge: 03de444 98087a3
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-05-07 15:33:15 +0100

Merge branch '16776-make-dbus-errors-throwable'

Reviewed-by: Colin Walters <walters@verbum.org>

commit 03de44461132ed610c3d52133d4b4760f20e4cb2
Merge: b6b3787 f363811
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-27 10:47:58 +0100

Merge branch '20884-proxy-manager-replace-name-owner'

Reviewed-by: Colin Walters <walters@verbum.org>

commit b6b37871960ee06d9cc589e031ec3d5d6a4a6b92
Merge: e11c0ac d275c2e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 19:25:02 +0100

Merge branch '20886-configure-cleanup'

Reviewed by Colin Walters, fd.o #20886.

commit e11c0aced3b00733766207f15876fb300d63d442
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-25 11:27:57 -0500

Bug 19927 - Use const for GError * param we're not modifying

This makes it clearer we're not taking ownership of the error, and interoperates more nicely with functions which provide const GError *.

```
dbus/dbus-glib.h      |      2 +-  
dbus/dbus-gobject.c  |      2 +-  
2 files changed, 2 insertions(+), 2 deletions(-)
```

commit a809ea3016a0eae166df13c757f3e917dea0c0f9
Author: Vincent Untz <vuntz@gnome.org>
Date: 2009-04-16 12:35:42 -0400

Bug 20879 - Use --skip-source argument for glib-genmarshal

This avoids adding debug information for the temporary file.

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-binding-tool-glib.c |      1 +  
1 file changed, 1 insertion(+)
```

commit b93c2a19f468930c6badf054f79597d99d86e76c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 12:59:10 +0100

fd.o #21219: regression test


```
test/core/Makefile.am      |    22 ++++++++-----
test/core/run-test.sh     |     1 +
test/core/unregister.c    |    71
+++++++
3 files changed, 93 insertions(+), 1 deletion(-)
```

commit e869fda44331e6f590f974ad779115a22f5af22e
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 12:57:43 +0100

dbus-gobject: save the ObjectRegistration on each object, not just the path

```
dbus/dbus-gobject.c      |    20 ++++++++-----
1 file changed, 13 insertions(+), 7 deletions(-)
```

commit b4911558384de9919a231c29d75d9560f81fad0a
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 12:58:32 +0100

fd.o #21219: implement unregistration of objects

```
dbus/dbus-gobject.c      |    25 ++++++++-----
doc/reference/dbus-glib-sections.txt |     1 +
2 files changed, 25 insertions(+), 1 deletion(-)
```

commit e42535c3c0dace3ede96c50f5a9b3cc315c80190
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 12:11:15 +0100

fd.o #5688: Add a regression test

Also assert that `dbus_g_connection_lookup_g_object` works (my previous attempt to fix #5688 broke this).

```
test/core/5688.c         |    81
+++++++
test/core/Makefile.am    |    13 ++++++--
test/core/run-test.sh   |     1 +
3 files changed, 94 insertions(+), 1 deletion(-)
```

commit 6de1441865da2816c6bcd8cae842be93a8a96304
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-04-16 12:06:26 +0100

fd.o #5688: don't assert when exported object is destroyed *after* D-Bus connection closes

My solution was to introduce an `ObjectRegistration` struct which encapsulates the D-Bus <-> GObject glue.

Also, warn and keep the first object path if the library user registers an object at two object paths (previously, this would fail silently, use the second object path, and leak memory).

```
dbus/dbus-gobject.c |      111
+++++-----
1 file changed, 88 insertions(+), 23 deletions(-)
```

commit 725e98ffa810d04baf38e5d5d3f62091334b0888
Author: Peter Korsgaard <jacmet@sunsite.dk>
Date: 2009-04-15 21:29:09 +0100

add --with-dbus-binding-tool option to use an external
dbus-binding-tool

Cross compilation fix. dbus-binding-tool needs to run on the host, but gets compiled for the target. Add an option to use an external program (host version) instead, similar to the --with-introspect-xml option.

Signed-off-by: Peter Korsgaard <jacmet@sunsite.dk>
Signed-off-by: Robert Schwebel <r.schwebel@pengutronix.de>

```
configure.ac |      5 +++++
dbus/examples/Makefile.am |      4 +-
dbus/examples/statemachine/Makefile.am |      4 +-
test/core/Makefile.am |      6 +++-
test/interfaces/Makefile.am |     12 +++++-----
tools/Makefile.am |      2 +-
6 files changed, 19 insertions(+), 14 deletions(-)
```

commit a0124a5ecc829b73e2293cd7b23bf0cf0eb0a87c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-03-26 18:06:53 +0000

dbus_g_type_specialized_init: make some effort at being thread-safe

dbus-glib isn't really thread-safe and needs a systematic audit, but let's at least be preemptively thread-safe in new code.

Signed-off-by: Simon McVittie <simon.mcvittie@collabora.co.uk>

```
dbus/dbus-gtype-specialized.c |     21 +++++-----
1 file changed, 15 insertions(+), 6 deletions(-)
```

```
commit f36381131b6f410333a9a823a4fc131ac799394f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2009-03-26 18:00:16 +0000
```

fd.o#20884: dbus_g_proxy_manager_replace_name_owner: don't leave freed memory in the hash table if the name was the owner's first

Here's a situation where this code would fail:

```
* an owner :1.42 owns a name com.Example and a name org.Example
* the owner_names hash table contains { :1.42 => c }, where
  c is a GSList link with data = "com.Example", next = o and
  o is a GSList link with data = "org.Example", next = NULL
* the name owner for com.Example changes so :1.42 no longer owns
the
  name
* initially, names == c
* g_slist_delete_link unlinks and frees c, and sets names = o
* but c is still in the hash table, so next time we look in the
hash
  table, we crash
```

The fix is to replace c with o in the owner_names hash table.

```
dbus/dbus-gproxy.c | 13 ++++++++---
1 file changed, 10 insertions(+), 3 deletions(-)
```

```
commit d275c2e646f67a3929d9290c6bdb270d6346b83c
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:14:47 +0000
```

Remove all sorts of libdbus cruft from configure.ac

```
configure.ac      | 102
-----
test/Makefile.am | 1 -
2 files changed, 103 deletions(-)
```

```
commit 6791d09f72a63dcb72d57a187210bc133f463fcf
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:07:06 +0000
```

Don't check for atomic integer ops

GLib has perfectly good ones, and `git grep ATOMIC` confirms that we no longer use the result of the check

```
configure.ac | 26 -----
1 file changed, 26 deletions(-)
```

```
commit 6de33eb0a3e53d92e47a717849f144a0a313e923
```

Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:06:29 +0000

Don't check whether va_copy works

```
`git grep DBUS_VA_COPY` confirms that we no longer care  
configure.ac | 75  
-----  
1 file changed, 75 deletions(-)
```

commit 7bf91463598fc1e6ad44096f33aa4b61bf58443f
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:04:09 +0000

configure.ac: Don't check endianness

```
`git grep ENDIAN` confirms that we no longer need it.  
configure.ac | 3 ---  
1 file changed, 3 deletions(-)
```

commit dc93d2a6eaf75bd783d16d90f62d8abefd4e26a4
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:01:32 +0000

Don't define DBUS_API_SUBJECT_TO_CHANGE

It isn't, and we need libdbus >= 1.1 in any case

```
configure.ac | 2 --  
1 file changed, 2 deletions(-)
```

commit 094f45d0097679e8d458431231bcb3c999568d60
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 17:00:06 +0000

configure.ac: don't check the sizes of integers

This is presumably a relic of libdbus.

```
configure.ac | 105  
-----  
1 file changed, 105 deletions(-)
```

commit d30b06afd0a8bb6dddeefd7ac4d978799df03a52
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-11-28 16:58:37 +0000

Remove CXX cruft from configure.ac

There is no C++ in this package, so why are we checking for a C++

compiler?

```
configure.ac |      3 ---  
1 file changed, 3 deletions(-)
```

```
commit 98087a3d62523b621a1bd2359661ffe24553c34e  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2008-11-28 17:22:59 +0000
```

Bug #16776: add a regression test

```
test/core/my-object.c          |    11 ++++++++  
test/core/my-object.h          |     1 +  
test/core/test-dbus-glib.c     |    13 ++++++++  
test/core/test-service-glib.xml |     3 +++  
4 files changed, 28 insertions(+)
```

```
commit 20c474b8b8b356836e50af0e9bf3b32584d83ac4  
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>  
Date: 2008-11-28 16:23:43 +0000
```

Bug 16776: teach dbus_g_method_return_error about DBUS_GERROR

Code called by dbus_g_method_return_error assumes that errors in the

domain DBUS_GERROR always have the code DBUS_GERROR_REMOTE_EXCEPTION.

This is clearly not true, and it would be nice to be able to raise the

"well-known" D-Bus errors from library user code.

```
dbus/dbus-gobject.c |      69  
+++++-----  
1 file changed, 66 insertions(+), 3 deletions(-)
```

```
commit e18f5dd13c562333bc580f3b5bdf3ecd8d7555ce  
Author: Mikkel Kamstrup Erlandsen <mikkel.kamstrup@gmail.com>  
Date: 2008-11-28 15:58:43 +0000
```

Bug 13908: make dbus_g_type_specialized_init() safe for library users to call

```
dbus/dbus-gtype-specialized.c |     6 +++++  
dbus/dbus-gvalue.c            |     1 -  
2 files changed, 6 insertions(+), 1 deletion(-)
```

```
commit bc474819fa638a0daf3c89e6d041949b73e9228e  
Author: Sjoerd Simons <sjoerd@luon.net>  
Date: 2009-03-17 11:21:56 +0000
```

Use g_strdup instead of strdup in dbus_g_method_get_sender

dbus/dbus-gobject.c | 6 +-----
1 file changed, 1 insertion(+), 5 deletions(-)

commit dfef72c61c050e7f57e1d2eb601701e0a27827cc
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-05 11:17:15 -0500

Bug 14183 - Listen to NameOwnerChanged using arg0 matching

This is more efficient - we avoid waking up every dbus-glib using
process
when a process joins or leaves the bus.

dbus/dbus-gproxy.c | 57
+++++-----
1 file changed, 29 insertions(+), 28 deletions(-)

commit 42b17678dec6d0100bc433441f6a75df747e5c
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-05 10:59:15 -0500

Use -fno-strict-aliasing by default

We're unlikely to be strict-aliasing safe anytime soon. This
change
corresponds with http://bugs.freedesktop.org/show_bug.cgi?id=10599

configure.ac | 8 +++++++
1 file changed, 8 insertions(+)

commit 56b9773dc5dd589638e22f48ec4222ae491c2bc1
Author: Frederic Crozat <fcrozat@mandriva.com>
Date: 2009-02-03 14:00:50 +0100

Fix format-security warning

dbus/dbus-gobject.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 300b4fd822b8f57802ca35f4c867464b881632e3
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-02 19:40:51 -0500

Bump configure for unstable cycle

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 49cad2ca523de6a85f12d22ade23046c548ea601
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-02 19:40:29 -0500

Release 0.80

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit b161082148a51f2a2d8a4ee85f5736791dedc276
Author: Colin Walters <walters@verbum.org>
Date: 2009-02-02 15:07:51 -0500

Bug 19065 - handle nested nodes in dbus-binding-tool

Patch modified from one provided by Stian Skelstad
<stian@nixia.no>.

Split the generate_glue function explicitly into a toplevel
function and
one for processing recursive nodes.

```
dbus/dbus-binding-tool-glib.c | 108  
+++++-----  
m4/gtk-doc.m4 | 2 +-  
test/Makefile.am | 5 ++  
test/core/Makefile.am | 6 +--  
test/data/nested-introspect.xml | 10 ++++  
test/test-compile-nested.sh | 5 ++  
6 files changed, 85 insertions(+), 51 deletions(-)
```

commit 64ec55f59a9ebe699e240c2a3b4d82e85f2db4a0
Author: Doug Goldstein <cardoe@gentoo.org>
Date: 2009-01-30 17:38:19 -0500

Bug 19325 - parallel build fix

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/Makefile.am | 8 ++++----  
1 file changed, 4 insertions(+), 4 deletions(-)
```

commit 299460549e92665e5ea09d459d3985fc85d8d73c
Author: Colin Walters <walters@verbum.org>
Date: 2009-01-30 16:03:11 -0500

Bug 19259: Always use /etc/bash_completion.d

This looks like what bash upstream has standardized on.

```
dbus/Makefile.am | 4 +++-  
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 78eb504bd9954918c363f357d0ab71ca85e3fc23
Author: Colin Walters <walters@verbum.org>
Date: 2009-01-27 18:59:01 -0500

Bug 19647: Move test-types inside run-test to avoid dep on existing session bus

All tests that use DBUS_BUS_SESSION should be in run-test.sh to avoid introducing a dependency on an existing session bus.

```
test/core/Makefile.am |      2 +-
test/core/run-test.sh |      1 +
2 files changed, 2 insertions(+), 1 deletion(-)
```

commit 3d25c1640d59e9770c33fafb1a12766feb6494e8
Merge: d92a441 8868486
Author: Colin Walters <walters@verbum.org>
Date: 2009-01-27 18:43:50 -0500

Merge branch 'master' of
ssh://walters@git.freedesktop.org/git/dbus/dbus-glib

commit d92a44109e3fdc766e34b53f7ec5329e98e13909
Author: Colin Walters <walters@verbum.org>
Date: 2009-01-27 17:00:37 -0500

Bug 19441: Don't send replies for messages explicitly not requesting one

In sending a reply when a message has the dbus_message_set_no_reply flag set, we can cause spurious denials logged on the system bus, aside from being inefficient.

```
dbus/dbus-gobject.c |      67
+++++-----
1 file changed, 47 insertions(+), 20 deletions(-)
```

commit 88684866d06df99f89b30a07e4809369b4cb0453
Author: Colin Walters <walters@verbum.org>
Date: 2008-12-19 11:18:26 -0500

Add new files missed from last commit

```
test/core/my-object-subclass.c | 115
+++++
test/core/my-object-subclass.h | 33 ++++++
test/core/test-service-glib-subclass.xml | 8 +++
3 files changed, 156 insertions(+)
```

commit d55ffa5bbccd027caf7f2e7376b555e397a7ac3c

Author: Dan Williams <dcbw@redhat.com>
Date: 2008-12-18 13:09:33 -0500

Bug 19145: test cases for GetAll

Add test cases for GetAll.

Signed-off-by: Colin Walters <walters@verbum.org>

```
test/core/Makefile.am          |    11 +-  
test/core/test-dbus-glib.c     |   187  
+++++  
test/core/test-service-glib.c  |     7 ++  
test/core/test-service-glib.xml |     2 +  
4 files changed, 205 insertions(+), 2 deletions(-)
```

commit 94d68f00d9d244de3b1d66d3cf78bb5171552311
Author: Tambet Ingo <tambet@gmail.com>
Date: 2008-12-18 12:09:43 +0200

Use the provided interface for
org.freedesktop.DBus.Properties.GetAll
call.

```
dbus/dbus-gobject.c           |     82  
+++++-----  
1 file changed, 68 insertions(+), 14 deletions(-)
```

commit 1b150c32aa574b421161b63bb01eeeb5ad23bf3a
Author: Frederic Crozat <fcrozat@mandriva.com>
Date: 2008-12-05 11:16:46 -0500

Fix linking order so -Wl,--as-needed works

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/Makefile.am              |     2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit f4a6c4755d7a82bb7ec0a8bf472bfe091c310ef0
Author: Colin Walters <walters@verbum.org>
Date: 2008-12-04 16:12:18 -0500

Do NEWS for 0.78

Should have been in earlier commit, oops.

```
NEWS |    18 +++++  
1 file changed, 18 insertions(+)
```

commit 3a713710a34d75d174eb632a6444e87dbf3923f4
Author: Colin Walters <walters@verbum.org>

Date: 2008-12-04 16:02:12 -0500

Bump configure for unstable cycle

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit beaff4e7c8e6ef601a00c03a4bed533becd273be
Author: Colin Walters <walters@verbum.org>
Date: 2008-12-04 15:58:30 -0500

Release 0.78.

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit d77506652b555b147e79889d6163208535e1cd74
Author: Colin Walters <walters@verbum.org>
Date: 2008-12-04 15:58:12 -0500

Clean bash completion file

dbus/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit 6ba71576f4417bab46ca666fa44dda52dee95e88
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-07-31 12:28:07 -0400

add bash completion for dbus-send(1)

For now, it's in dbus-glib since dbus doesn't have an introspection XML parser (yet).

Signed-off-by: Colin Walters <walters@verbum.org>

configure.ac | 13 +
dbus/Makefile.am | 17 +-
dbus/dbus-bash-completion-helper.c | 513
++++
dbus/dbus-bash-completion.sh.in | 21 ++
4 files changed, 563 insertions(+), 1 deletion(-)

commit 48738e529dc7a0f27d185f2e79174b0e369fd0af
Author: Tomas Pelka <tpelka@redhat.com>
Date: 2008-12-04 15:46:31 -0500

Bug 18497 - Correct type in test case

Changing to INT32 matches the sending side.

Signed-off-by: Colin Walters <walters@verbum.org>

test/core/test-thread-server.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit aebec4bd82c023502600786793ba0ce0c7323fbe
Author: Colin Walters <walters@verbum.org>
Date: 2008-12-04 15:42:08 -0500

Bug 18698 - Use AC_SEARCH_LIBS to avoid unneeded dep on nsl

Also changed the socket search to be the same way.

configure.ac | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit f10ee141185f9501c852acaf557b100abafc60fb
Author: Nick Welch <nick@incise.org>
Date: 2008-12-04 15:28:45 -0500

Typo fix in docs

Signed-off-by: Colin Walters <walters@verbum.org>

dbus/dbus-gobject.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit c0bba240c4beb6ecfd71aefa184011e8adc27f5f
Author: Robert McQueen <robert.mcqueen@collabora.co.uk>
Date: 2008-09-23 23:42:02 +0300

Bug #17329: allow hash tables to contain complex types

Previously, dbus-glib has provided destroy functions for the keys and values when constructing hash tables, so that any hash tables it constructed could be entirely freed (along with their contents) by destroying/unreffing. Unfortunately this meant that any "complex" types, where you need to know the GType in order to free them, were not allowed in hash tables. In real terms, this was anything which dbus-glib marshalled to a GPtrArray, so any array of arrays, variants, structures, object paths, or other boxed types were not allowed as hash values.

This patch allows a broader range of types to be used as the values in hash tables, including those where no simple free function is available. Instead of relying on the key/value destroy functions, the new

hash_free
function uses g_hash_table_foreach_steal to remove the keys and
values
pairwise and free them when the type is known.

Unfortunately, it's part of the API assumptions that hash tables
which
were produced through the current API had valid free functions,
and
particularly that if the hash table was refed by the application,
that
the keys/values would persist beyond when dbus-glib had unrefed
it, and
be freed when the hash table was later destroyed. So it's not
sufficient
to use only this new freeing method on all hash tables from now on
- we
have to behave in the old way for all of the previously allowable
types
(including any hash tables which contain any hash tables which
were
freeable in the old way).

However, as these new hash tables contain values which will not
be freed
if you manipulate the hash table directly (removing or replacing
keys,
or destroying or unrefing it directly), and g_boxed_free should
be used
instead, a false free function is provided to print a critical
warning
for the developer in the case that memory would be leaked.

```
dbus/dbus-gvalue-utils.c | 151  
+++++-----  
test/core/test-dbus-glib.c | 37 +++++-----  
test/core/test-service-glib.xml | 4 +-  
3 files changed, 168 insertions(+), 24 deletions(-)
```

commit 143b09b42d73451dcc3b835f76f8d43da786342a
Author: Robert McQueen <robert.mcqueen@collabora.co.uk>
Date: 2008-09-22 21:40:48 +0100

Bug #17798: add support for 'o', 'g' and 'as' in dictionaries

This teaches the parameterised hash table about how to hash,
compare and
free object paths and signatures, allowing them to be used as
hash keys
and values, and also how to free strvs, so they can be used as
values.

Adds some simple test methods which echo a{gas} and a{oas}

dictionaries
back immediately and compare the results.

```
dbus/dbus-gvalue-utils.c | 29 ++++++++
test/core/my-object.c    | 16 ++++++
test/core/my-object.h    | 4 ++
test/core/test-dbus-glib.c | 96
+++++++
test/core/test-service-glib.xml | 12 +++++
5 files changed, 157 insertions(+)
```

commit e4c917189dc9478c261c94703b311e1579292f22
Author: Robert McQueen <robert.mcqueen@collabora.co.uk>
Date: 2008-09-14 21:16:10 +0100

Bug #17797: add support for 'g' type, as a boxed char *

Adds DBUS_G_TYPE_SIGNATURE, which is a boxed type just like
DBUS_G_TYPE_OBJECT_PATH.

```
dbus/dbus-binding-tool-glib.c | 4 ++
dbus/dbus-glib.h              | 3 ++
dbus/dbus-gmain.c             | 5 ++-
dbus/dbus-gsignature.c        | 2 +
dbus/dbus-gvalue.c            | 82
+++++++
test/core/my-object.c         | 7 ++++
test/core/my-object.h         | 1 +
test/core/test-dbus-glib.c    | 19 ++++++
test/core/test-service-glib.xml | 5 +++
9 files changed, 126 insertions(+), 2 deletions(-)
```

commit f89b8de02f21891f430a2e370c391238b575403a
Author: Robert McQueen <robert.mcqueen@collabora.co.uk>
Date: 2008-09-23 23:45:14 +0300

Bug #17795: depend on Glib 2.10 for hash unref consistency

Commit a8bf32 (for bug #11396) introduced a compile-time version
check
for Glib 2.10 which changed whether hash_simple_free called
g_hash_table_unref or g_hash_table_destroy. This is at best
useless
for
people who actually wanted to rely on _unref being used instead of
_destroy, because dbus-glib can legitimately be built against
older
versions of Glib than that in use by the application itself, and
at
worst harmful as applications relying on the _unref semantics
would have

their assumptions violated on platforms where this was the case.
Given
in /most/ cases now, Glib 2.10 is readily available, and we were
in this
kinda vague state where people could've used `_unref` and gotten
away with
it, I'm going to say that's what our ABI is now, and stick to
it. People
can depend on the next version of `dbus-glib` if they rely on
`_unref`.

```
configure.ac | 2 +-  
dbus/dbus-gvalue-utils.c | 4 ----  
2 files changed, 1 insertion(+), 5 deletions(-)
```

commit 24b94fb83c0d22cf2803af692dd8ccbc485d890c
Author: Robert McQueen <robert.mcqueen@collabora.co.uk>
Date: 2008-09-22 21:25:03 +0100

Bug #17794: make `run-test.sh` fail when `die` is called

Missing `die()` function meant this wasn't returning non-zero even
when
tests were failing. Oops.

```
test/core/run-test.sh | 6 ++++++  
1 file changed, 6 insertions(+)
```

commit 3ca014ea6b55cd843f1c80b04a4e0ec41af6210e
Author: Colin Walters <walters@verbum.org>
Date: 2008-11-17 18:30:32 -0500

Bug 18573 - Fix race when service appears and disappears quickly

We can't assert that a `for_name_owner` proxy is associated when
we disassociate; if a service appears and disappears quickly, we
won't be associated yet. So remove the assertion.
Correspondingly we need to cancel any inflight `GetNameOwner` call
when disassociating proxies to avoid leaking.

```
dbus/dbus-gproxy.c | 11 ++++++++--  
1 file changed, 9 insertions(+), 2 deletions(-)
```

commit 0a883a523f8d133f71271276b755162c56480021
Author: Philip Van Hoof <pvanhoof@gnome.org>
Date: 2008-09-18 18:22:59 -0400

Bug 17614: Use `g_slice` for allocating temporary async call data

Using `g_slice` is going to be faster.

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-binding-tool-glib.c | 10 ++++++--
1 file changed, 8 insertions(+), 2 deletions(-)
```

```
commit 4c7e60e3ff4148a181a496f85b491e8cb39d7e46
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-20 14:08:47 -0400
```

Bug 10373: Use of deprecated API (was used incorrectly)

```
* dbus/dbus-gvalue.c: The use of dbus_message_iter_get_array_len
here is incorrect; we don't want to allocate a number of pointers
based on the length of the strings. Instead, we now use the
regular
GArray type which handles reallocating as size increases
appropriately.
```

```
dbus/dbus-gvalue.c | 25 ++++++-----
1 file changed, 11 insertions(+), 14 deletions(-)
```

```
commit 2514b84b3644223653018cd75d0d6e1459a7ab70
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-19 12:47:16 -0400
```

Bug 16419: stack overflow demarshaling recursive variants

```
* dbus/dbus-gvalue.h: Add a recursion_depth member.
* dbus/dbus-gvalue.c: Keep track of recursion depth in
_dbus_gvalue_demarshal, cut it off at a default
(right now 32).
* dbus/dbus-gobject.c: Initialize recursion depth.
* dbus/dbus-gproxy.c: Ditto.
* test/core/test-variant-recursion.c: Test the
variant recursion case, make sure the remote site
will throw an error.
* test/core/Makefile.am: Add test-variant-recursion.
```

```
dbus/dbus-gobject.c | 2 +
dbus/dbus-gproxy.c | 2 +
dbus/dbus-gvalue.c | 21 ++++++--
dbus/dbus-gvalue.h | 1 +
test/core/Makefile.am | 6 +-
test/core/run-test.sh | 1 +
test/core/test-variant-recursion.c | 89
+++++
7 files changed, 119 insertions(+), 3 deletions(-)
```

```
commit d1b80d803a0268bd4b3dd5b9a9522230461f2947
Author: Dan Williams <dcbw@redhat.com>
Date: 2008-06-05 17:57:53 -0400
```

Bug 16114 [patch] wincaps-to-uscore property names for GetAll()

* dbus/dbus-gobject.c: We need to uscore property names so that we actually find the right properties.

dbus/dbus-gobject.c | 10 ++++++--
1 file changed, 8 insertions(+), 2 deletions(-)

commit 8054025ae498cc586676fd0ed6573828dd0de4a0
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 16:05:39 -0400

One final large HACKING update

HACKING | 32 ++++++-----
1 file changed, 29 insertions(+), 3 deletions(-)

commit d80bc7686a0e5cc7d45d940e8404ac4c8742b8f5
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 15:40:35 -0400

Pacify gtk-doc by adding new files

* doc/reference/dbus-glib-undeclared.txt:
* doc/reference/dbus-glib.types: New files
without which gtk-doc complains.

0 files changed

commit d91cf40b83f7900899fe89cc83a75c82cd4bf833
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 15:37:16 -0400

Another HACKING tweak

HACKING | 6 +++--
1 file changed, 4 insertions(+), 2 deletions(-)

commit e4b73986d1b2d6f9badda4656e57023b2c921ff7
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 15:37:02 -0400

Bump for development

configure.ac | 2 +-
doc/reference/dbus-glib-undocumented.txt | 20 +++-----
2 files changed, 4 insertions(+), 18 deletions(-)

commit 96f144d25c7f3d573684bd50f87d912990f3c589
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 15:09:02 -0400

Release 0.76

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 1517f7f395527519402f8b37988de9d4ea24dbc5
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 15:08:40 -0400

Document release process

HACKING | 23 ++++++-----
1 file changed, 22 insertions(+), 1 deletion(-)

commit e0dcd92e33e566f9852df94036e74f48af795ea3
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 14:55:15 -0400

Update HACKING

HACKING | 13 ++++++----
1 file changed, 10 insertions(+), 3 deletions(-)

commit 4f776e9db9f299dab96a12a00a6279b6977f6c32
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 14:48:18 -0400

Update gtk-doc

doc/reference/dbus-glib-undocumented.txt | 20 ++++++-----
1 file changed, 17 insertions(+), 3 deletions(-)

commit fa46dc8d35de877b8bdcb53f9508252249138bd6
Author: Colin Walters <walters@verbum.org>
Date: 2008-06-05 14:40:09 -0400

Update NEWS

NEWS | 25 ++++++-----
1 file changed, 25 insertions(+)

commit acd44d2ae3069667dbbb8b9c9c8b9c2afcad96d9
Author: Christian Persch <chpe@gnome.org>
Date: 2008-06-03 18:01:54 -0400

Bug 16217: generated bindings don't build with
-DG_DISABLE_SINGLE_INCLUDES

* dbus/dbus-binding-tool-glib.c: Only include glib.h
because
individual includes are disallowed.

Signed-off-by: Colin Walters <walters@verbum.org>

```
dbus/dbus-binding-tool-glib.c |    3 +--
1 file changed, 1 insertion(+), 2 deletions(-)
```

```
commit 1744bf88024756d17f867ab9b86aca25e2b77e64
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-28 14:11:01 -0400
```

Remove GConnection tests that assert, since we abort on those

```
test/core/test-types.c |   26 -----
1 file changed, 26 deletions(-)
```

```
commit a94e11973467e1b2cc43bale77b7413655b1dff7
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-28 00:25:57 -0400
```

Bug 11671: Don't use well known name in /tmp

This as a side effect fixes the test suite on Solaris.

```
test/core/peer-client.c |   13 ++++++++
test/core/peer-server.c |   26 ++++++++-----
test/core/run-peer-test.sh |    9 +-----
3 files changed, 29 insertions(+), 19 deletions(-)
```

```
commit dceee795d0cdc5889701064ac38d3c64451b02ec
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 23:19:52 -0400
```

From OpenSUSE: Add return statement in case assertions not enabled

```
dbus/dbus-gvalue-utils.c |    1 +
1 file changed, 1 insertion(+)
```

```
commit 681f3ae573b2e00dad5c1193a044a994fc50cf38
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 23:08:59 -0400
```

Bug 13060: Remove NameOwnerChange matches on unref (Kimmo Härmäläinen)

```
dbus/dbus-gproxy.c |  14 ++++++++
1 file changed, 14 insertions(+)
```

```
commit eef148bdc0f602b0fba09a851c1ca6b55b8bb71c
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 23:01:39 -0400
```

Bug 15733: Cosmetic fix for removing redundant declarations

```
dbus/dbus-gtype-specialized.h |    2 --
```

1 file changed, 2 deletions(-)

commit d043342c41535e610268b9bb2a45143bad0b597d
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 22:56:00 -0400

Bug 12505: Avoid receiving duplicate NameOwnerChanged (Dan Williams)

dbus/dbus-gproxy.c | 13 ++++++++--
1 file changed, 11 insertions(+), 2 deletions(-)

commit 43db9baa4cd0921d2ee830185ab46b4646b4e73b
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 16:49:26 -0400

Bug 10834: Fix error handling path for
dbus_g_proxy_end_call_internal

This patch was based initial work by Peter Kjellerstedt.

This patch made obvious the need to correctly handle type mismatches in demarshal_basic, similarly to what the other demarshalers are doing.

Also add some tests for error handling.

dbus/dbus-gproxy.c | 25 ++++++++-----
dbus/dbus-gvalue.c | 28 ++++++++
test/core/test-dbus-glib.c | 29 ++++++++
3 files changed, 77 insertions(+), 5 deletions(-)

commit dc3bfd517a3bd0e28f4fc2d030ac38bdc848eb86
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 16:36:58 -0400

Fix some test suite memory leaks

test/core/test-dbus-glib.c | 3 +++
1 file changed, 3 insertions(+)

commit 167ea555cf3a763d148a69d0af9453b83d70d107
Author: Julien Danjou <julien@danjou.info>
Date: 2008-04-24 14:03:42 +0200

Fix typo in --help print strings

Signed-off-by: Julien Danjou <julien@danjou.info>
Signed-off-by: Colin Walters <walters@verbum.org>

dbus/dbus-glib-tool.c | 4 +--
1 file changed, 2 insertions(+), 2 deletions(-)

commit c057283a86230904552d5c51623bbce5ab0c9770
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 15:15:35 -0400

Bug 10244: Fix error handling case in parser (William Jon McCann)

We need to return FALSE when handling an error.

dbus/dbus-gparser.c | 11 ++++++----
1 file changed, 7 insertions(+), 4 deletions(-)

commit 361dc90eaa269325ddc5dae0003calle1d7b811e
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 14:57:20 -0400

Bug 9867: Respect ACLOCAL_FLAGS (Kalle Vahlman)

autogen.sh | 1 +
1 file changed, 1 insertion(+)

commit a8bf32ab8b0e30e0c74e07c58e9bc79a448683b2
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 14:31:58 -0400

Bug 11396: Use g_hash_table_unref if available (Marco Barisione)

This lets users ref hashes with g_hash_table_ref.

dbus/dbus-gvalue-utils.c | 4 +++
1 file changed, 4 insertions(+)

commit ad35bf13f93d18b0b0e8f930ff79af9dcc1c8508
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 14:18:49 -0400

Bug 12675: Handle disconnected connections in calls (Kimmo
Hämäläinen)

dbus/dbus-gproxy.c | 22 ++++++-----
1 file changed, 19 insertions(+), 3 deletions(-)

commit f110cd3aa0532ab6cdfblee9a19f839700fe32e6
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 13:57:40 -0400

Bug 12857: Balance va_start/va_end in error case (Kimmo
Hämäläinen)

dbus/dbus-gparser.c | 6 ++++-
1 file changed, 5 insertions(+), 1 deletion(-)

commit d20c90aab3b0eca8715deaa59877f35a7d4e345e
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 13:54:11 -0400

Bug 12849: Fix use after free (Kimmo HÃ¤mÃ¤lÃ¤inen)

dbus/dbus-gvalue.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit fde77bff42fac9ed95869a69acff4f9daab23e04
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 13:35:20 -0400

Bug 16079: Return an error on unknown property

https://bugs.freedesktop.org/show_bug.cgi?id=16079

Previously we just asserted, not very useful.

dbus/dbus-gobject.c | 6 +++++
test/core/test-dbus-glib.c | 62
+++++
2 files changed, 67 insertions(+), 1 deletion(-)

commit b26f1887f6907d4c9aldac013649f28950c6b2d3
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 13:34:55 -0400

Support for monitoring conversation during tests

test/core/run-test.sh | 3 +++
1 file changed, 3 insertions(+)

commit b783ce802db01a5370ce70514755e1885f6e6eb1
Author: Colin Walters <walters@verbum.org>
Date: 2008-05-27 12:39:44 -0400

Also make CRITICAL messages fatal in tests.

test/core/test-dbus-glib.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 8fe656de630e851bc512bbd7e7a6c18c53aecb61
Author: Simon McVittie <simon.mcvittie@collabora.co.uk>
Date: 2008-04-14 12:56:43 +0100

Freeze error ABI at the ABI used in Fedora 8 and Ubuntu gutsy.

This avoids getting a different ABI depending on the version of
libdbus
we're compiled against. fd.o #15430, Debian #476080.

```
dbus/dbus-glib.h | 34 ++++++-----
dbus/dbus-gobject.c | 72
+++++-----
dbus/make-dbus-glib-error-enum.sh | 25 +-----
dbus/make-dbus-glib-error-switch.sh | 29 +-----
4 files changed, 106 insertions(+), 54 deletions(-)
```

commit 7fccdc8386d97d8c7e963800b1fbedd47f72b66b
Author: Colin Walters <walters@space-ghost.verbum.private>
Date: 2008-03-31 11:23:29 -0400

Add some docs to dbus_set_g_error.

```
dbus/dbus-gobject.c | 17 ++++++-----
1 file changed, 16 insertions(+), 1 deletion(-)
```

commit 34042c4cf2fcf18e101d07e495f9fa5dda99aa58
Merge: df7c2b7 bd53ac9
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-03-17 13:11:45 -0400

Merge branch 'master' of
ssh://david@git.freedesktop.org/git/dbus/dbus-glib

commit df7c2b783ed92f33a1bedcba12a9c30fc34ba8d5
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-03-17 13:10:47 -0400

add some design advice to dbus_g_proxy_set_default_timeout()
... as requested on the mailing list.

```
dbus/dbus-gproxy.c | 7 ++++++
1 file changed, 7 insertions(+)
```

commit bd53ac9f7ef9a6c2c9d1d12af382b1a8a10e9dba
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2008-03-17 15:10:43 +0100

allow namespaced extentions to introspection XML

This allows us to stick namespaced documentation stuff into the
introspection
XML and have dbus-glib not puke on it.

Patch credit goes to Colin Walters.

```
dbus/dbus-glib-tool.c | 2 +-
dbus/dbus-gparser.c | 40 ++++++-----
2 files changed, 30 insertions(+), 12 deletions(-)
```

commit cb732d38f78dfb7c1ef73179e798fa4f42fd962e

Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2008-03-17 14:26:50 +0100

bump version and dbus dependancy

Bumps version to 0.75. Bumps dbus dependancy to 1.1.

```
configure.ac |      4 ++--  
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 09b0fc5818812d0167243bae9fd52cdaf67f0af0
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-03-15 16:51:48 -0400

Export the recently added GetAll() method on
org.freedesktop.DBus.Properties

Because round-trip city is a bad place.

```
dbus/dbus-gobject.c |      220  
+++++-----  
1 file changed, 177 insertions(+), 43 deletions(-)
```

commit bf0c9b3d6adc95863d5b5a4ce6ca994fd7fdc137
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-03-15 15:32:56 -0400

Add new function to specify the default timeout for a proxy

Without a function like this the generated client glue code is
unusable for D-Bus methods that take a long time to complete (such
as
disk operations like mkfs and partitioning).

Also add some missing `_with_timeout` functions on DBusGProxy to the
gtk
docs.

```
dbus/dbus-glib.h |      3 +++  
dbus/dbus-gproxy.c |     35  
+++++-----  
doc/reference/dbus-glib-sections.txt |      3 +++  
doc/reference/dbus-glib-undocumented.txt |    15 ++++++----  
4 files changed, 50 insertions(+), 6 deletions(-)
```

commit 1fa4129f67208e24f6ae5f9ab38a3ee26c521f87
Author: David Zeuthen <davidz@redhat.com>
Date: 2008-03-15 15:21:06 -0400

Bump version to 0.75

```
configure.ac |      2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

1 file changed, 1 insertion(+), 1 deletion(-)

commit 1124256e1a87291101a6c287c1248c2f7276b993
Author: Ross Burton <ross@burtonini.com>
Date: 2008-02-27 14:27:49 +0000

Fix incorrect assign in test suite

We were assigning a DBusConnection* to a DBusGConnection*, which is bad.

test/core/test-types.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit eea8f7247e1cbeb8c00fe316a37520b98acd55dc
Author: Ross Burton <ross@burtonini.com>
Date: 2008-02-27 14:22:16 +0000

Use dbus_watch_get_unix_fd not dbus_watch_get_fd

The latter is deprecated, so don't use it.

dbus/dbus-gmain.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 2cf62d7ff7d3a7bc450d0b60bb81a8365ffd310b
Author: Ross Burton <ross@burtonini.com>
Date: 2008-02-27 14:19:48 +0000

Fix pending call cancelling in proxy dispose

The dispose treated the hash values as DBusGProxyCall objects, but they are DBusPendingCall (thanks Dafyd Harries).

dbus/dbus-gproxy.c | 11 ++++++-----
1 file changed, 6 insertions(+), 5 deletions(-)

commit 8e024ae2252e6e948c28203f07aa416df3dac0b8
Author: Ross Burton <ross@burtonini.com>
Date: 2008-02-27 14:02:36 +0000

Unref the connection and message on dbus_g_return_error

dbus/dbus-gobject.c | 3 +++
1 file changed, 3 insertions(+)

commit 98423ae4e6b85741ce15f097652620886a49c375
Author: Ross Burton <ross@burtonini.com>
Date: 2007-11-13 17:56:37 +0000

Allow daemondir to be configured (#10668)

The daemon can be at any location, and may not be on the path.
Fetch the path
from the pkg-config file and use it if it is set.

Thanks to Brian Cameron for this patch.

```
configure.ac      | 9 ++++++++
tools/Makefile.am | 2 +-
2 files changed, 10 insertions(+), 1 deletion(-)
```

```
commit 33b7a7f05372baaaf95d0e1c2c3b758321e4b0c5
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:52:07 +0100
```

Update ignores

Add the new tests and *~ to the ignore list.

```
.gitignore | 5 ++++
1 file changed, 4 insertions(+), 1 deletion(-)
```

```
commit 4b2cf9d75f6c315071adc925ec56e3bac3423730
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:49:49 +0100
```

Fix bashism

"function" is a bashism, so don't use it.

```
tools/run-with-tmp-session-bus.sh | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 3f6e2c0c76d3643a1823b5ea7c8f5486a6b448de
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:37:47 +0100
```

Fix broken introspection XML

When writing XML for properties a duplicate closing tag was
written. (#8607,
thanks William Jon McCann).

```
dbus/dbus-gobject.c | 2 --
1 file changed, 2 deletions(-)
```

```
commit c08a7f910c4c746ba790375be2d0f0682471cf06
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:34:07 +0100
```

Fix build with non-gcc compilers

AIX's compiler and some non-c99 compilers are braindead, massage the code to work with them (#11675, thanks Peter O'Gorman).

```
dbus/dbus-gtype-specialized.c | 4 +++-
dbus/examples/statemachine/statemachine.c | 2 +-
2 files changed, 4 insertions(+), 2 deletions(-)
```

commit b2bcb0f3fdf8eaec9257f48dd0a0e2ca03220775
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:17:12 +0100

Fix bashisms

The test suite used /bin/bash for no good reason, remove all bashisms.
(#11672,
thanks Peter O'Gorman).

```
test/core/run-peer-test.sh | 4 +++-
test/core/run-test.sh | 18 ++++++-----
test/interfaces/run-test.sh | 14 ++++++-----
tools/run-with-tmp-session-bus.sh | 9 +++++----
4 files changed, 23 insertions(+), 22 deletions(-)
```

commit 377831cc8496bf2b3f96e96a44ff4dfc1a6c690e
Author: Ross Burton <ross@burtonini.com>
Date: 2007-09-11 10:11:12 +0100

Fix build with Glib 2.6

configure.ac checks for glib 2.6, but the test suite used g_intern_static_string which was added in 2.10. This removes the use of that function because it isn't required (#11674, thanks Peter O'Gorman).

```
test/interfaces/test-interfaces.c | 4 +++-
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 2929a9d676b9961abff6c6acb1b67431f7d62ab1
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-06-27 11:49:02 +0100

Update version in configure.ac

Update version to 0.74 in configure.ac

```
configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit de33a575aa58867642bcfd8bb8cb2efc631d9f1d

Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-06-27 11:39:08 +0100

Update NEWS

Update NEWS from git history.

NEWS | 20 ++++++++-----
1 file changed, 14 insertions(+), 6 deletions(-)

commit a732cbc211b562660c5a17c898ad032fb8c041fd
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-27 11:05:03 +0100

Sleep after starting the peer server, before starting the peer client.

This fixes random failures due to the race.

test/core/run-peer-test.sh | 4 +--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 4335b1204a483ffa5c5a72d58b585f77827d852d
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-27 10:58:03 +0100

Update NEWS for pending release.

NEWS | 12 ++++++++
1 file changed, 12 insertions(+)

commit 0d77d83b7afd5837d4427b7cdc5940f54625b65c
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-06-22 17:15:34 +0100

make test/core/run-peer-test.sh executable

0 files changed

commit 0ba73bf7cf1a2770b7aac8d57e61e755aca45b4d
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:46:37 +0100

Add missing include.

test/core/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit bf647c411b42064fe48877c30803cbe10d751204
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:45:57 +0100

Install the type info before creating signals

This fixes the type system warnings from peer-server, because the types were used (when creating signals) before they were created. Installing type info initialises the types.

```
test/core/my-object.c | 6 +++---
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
commit a3a6efda1117633cc8a5525811f6a3ba41f2d31b
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:28:48 +0100
```

Init threading first to stop a warning from new GLib.

```
test/core/peer-client.c | 2 +-
test/core/peer-server.c | 2 +-
2 files changed, 2 insertions(+), 2 deletions(-)
```

```
commit 36ed523539074a84cd990a8513b5d02a6223c44c
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:27:18 +0100
```

Make test script executable

```
0 files changed
```

```
commit 514fd82183c214e3afd4cca630a20ff24b62e482
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:25:14 +0100
```

Remove unused variable.

```
test/core/run-peer-test.sh | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit d428aa4edf426f662e89f2b08a377b0dd481f3ac
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 15:15:37 +0100
```

Remove the XML documentation support in configure

It isn't used and is legacy from when this was part of dbus itself.

```
configure.ac | 34 +-----
1 file changed, 1 insertion(+), 33 deletions(-)
```

```
commit 1f385f6b10799cdd0bf57d69d5212ac16ac17266
Author: Ross Burton <ross@burtonini.com>
```

Date: 2007-06-22 15:02:24 +0100

Fix typo in _dbus_gvalue_signals_error (#10837)

It was using g_value_get_boxed() when looking at a object-holding value. Thanks
to Peter Kjellerstedt for noticing this.

```
dbus/dbus-gvalue-utils.c | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 68f03928aaac849dd12e38c01e448b569f578976
Author: Ross Burton <ross@burtonini.com>
Date: 2007-06-22 14:41:36 +0100

Update GLib requirement (#10889).

```
INSTALL | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 58cacc6872e1b8cf365ac056775498b9dede71b4
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-04-16 08:23:17 -0700

Document dbus-gtype-specialized

Documents all the functions in dbus-gtype-specialized.c and adds an overview of the functionality.

```
dbus/dbus-gtype-specialized.c | 312  
+++++++-----  
dbus/dbus-gtype-specialized.h | 14 +-  
doc/reference/dbus-glib-sections.txt | 2 +-  
doc/reference/dbus-glib-undocumented.txt | 38 +---  
4 files changed, 304 insertions(+), 62 deletions(-)
```

commit 26d8584c04c05be29ae35acc3a2e8f573cc79206
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-30 18:12:11 +0100

Add simple test suite for peer objects.

Add a simple test suite for peer proxies. This involved refactoring the MyObject class into its own file so that it can be used by multiple tools. Also added is a test suite for dbus_connection_get_g_connection.

```
test/core/Makefile.am | 26 +-  
1 file changed, 26 insertions(+), 0 deletions(-)
```

```

test/core/my-object.c | 737
+++++
test/core/my-object.h | 104 +++++
test/core/peer-client.c | 127 +++++
test/core/peer-server.c | 55 +++
test/core/run-peer-test.sh | 9 +
test/core/test-service-glib.c | 832
+-----
test/core/test-types.c | 74 +++
8 files changed, 1129 insertions(+), 835 deletions(-)

```

```

commit af91f5e0e2b5a20adf124707356b12da0c2f1e00
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-30 18:05:40 +0100

```

Support peer-to-peer proxies.

Previously DBus-GLib didn't support peer proxies despite having API for them. This patch stops dbus-glib from crashing when you use a peer signal (#10233).

```

dbus/dbus-gproxy.c | 35 ++++++-----
1 file changed, 18 insertions(+), 17 deletions(-)

```

```

commit 25f056826336d4385b5488706f0flecc0843bf2f
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-30 17:54:58 +0100

```

Add `dbus_connection_get_g_connection`.

Add `dbus_connection_get_g_connection`, which is useful when you have a `DBusConnection` which was originally a `DBusGConnection`, and want it back again.

```

dbus/dbus-glib-lowlevel.h | 1 +
dbus/dbus-glib.c | 24 ++++++
dbus/dbus-gmain.c | 12 +++++-----
3 files changed, 31 insertions(+), 6 deletions(-)

```

```

commit 0c46ca1a20eb8ba676ea2f1e829d039d5e3c383d
Author: Ross Burton <ross@blackadder.(none)>
Date: 2007-03-22 15:04:14 +0000

```

Stop compiler warnings (#10374).

```

dbus/dbus-gthread.c | 1 +
test/interfaces/test-server.c | 2 ++

```

2 files changed, 3 insertions(+)

commit 72e9f061668a28cbbbeae2400290e1993c5d60d3c
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-28 16:32:00 +0100

Fix error handling

dbus/dbus-gproxy.c | 9 ++++-----
1 file changed, 4 insertions(+), 5 deletions(-)

commit 9ca04c94ed6ddc2ff7459ce2031740704ec67ff2
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-22 15:38:03 +0000

Handle dbus errors which are not name has no owner

dbus/dbus-gproxy.c | 3 +++
1 file changed, 3 insertions(+)

commit 85b269cc7ca221ee4fe0235154e783b3640ee62d
Author: Ross Burton <ross@blackadder.(none)>
Date: 2007-03-22 15:05:02 +0000

Update abstract socket test from dbus

The abstract socket test doesn't cross compile at all, updating
the
test from
the original copy in Dbus.

configure.ac | 17 ++++++++-----
1 file changed, 9 insertions(+), 8 deletions(-)

commit a64b26cdde04e624b2bb42ccadd3fc252bf05667
Author: Ross Burton <ross@burtonini.com>
Date: 2007-03-22 15:10:47 +0000

Rename the error quark.

The error quark was copied from GLib, renaming it to be unique.

dbus/dbus-glib.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 5ca3418e8e00f62e3e60578e3951b51cd158f8dc
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-03-29 12:44:01 +0100

Update AUTHORS

Correct spelling of S.Nallammai, add Ross Burton, who's been missing here for quite some time.

AUTHORS | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit 7fbd3a1c49614e71a9ff2b3d6e43bf380527301b
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-03-29 12:32:46 +0100

Update AUTHORS

I had failed to update AUTHORS in the last couple of releases, this commit makesit current.

AUTHORS | 6 ++++++
1 file changed, 6 insertions(+)

commit f78b90659f32474b7ca56925ddebd88af7f06c86
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-03-29 12:20:30 +0100

Update .gitignore

Ignore autoconf errors, autom4te.cache, some new generated files.

.gitignore | 7 ++++++-
1 file changed, 6 insertions(+), 1 deletion(-)

commit 138e4ce76ae53b86488e12c43db511a838be4fc9
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-13 14:16:22 +0000

Do libtool versioning.

As we're going to start behaving like a proper library, update libtool versionsing appropriately for our added api.

configure.ac | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit d19c5c0eac749704af70edffa1fa0ac70eb9373
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-13 13:58:38 +0000

Update NEWS

Releasing today..

NEWS | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit e368483363fc920cf2978180f3a9fe6687465807
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-12 04:23:42 +0000

Update NEWS

Update NEWS, adding fix for bug #9769.

NEWS | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit 814b029d9291199bc9fdd2998c429ee15f243a93
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-12 04:21:50 +0000

Add pkg-config support for uninstalled use.

Adds dbus-glib-1-uninstalled.pc for use in build environments
where
dbus-glib is used as an uninstalled build dependency. Fix due to
Damien Carbery <damien.carbery@sun.com>.
Fixes bug #9769.

configure.ac | 1 +
dbus-glib-1-uninstalled.pc.in | 12 ++++++++
2 files changed, 13 insertions(+)

commit 6bcb544eb6bc164e367a0e849b6c41274809401e
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-09 02:49:57 +0000

Update version in configure.ac

Updates version 0.73 in configure.ac

configure.ac | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit dba710006cb0b25454989f41babc53c12d3fc901
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-09 02:46:57 +0000

Update NEWS ready for 0.73 release

Updates NEWS with all the changes for 0.73

NEWS | 26 ++++++++
1 file changed, 26 insertions(+)

commit 133bc8003657169f622d4340360a0e5b15a455de
Author: Rob Taylor <rob.taylor@codethink.co.uk>

Date: 2007-02-09 02:21:20 +0000

Allow passing of NULL to strv out arguments.

A nicety for the user, allows passing NULL to strv out arguments as

shorthand for an empty array.

Patch due to Luiz Augusto von Dentz <luiz.dentz@gmail.com>

Fixes bug #8795.

dbus/dbus-gvalue.c | 20 ++++++++-----
1 file changed, 13 insertions(+), 7 deletions(-)

commit 5f70d7b59445b634f94469a8c12cf31f8f81fdae
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-09 02:13:40 +0000

Make uscore_to_wincaps return NULL when passed NULL.

A null pointer dereference occurred when uscore_to_wincaps was passed

NULL, which could happen in some cases.

Fixes bug #8318.

dbus/dbus-gobject.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 678f8e5fdaf5c587547a96b173a5532f14337988
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-09 02:08:57 +0000

Only respond to NameOwnerChanged if its one of our names.

This fixes a crash due to code in

dbus_g_proxy_manager_replace_name_owner that was dereferencing a null pointer when the process received a nameownerchanged for an object not registered with dbus-glib.

Patch by Kimmo Hamalainen <kimmo.hamalainen@nokia.com>.

Fixes bug #8235.

dbus/dbus-gproxy.c | 29 ++++++++-----
1 file changed, 19 insertions(+), 10 deletions(-)

commit 1ce4d3e7324d92d797b7e41e869b1d0ff51f129a
Author: Rob Taylor <rob.taylor@codethink.co.uk>
Date: 2007-02-09 01:51:55 +0000

Fix dbus-binding-tool to generate headers usable from C++

Adds a cast that isn't needed in C, but is needed in C++

Thanks to Christian Persch <chpe@gnome.org>, though his patch had a typo :)

Fixes bug #6358.

```
dbus/dbus-binding-tool-glib.c |    2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 6d61bad839512dedb62318fd82e2c0b20906d62f  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-09 01:26:48 +0000
```

Only require --prefix for server side binding generation

In dbus-binding-tool, only require --prefix for server side binding generation, as client-side defaults to the useful org_foo_bar_baz. Also fixes up the help string.

Closes bug #4185

```
dbus/dbus-glib-tool.c |    5 +++--  
1 file changed, 3 insertions(+), 2 deletions(-)
```

```
commit cb866d68e733e48bd6a6cea21fa4b5ca648d861e  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-09 00:04:13 +0000
```

Clarify documentation for dbus_g_method_get_sender.

Adds a note that the caller is responsible for freeing the returned value.

```
dbus/dbus-gobject.c |    5 +++--  
1 file changed, 3 insertions(+), 2 deletions(-)
```

```
commit 22bd9971b4ac95f926b577e416f54ec57bb16d2d  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-08 23:54:33 +0000
```

Add new API for specifying the timeout in DbusGProxy calls.

This applies the patch from S. Nalliami <snallammai@novell.com> to provide new glib API's to allow the user to specify the timeout parameter for the method calls.

Currently, in the dbus-glib APIs, the timeout value for synchronous and asynchronous method calls is hard coded as -1 (ie. 25seconds) which inhibits the user from specifying shorter or longer timeout values. The new APIs take the timeout value as an argument and processes the method calls.

Fixes bug #9832.

```
dbus/dbus-glib.h | 16 +++++++  
dbus/dbus-gproxy.c | 121
```

```
+++++-----  
2 files changed, 128 insertions(+), 9 deletions(-)
```

```
commit b226873fdc2e7b98740afb93a7bae675952a29f0  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-08 23:17:40 +0000
```

Dont check for libxml2 when expat not found.

There isn't a version of dbus-gloader that works with libxml2, so this changes configure.ac to not check for libxml2, and error out if expat isn't found.

```
configure.ac | 47 +++++-----  
1 file changed, 6 insertions(+), 41 deletions(-)
```

```
commit c51ff16bc93b4741061c362de638ab5e8cace000  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-08 15:06:08 +0000
```

Add configure flags --with-introspect-xml

Adds the configure flags --with-introspect-xml, which allows dbus-glib to be built with pre-generated bus daemons introspection xml (such as can be gained with dbus-daemon --introspect). This allows dbus-glib to be built without a running dbus-daemon, useful for embedded systems.

```
configure.ac | 15 ++++++  
tools/Makefile.am | 5 +++++  
2 files changed, 20 insertions(+)
```

```
commit c5c0bed2166767162f599958270e92122da3ff2a  
Author: Rob Taylor <rob.taylor@codethink.co.uk>  
Date: 2007-02-08 15:00:17 +0000
```

update doc/reference/dbus-glib-undocumented.txt

dbus_g_thread_init is now documented.

```
doc/reference/dbus-glib-undocumented.txt | 7 +++----  
1 file changed, 3 insertions(+), 4 deletions(-)
```

```
commit c807d408924e9a9bcc7b2eb67b411c25888090ee  
Author: Rob Taylor <rob.taylor@collabora.co.uk>  
Date: 2007-01-08 08:41:11 +0000
```

Use dbus_threads_init_default() rather than using own threading primitives.

Modifies `dbus_g_threads_init()` to just call `dbus_threads_init_default()`, which with current `dbus` gives us full thread primitives with recursive locking.

Fixes #9259.

```
dbus/dbus-gthread.c |      143
+++-----
1 file changed, 7 insertions(+), 136 deletions(-)
```

commit 9bb928331e2ad602728320598698de90c9a246d6
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2007-01-08 08:39:22 +0000

Reduce dependancy to `dbus` version 0.93, error out if correct version not found.

In `configure.ac`, add code to error out if correct version of `dbus-1` not found. Also use correct dependancy of 0.93, as this is when the `--introspect` flag was introduced to `dbus-daemon`. Patch due to Luiz Augusto von Dentz <luiz.dentz@gmail.com>.

Fixes #8793.

```
configure.ac |      9 ++++++--
1 file changed, 7 insertions(+), 2 deletions(-)
```

commit 79d115030d45fef96d5ffc7853f80fc454c13b1
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2007-01-07 13:34:26 +0000

Allow `dbus` and `dbus-glib` to live in different prefixes

Adds `Cflags: -I${includedir}/dbus-1.0` to `dbus-glib-1.pc.in`.
Fixes #9384.

```
dbus-glib-1.pc.in | 1 +
1 file changed, 1 insertion(+)
```

commit 6de547af3bd7d2c8b77464435c45d22bba702cc7
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-26 10:22:26 +0100

Correct spellings in NEWS

```
NEWS | 37 ++++++-----
1 file changed, 23 insertions(+), 14 deletions(-)
```

commit 83f8ed0935c37b3b5668d1a512370a51f73ccc27
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 22:14:02 +0100

Update NEWS for release.

NEWS | 36 ++++++
1 file changed, 36 insertions(+)

commit c449668fd2b8b92fde03a83ba76a0eb8bb2a9326
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 21:12:03 +0100

only use -Wfloat-equal if compiler supports it
Closes #7658. Thanks to Jens Granseuer <jensgr@gmx.net> for the patch.

configure.ac | 31 ++++++-----
1 file changed, 27 insertions(+), 4 deletions(-)

commit 505e48a851e38664c45e482b8f75ebbfd27fe278
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 21:00:18 +0100

return NULL from g_return_val_if_fail in dbus_g_proxy_begin_call
Closes #4159.

dbus/dbus-gproxy.c | 4 +--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 62acdb49360dfbf994ef38c8e97d9190d30149f3
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 20:26:04 +0100

Add dbus-gidl.h to IGNORE_HFILES for doxygen docs

doc/reference/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit 0d9889a99241aebc863237faf8126663c97ae059
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 20:17:55 +0100

Update tools/Makefile.am for new dbus-binding-tool behaviour

tools/Makefile.am | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 0aa2db3f6f18bff7d5c62a00f052ca89e21b269f
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 20:03:31 +0100

Remove bashism in make-dbus-glib-error-enum.sh
Closes #6700.

dbus/make-dbus-glib-error-enum.sh | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 74bcfcfbc242325ca470d483817ae37fa8d41f25
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 19:37:12 +0100

Fix introspection when object has exported properties.
dbus-gobject:write_interface was completely broken
Fix thanks to mccann@jhu.edu.
Closes #8607.

dbus/dbus-gobject.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)

commit 3647d0df5780967dd505d3c046ab37ada70529bf
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 19:14:50 +0100

Require --prefix in dbus-binding-tool
Closes #4185.

dbus/dbus-glib-tool.c | 8 ++++++-
1 file changed, 7 insertions(+), 1 deletion(-)

commit 054c6476495a6f85ae708ed81b9053bf666897b7
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 18:34:49 +0100

Dont shadow index.
Rename useage of index to index_. Thanks stdlib...
Closes #8353.

dbus/dbus-gtype-specialized.c | 12 ++++++-----
dbus/dbus-gvalue.c | 14 ++++++-----
2 files changed, 13 insertions(+), 13 deletions(-)

commit d3a494dced8e3f2de111f23ae3fbab6e32f4051a
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 18:27:20 +0100

Fix small leak when marshal_table is destroyed
Closes #6870 with patch from Richard Hult <richard@imendio.com>

dbus/dbus-gobject.c | 9 ++++++-
1 file changed, 8 insertions(+), 1 deletion(-)

commit 9bfec032a72e7af5945336fecbb9b6e0b6f2de9e
Author: Rob Taylor <rob.taylor@collabora.co.uk>

Date: 2006-10-25 18:24:53 +0100

Fixes crash if disposing one DbusGProxy causes another for the same service to be unrefed in a destroyed callback.

```
dbus/dbus-gproxy.c      | 14 ++++++--
test/core/test-dbus-glib.c |    67
+++++-----
2 files changed, 78 insertions(+), 3 deletions(-)
```

commit 53bf71bd852810a271d13a73ba445622616c711c
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 16:41:38 +0100

Bump version to 0.72
and use modern AC_INIT, AM_INIT_AUTOMAKE

```
configure.ac |    4 +-
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 28a54df202f089a3e00930e36a787c175501be41
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 14:36:25 +0100

Clean generated run-with-tmp-session-bus.conf on make clean

```
test/core/Makefile.am |    4 +-
1 file changed, 3 insertions(+), 1 deletion(-)
```

commit bcf15d72892582c28755563ca63182c198ef8f64
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 14:35:25 +0100

Actually run unit tests and checks when doing make distcheck

```
Makefile.am |    2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 7c21166ef7ce9b370448a39c6c6744813933abee
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 14:33:53 +0100

Use TEST_CORE_SERVICE_BINARY path for core test service file

```
test/data/valid-service-files/debug-glib.service.in |    2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit d6980d9d59edcf14cc1ab5af3945bd740a3f2495
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 14:32:14 +0100

Use dbus-daemon --introspect to generate DBus service introspect xml

```
tools/Makefile.am | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 505f5486f055558934718672f1328b52fab6cd6e
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-25 14:30:23 +0100

Actually make interfaces tests work
Working interfaces/Makefile.am
Make service file for interfaces test service
Fix up configure.ac appropriately.
Sanitise service/object namespace for interfaces tests
Remove accidentally added test/interfaces/.Makefile.am.sw

```
configure.ac | 6 ++-  
test/interfaces/Makefile.am | 81  
+++++-----  
test/interfaces/run-test.sh | 23 +-----  
test/interfaces/test-client.c | 22 +++++-----  
test/interfaces/test-goodbye.xml | 4 +-  
test/interfaces/test-hello.xml | 4 +-  
test/interfaces/test-interfaces.c | 4 +-  
test/interfaces/test-objects.c | 8 ++--  
test/interfaces/test-server.c | 8 ++--  
test/interfaces/test-song.xml | 4 +-  
10 files changed, 69 insertions(+), 95 deletions(-)
```

commit 4a479fe9516b141d94e57489f7e624ea6b94cb94
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-24 19:51:12 +0100

Make interfaces tests work
Working interfaces/Makefile.am
Make service file for interfaces test service
Fix up configure.ac appropriately.
Sanitise service/object namespace for interfaces tests
Remove accidentally added test/interfaces/.Makefile.am.swp

```
test/data/valid-service-files/interfaces-test.service.in | 3 +++  
test/interfaces/.Makefile.am.swp | Bin 12288 ->  
0 bytes  
2 files changed, 3 insertions(+)
```

commit 4db00a26f0c67db96ff6185286658278678e23bd
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-24 18:07:26 +0100

Changes for test/interfaces
Changed configure.ac and test/Makefile.am

```
configure.ac      |      1 +
test/Makefile.am |      4 ++--
2 files changed, 3 insertions(+), 2 deletions(-)
```

commit 4f6a75760f55eeb79113d09baa46ecdd09e100e6
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-24 16:36:08 +0100

Add tests for new interfaces functionaility
From bug #5173

```
test/interfaces/.Makefile.am.swp | Bin 0 -> 12288 bytes
test/interfaces/.gitignore      |  4 ++
test/interfaces/Makefile.am     |  86 ++++++
test/interfaces/run-test.sh     |  38 ++++++
test/interfaces/test-client.c   |  85 ++++++
test/interfaces/test-goodbye.xml |   9 +++
test/interfaces/test-hello.xml  |  10 +++
test/interfaces/test-interfaces.c | 126
+++++
test/interfaces/test-interfaces.h |  48 ++++++
test/interfaces/test-objects.c  |  73 ++++++
test/interfaces/test-objects.h  |  31 ++++++
test/interfaces/test-server.c   |  56 ++++++
test/interfaces/test-song.xml   |   9 +++
13 files changed, 575 insertions(+)
```

commit 4467ffca15e4783b7e90f1349be18fca3807b6cb
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-24 15:42:42 +0100

Bump GLib dependancy to 2.6
Closes #4390.

```
configure.ac |      4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 6ac144ff50d6c87795b470fb74cc1c0dd3e549a7
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-17 17:06:48 +0100

Add gobject-2.0 to dbus-glib-1.pc.in
glib bindings depend on gobject. Its amazing we've got this far
with
noone noticiing this bug...

```
dbus-glib-1.pc.in |  2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 03e629cabe1100527c467fc5bf2a8aa4f7d3a6f5
Author: Rob Taylor <rob.taylor@collabora.co.uk>

Date: 2006-10-04 19:30:46 -0400

Remove accidentally added test/interfaces stuff in the autotools

```
configure.ac      |    1 -
test/Makefile.am |    4 +++-
2 files changed, 2 insertions(+), 3 deletions(-)
```

commit 8e2c2ba5f8f0a25708bac86cafa46be02a1bdb78
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-04 19:16:35 -0400

Fix up tests

Puts all exposed servies in the org.freedesktop.DBus.GLib
namespace

Make tests run again!

```
test/core/run-test.sh          |    6 +-
test/core/test-dbus-glib.c     |   142
+++++++-----
test/core/test-profile.c       |    6 +-
test/core/test-service-glib.c  |   16 +-
test/core/test-service-glib.xml |    6 +-
test/core/test-thread-client.c |    4 +-
test/core/test-thread-server.c |    2 +-
.../data/valid-service-files/debug-echo.service.in |    2 +-
.../data/valid-service-files/debug-glib.service.in |    2 +-
test/test-service.c            |   28 +-
10 files changed, 107 insertions(+), 107 deletions(-)
```

commit c0bb50ca11675274464d91ed5651676449ce360d
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-04 19:13:34 -0400

Update autofoo for new tests layout

```
configure.ac      |    1 +
test/Makefile.am |    4 +++-
2 files changed, 3 insertions(+), 2 deletions(-)
```

commit 33412cde926a9fc6ce86618835661bbc48475177
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-10-04 23:43:29 +0100

Make autogen a bit more useful for maintainers

Adds default configure parameters - the ones you usually want ;)

```
autogen.sh |   12 ++++++----
1 file changed, 8 insertions(+), 4 deletions(-)
```

commit b47e9bca585b1104f6d7309b02da27b123015611
Author: Rob Taylor <rob.taylor@collabora.co.uk>

Date: 2006-09-27 14:12:26 +0100

Update COPYING and HACKING to be correct for dbus-glib

```
COPYING |    9 ++++-----
HACKING |   11 ++++++++
2 files changed, 15 insertions(+), 5 deletions(-)
```

commit 50541a23757d59d93918db52c9e9c65596d943c5
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-27 14:04:19 +0100

Autoconf changes from test/glib -> test/core move

```
configure.ac      |    4 ++--
test/Makefile.am |    4 ++--
2 files changed, 4 insertions(+), 4 deletions(-)
```

commit 01024785e818e72c06ba44e8f6ea926e8aa8f7a8
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-27 14:02:42 +0100

add an .gitignore

```
.gitignore | 205
+++++
1 file changed, 205 insertions(+)
```

commit 003840bd5f08802fb2a63b3c49f9d0685f6f292b
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-27 13:56:39 +0100

Move tests/glib to test/core
Move tests/glib to test/core in preparation for separate
functionality
tests

```
test/core/Makefile.am      |    86 ++
test/core/my-object-marshal.list |    2 +
test/core/run-test.sh      |    38 +
test/core/test-dbus-glib.c | 1614
+++++
test/core/test-profile.c   | 1150 ++++++
test/core/test-service-glib.c |   928 ++++++
test/core/test-service-glib.xml |   179 +++++
test/core/test-thread-client.c |    98 +++
test/core/test-thread-server.c |   209 +++++
test/core/test-thread.h    |     1 +
test/glib/Makefile.am      |    86 --
test/glib/my-object-marshal.list |    2 -
test/glib/run-test.sh      |    38 -
test/glib/test-dbus-glib.c | 1614
```

```
-----
test/glib/test-profile.c      | 1150 -----
test/glib/test-service-glib.c |  928 -----
test/glib/test-service-glib.xml | 179 -----
test/glib/test-thread-client.c |  98 ---
test/glib/test-thread-server.c | 209 -----
test/glib/test-thread.h      |  1 -
20 files changed, 4305 insertions(+), 4305 deletions(-)
```

commit de74dd48194f9aa3629d7b65dd89e0b1339f8654
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-27 13:39:24 +0100

Rename configure.in to configure.ac for modernity

```
configure.ac | 902
+++++
configure.in | 902
-----
2 files changed, 902 insertions(+), 902 deletions(-)
```

commit b17fc759d2d5422ce287706d276443fa338cbf5a
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-24 14:34:19 +0100

Fix memleak in lookup_or_register_specialized.
Applies fix from Daniel d'Andrada TenÃ³rio de Carvalho, closing
bug #7352.

```
dbus/dbus-gtype-specialized.c | 3 +--
1 file changed, 1 insertion(+), 2 deletions(-)
```

commit 4d6aa726814295cfabab95a51d2e320d02b72dde
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-24 14:32:52 +0100

Add m4 directory
Add missing changes from last commit.

```
Makefile.am | 9 +++++----
autogen.sh  | 6 +++++-
configure.in | 1 +
3 files changed, 11 insertions(+), 5 deletions(-)
```

commit 1571d556aa5f087e52074e5fd7c5e5c29667c551
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-24 14:07:50 +0100

Add m4 directory
Add an m4 directory and add gtk-doc.m4, which is installed in the
tree by gtkdocize.

m4/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit fcc2ccdb19bfa25db005494271dbd91a3205c3a6
Merge: 99e1085 3f978cf
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-09-21 03:44:22 +0100

Merge branch 'master' of
git+ssh://git.freedesktop.org/git/dbus/dbus-glib

commit 3f978cf34b91e5404dc3ea559995a9648734a2da
Author: John (J5) Palmieri <johnp@remedyz.boston.redhat.com>
Date: 2006-09-08 10:56:06 -0400

* tools/Makefile.am: Add tools/session.conf to EXTRA_DIST so make
check works
from tarballs

tools/Makefile.am | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 99e10857c50a025fb37613c4dae720b5afc890f5
Merge: b193f2b 22ea91b
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-24 23:27:58 +0100

Merge <http://ms800.montefiore.ulg.ac.be/~frecinau/dbus-glib>

commit 22ea91b33bff95cf480a4c992e4774b671f1534b
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-22 22:57:44 +0200

Fix compilation with -Werror

dbus/dbus-gmain.c | 2 +-
test/glib/test-dbus-glib.c | 9 +-----
2 files changed, 2 insertions(+), 9 deletions(-)

commit 368aaee38049dc599fb8d06d7096bcd04befab
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-22 22:26:24 +0200

Make test scripts run during out-of-tree compilation

test/glib/run-test.sh | 4 +++-
tools/run-with-tmp-session-bus.sh | 5 ++++-
2 files changed, 7 insertions(+), 2 deletions(-)

commit b193f2b28472790ff66fc575d55d4e6d5a9d3e78
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-22 19:36:44 +0200

Fix out-of-tree compilation
(cherry picked from 4b85a9e92bcf21356f25c535b3550a330cb018a4
commit)

```
dbus/Makefile.am          | 13 ++++++++--  
dbus/examples/Makefile.am | 8 ++++++-  
dbus/examples/statemachine/Makefile.am | 9 +++++++-  
test/Makefile.am         | 7 ++++++-  
test/glib/Makefile.am    | 8 ++++++-  
5 files changed, 39 insertions(+), 6 deletions(-)
```

commit 4b85a9e92bcf21356f25c535b3550a330cb018a4
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-22 19:36:44 +0200

Fix out-of-tree compilation

```
dbus/Makefile.am          | 13 ++++++++--  
dbus/examples/Makefile.am | 8 ++++++-  
dbus/examples/statemachine/Makefile.am | 9 +++++++-  
test/Makefile.am         | 7 ++++++-  
test/glib/Makefile.am    | 8 ++++++-  
5 files changed, 39 insertions(+), 6 deletions(-)
```

commit 46e7e573203e354b676e031067b8607b4f2640df
Merge: 355ef78 3be8161
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-22 19:07:59 +0200

Merge branch 'master' of
git://anongit.freedesktop.org/git/dbus/dbus-glib

commit 3be816118dfc560955e5cd8e85203129087959ce
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-22 12:25:39 +0100

replace doxygen with gtk-doc in INSTALL

```
INSTALL | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit 27e76bb78db2b467ce93fa0088c6c2407836e4ff
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-22 12:25:13 +0100

Combine .PHONY rules in toplevel Makefile.am
A .PHONY rule was added by the gtk-doc patch, this combines it
with
the existings .PHONY rule

```
Makefile.am | 5 +----
```

1 file changed, 2 insertions(+), 3 deletions(-)

commit d49a2a2486cd60e22b360dda8ae5a50d25486667
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-22 12:24:00 +0100

Modified autogen.sh to check for autotools and gtkdocize versions
autogen.sh now does a full check for all the correct versions of
autotools. It uses autoreconf to do the various steps. It also
checks for correct gtkdocize version and pretty prints a bit,
a-la gnome-autogen.sh.

autogen.sh | 143
+++++-----
1 file changed, 101 insertions(+), 42 deletions(-)

commit 62419419257d6353c5f35001e1f538fca300f3c4
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-22 11:44:42 +0100

add missing doc/Makefile.am from gtk-doc update.

doc/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit f812d9044aa73b0c86f92dc0dcae03d26e23193a
Author: Rob Taylor <rob.taylor@collabora.co.uk>
Date: 2006-08-22 11:43:43 +0100

Add missing m4/gtk-doc.m4 from gtk-doc patch.

m4/gtk-doc.m4 | 1 +
1 file changed, 1 insertion(+)

commit 2effafca679d6331201e7637b18888a70b958eb8
Author: Rob Taylor <robtaylor@lancasteria.(none)>
Date: 2006-08-18 20:55:17 +0100

Commit patch to switch to gtk-doc with gtype-specialized doc and
updates from
Marc-Andre Lureau <marcandre.lureau@gmail.com>, with minor
cleanup.

Closes #7726.

Makefile.am | 5 +-
autogen.sh | 4 +
configure.in | 8 ++
dbus/dbus-gidl.c | 2 +-
dbus/dbus-glib-tool.c | 2 +-
dbus/dbus-glib.c | 134 ++++++-----
dbus/dbus-glib.h | 41 ++++---


```

dbus/dbus-gmain.c          |    46 ++++---
dbus/dbus-gobject.c       |   140 ++++++-----
dbus/dbus-gproxy.c        |   193
+++++-----
dbus/dbus-gtype-specialized.c |    20 +++-
dbus/dbus-gutils.c        |     2 +-
dbus/dbus-gvalue.c        |     6 +-
doc/reference/Makefile.am |    94 ++++++
doc/reference/dbus-binding-tool.xml |    62 ++++++
doc/reference/dbus-glib-docs.sgml |    83 ++++++
doc/reference/dbus-glib-sections.txt |   166
+++++
doc/reference/dbus-glib-undocumented.txt |    78 ++++++
doc/reference/version.xml.in |     1 +
19 files changed, 858 insertions(+), 229 deletions(-)

```

```

commit 355ef78d98d6fc65715845d56232199162cab12a
Author: Steve FrÃ©cinaux <steve@istique.net>
Date: 2006-08-17 11:48:20 +0200

```

Interface support for bindings.

```

dbus/dbus-gobject.c |    286
+++++-----
1 file changed, 161 insertions(+), 125 deletions(-)

```

```

commit e6c7815f9457757d7364463759766c85981ce207
Author: John (J5) Palmieri <johnp@remedyz.boston.redhat.com>
Date: 2006-07-24 14:44:36 -0400

```

* Released 0.71

```

NEWS          |     7 +++++-
configure.in  |     2 +-
2 files changed, 7 insertions(+), 2 deletions(-)

```

```

commit 3f720d5a5028d2b4075b474ba17fb9b7508093ac
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-07-21 16:25:35 -0400

```

* Various Makefile cleanups

```

Makefile.am      |    26 ++++++-----
dbus/Makefile.am |     8 ++++---
test/Makefile.am |     2 --
3 files changed, 28 insertions(+), 8 deletions(-)

```

```

commit 05cb3fd6e2c139e61d1461443660badfbdf94a83
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-07-17 16:36:02 -0400

```

* Released 0.70

* Add NEWS to the dist

Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit d59c4a541397033b69d575c8cc495d0f3ba13ee3
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-07-17 16:28:45 -0400

* Added a NEWS entry for this release

NEWS | 18 ++++++
1 file changed, 18 insertions(+)

commit 990802d9f01a7c5a1ae582aad21963fe20b601e
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-07-17 16:06:52 -0400

* remove ChangeLog because this will now be generated during dist

0 files changed

commit 6a68e96e69cb17a981c801d8e7480c5bb4d51a6c
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-17 15:52:26 -0400

* add the Doxyfile.in file

Doxyfile.in | 180
+++++
1 file changed, 180 insertions(+)

commit 89a2a3b5b5795434900f1ca41fc73162b7bb09f2
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 19:16:07 -0400

- distcheck cleanups

dbus/Makefile.am | 2 +-
tools/Makefile.am | 6 +++++
2 files changed, 7 insertions(+), 1 deletion(-)

commit 46ea5dc1aa64ca906d28119e2d9eed3a48e6adfc
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 19:05:44 -0400

- Renabled another test

dbus/Makefile.am | 20 ++++++-----
1 file changed, 10 insertions(+), 10 deletions(-)

commit ae743d6feca5e24541124148a53a475f6f27e2a9

Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 18:52:50 -0400

- add gcov source

```
test/decode-gcov.c | 2652
+++++
1 file changed, 2652 insertions(+)
```

commit ae90f4bab7a98d89a85e8e6c11b4af72bb9a2525
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 18:52:16 -0400

- Adding placeholder HACKING file

0 files changed

commit 5f47ecc191cael2930652b708d3c52d293eba4da
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 18:05:37 -0400

- Add back the Doxygen doc generation

```
Makefile.am      | 2 +-
configure.in     | 1 +
doc/api/README   | 3 +++
3 files changed, 5 insertions(+), 1 deletion(-)
```

commit fdb8aaaab032ab4eb63aa80dbef7de91a3de346a
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 17:40:30 -0400

- Add back tests to the glib bindings

```
Makefile.am      | 5 +-
configure.in     | 17 +-
test/Makefile.am | 29 +
.../data/valid-service-files/debug-echo.service.in | 3 +
.../data/valid-service-files/debug-glib.service.in | 3 +
test/glib/Makefile.am | 80 +
test/glib/my-object-marshal.list | 2 +
test/glib/run-test.sh | 36 +
test/glib/test-dbus-glib.c | 1621
+++++
test/glib/test-profile.c | 1150 ++++++
test/glib/test-service-glib.c | 928 ++++++
test/glib/test-service-glib.xml | 179 +++
test/glib/test-thread-client.c | 98 ++
test/glib/test-thread-server.c | 209 +++
test/glib/test-thread.h | 1 +
test/test-service.c | 374 +++++
tools/Makefile.am | 13 +
```

```
tools/run-with-tmp-session-bus.sh          | 65 +
tools/session.conf                          | 24 +
19 files changed, 4833 insertions(+), 4 deletions(-)
```

commit 24e982d596036202850c679aa0460b729840e45d
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-07-06 14:40:14 -0400

- Fix up docs

```
INSTALL |    78
++-----
README  |    71 -----
-----
2 files changed, 2 insertions(+), 147 deletions(-)
```

commit 6aa2aba0a64ea19e504118dc56a1f9e559568f90
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-29 12:05:05 -0400

- Really commit the glib to dbus move and build fixes

```
Makefile.am          |    10 +++++-----
configure.in         |    18 ++++++++-----
dbus/dbus-glib.h     |     2 +-
dbus/examples/Makefile.am |    14 +++++-----
dbus/examples/statemachine/Makefile.am |    10 +++++-----
5 files changed, 31 insertions(+), 23 deletions(-)
```

commit 6a9ac47cc9143ee899455c576f557818dab1f7d2
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-29 11:26:07 -0400

- Created empty NEWS file
- renamed glib dir to dbus so include paths stay the same in the examples
- build now works

```
dbus/.cvsignore      |  13 +
dbus/Makefile.am     |  111 +
dbus/dbus-binding-tool-glib.c | 1625 ++++++++
dbus/dbus-binding-tool-glib.h |   40 +
dbus/dbus-gidl.c     |   788 +++++
dbus/dbus-gidl.h     |   158 ++
dbus/dbus-glib-lowlevel.h |   72 +
dbus/dbus-glib-tool.c |   489 ++++
dbus/dbus-glib-tool.h |   38 +
dbus/dbus-glib.c     |   304 +++
dbus/dbus-glib.h     |   256 ++
dbus/dbus-gloader-expat.c |   266 ++
dbus/dbus-gmain.c    |   814 +++++
dbus/dbus-gmarshal.c |   89 +
```

```

dbus/dbus-gmarshal.h | 21 +
dbus/dbus-gmarshal.list | 1 +
dbus/dbus-gobject.c | 2217
+++++
dbus/dbus-gobject.h | 43 +
dbus/dbus-gparser.c | 881 ++++++
dbus/dbus-gparser.h | 65 +
dbus/dbus-gproxy.c | 2748
+++++
dbus/dbus-gsignature.c | 210 ++
dbus/dbus-gsignature.h | 19 +
dbus/dbus-gtest-main.c | 51 +
dbus/dbus-gtest.c | 92 +
dbus/dbus-gtest.h | 38 +
dbus/dbus-gthread.c | 179 ++
dbus/dbus-gtool-test.h | 31 +
dbus/dbus-gtype-specialized.c | 778 ++++++
dbus/dbus-gtype-specialized.h | 176 ++
dbus/dbus-gutils.c | 130 +
dbus/dbus-gutils.h | 57 +
dbus/dbus-gvalue-utils.c | 1439 ++++++++
dbus/dbus-gvalue-utils.h | 73 +
dbus/dbus-gvalue.c | 1854 ++++++++
dbus/dbus-gvalue.h | 43 +
dbus/examples/.cvsignore | 17 +
dbus/examples/Makefile.am | 38 +
dbus/examples/example-client.c | 121 +
dbus/examples/example-service.c | 153 ++
dbus/examples/example-service.xml | 19 +
dbus/examples/example-signal-emitter.c | 132 +
dbus/examples/example-signal-emitter.xml | 13 +
dbus/examples/example-signal-recipient.c | 102 +
dbus/examples/statemachine/.cvsignore | 16 +
dbus/examples/statemachine/Makefile.am | 35 +
dbus/examples/statemachine/sm-marshal.list | 1 +
dbus/examples/statemachine/statemachine-client.c | 662 ++++++
dbus/examples/statemachine/statemachine-server.c | 229 ++
dbus/examples/statemachine/statemachine-server.h | 37 +
dbus/examples/statemachine/statemachine-server.xml | 14 +
dbus/examples/statemachine/statemachine.c | 353 +++
dbus/examples/statemachine/statemachine.h | 77 +
dbus/examples/statemachine/statemachine.xml | 33 +
dbus/make-dbus-glib-error-enum.sh | 25 +
dbus/make-dbus-glib-error-switch.sh | 29 +
glib/.cvsignore | 13 -
glib/Makefile.am | 105 -
glib/dbus-binding-tool-glib.c | 1625 -----
glib/dbus-binding-tool-glib.h | 40 -
glib/dbus-gidl.c | 788 -----
glib/dbus-gidl.h | 158 --
glib/dbus-glib-lowlevel.h | 72 -
glib/dbus-glib-tool.c | 489 ----

```

```

glib/dbus-glib-tool.h          | 38 -
glib/dbus-glib.c              | 304 ---
glib/dbus-glib.h              | 256 --
glib/dbus-gloader-expat.c     | 266 --
glib/dbus-gmain.c             | 814 -----
glib/dbus-gmarshal.c         | 89 -
glib/dbus-gmarshal.h         | 21 -
glib/dbus-gmarshal.list      | 1 -
glib/dbus-gobject.c          | 2217
-----
glib/dbus-gobject.h          | 43 -
glib/dbus-gparser.c          | 881 -----
glib/dbus-gparser.h          | 65 -
glib/dbus-gproxy.c           | 2748
-----
glib/dbus-gsignature.c        | 210 --
glib/dbus-gsignature.h        | 19 -
glib/dbus-gtest-main.c       | 51 -
glib/dbus-gtest.c            | 92 -
glib/dbus-gtest.h            | 38 -
glib/dbus-gthread.c          | 179 --
glib/dbus-gtool-test.h        | 31 -
glib/dbus-gtype-specialized.c | 778 -----
glib/dbus-gtype-specialized.h | 176 --
glib/dbus-gutils.c           | 130 -
glib/dbus-gutils.h           | 57 -
glib/dbus-gvalue-utils.c     | 1439 -----
glib/dbus-gvalue-utils.h     | 73 -
glib/dbus-gvalue.c           | 1854 -----
glib/dbus-gvalue.h           | 43 -
glib/examples/.cvsignore     | 17 -
glib/examples/Makefile.am     | 38 -
glib/examples/example-client.c | 121 -
glib/examples/example-service.c | 153 --
glib/examples/example-service.xml | 19 -
glib/examples/example-signal-emitter.c | 132 -
glib/examples/example-signal-emitter.xml | 13 -
glib/examples/example-signal-recipient.c | 102 -
glib/examples/statemachine/.cvsignore | 16 -
glib/examples/statemachine/Makefile.am | 35 -
glib/examples/statemachine/sm-marshal.list | 1 -
glib/examples/statemachine/statemachine-client.c | 662 -----
glib/examples/statemachine/statemachine-server.c | 229 --
glib/examples/statemachine/statemachine-server.h | 37 -
glib/examples/statemachine/statemachine-server.xml | 14 -
glib/examples/statemachine/statemachine.c | 353 ---
glib/examples/statemachine/statemachine.h | 77 -
glib/examples/statemachine/statemachine.xml | 33 -
glib/make-dbus-glib-error-enum.sh | 25 -
glib/make-dbus-glib-error-switch.sh | 29 -
112 files changed, 18315 insertions(+), 18309 deletions(-)

```

commit 7fa1f1471e5b878253695da46a89a3186fb2a0d1
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-28 14:39:18 -0400

- Take out comment in a literal

configure.in | 1 -
1 file changed, 1 deletion(-)

commit 1a7d56d33f83dde917871f8b2d5af19f25786449
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-28 14:24:40 -0400

- Have autogen.sh point to dbus-glib

autogen.sh | 4 +--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 24ec6aba3281c460f64bbe9d5bb2dd265d453cce
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-28 14:22:25 -0400

- Add files missed in the split

glib/dbus-glib-lowlevel.h | 72 ++++++++
glib/dbus-glib.h | 256
+++++
glib/make-dbus-glib-error-enum.sh | 25 +++
3 files changed, 353 insertions(+)

commit b8b86555280694d85d64b68d0f9131868598161a
Author: John (J5) Palmieri <quinticent@phuket.(none)>
Date: 2006-06-28 14:21:26 -0400

- Add boilerplate build files

AUTHORS | 16 +
COPYING | 551 ++++++
INSTALL | 294 ++++++
Makefile.am | 57 +++
README | 101 +++++
autogen.sh | 93 +++++
configure.in | 870
+++++
dbus-glib-1.pc.in | 12 +
8 files changed, 1994 insertions(+)

commit 698177269a46c2cb80fbb2c471f5ad6314bd106b
Author: Ross Burton <ross@openedhand.com>
Date: 2006-06-14 16:49:37 +0000

2006-06-14 Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c:
Free a leaking GArray (surely not!) in dbus_g_method_return.

glib/dbus-gobject.c | 1 +
1 file changed, 1 insertion(+)

commit 0b305ffe12e7273640226e323074f2cb7d7387cc
Author: Ross Burton <ross@openedhand.com>
Date: 2006-06-12 14:22:48 +0000

2006-06-12 Ross Burton <ross@openedhand.com>

* glib/dbus-gproxy.c:
Don't leak a GArray when firing signals (thank Rob Taylor for review).

glib/dbus-gproxy.c | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)

commit 66e5632e14760bbd396afbc4d474cbbed22f37f7
Author: Robert McQueen <robot101@debian.org>
Date: 2006-06-07 00:31:01 +0000

2005-05-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gtype-specialized.c: Fix obvious leak of GArray
in every
call to dbus_g_type_get_struct.

glib/dbus-gtype-specialized.c | 9 ++++++--
1 file changed, 7 insertions(+), 2 deletions(-)

commit 9b151fd7a396463aa3f7c6ac0a14df444c78a6b1
Author: Robert McQueen <robot101@debian.org>
Date: 2006-06-07 00:03:57 +0000

2005-05-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Fix the failing test where static
string
pointers were put into a GPtrArray-based specialised
collection,
and
then freed along with the array. GValues which you add into
collections or maps which have the NOCOPY flag set are assumed
to not
belong to the caller, so rather than the existing
pointer-stealing
semantics, they are copied instead. Given that the main
consumers of
this abstraction are the bindings themselves, I don't think

this is
too bad, but others should watch their choice of take vs
set_static.

glib/dbus-gvalue-utils.c | 24 ++++++-----
1 file changed, 14 insertions(+), 10 deletions(-)

commit 5559e67b90b3c186bf0002253587013264ce9dc0
Author: Robert McQueen <robot101@debian.org>
Date: 2006-06-06 23:07:04 +0000

2005-05-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Spotted a warning about the return
value
of `g_slist_prepend` not being used. Fixed copying of `slist`-based
specialised collections, then wrote a test case and found that
it was
all broken. Went on to fix iterating and appending too. Good
thing
nobody uses this code yet.

glib/dbus-gvalue-utils.c | 101
+++++-----
1 file changed, 89 insertions(+), 12 deletions(-)

commit 7ca236d43bc91778e90ac59500aeb1d68c8acd6f
Author: Robert McQueen <robot101@debian.org>
Date: 2006-06-06 19:45:39 +0000

2005-05-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Remove duplicated code by having
all of
the iterators use `gvalue_take_ptrarray_value` (the `GValues`
themselves
are discarded without unsetting, so it makes no difference
whether
we take or `set_static`). Remove cases for `G_TYPE_POINTER` because
there really is nothing useful we can do with them in our
specialised types - we **need** boxed copy/free functions at
the very
least.

glib/dbus-gvalue-utils.c | 91
+++++-----
1 file changed, 16 insertions(+), 75 deletions(-)

commit 8896d43457634c4c10d9909907d0d9082b520a07
Author: Havoc Pennington <hp@redhat.com>
Date: 2006-05-21 05:33:52 +0000

2006-05-21 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c: Put in a pile of assertions that the proxy name is not NULL when it shouldn't be. Also a couple of possible fixes for #4637 though I don't understand why the bug happens, to be honest... also the object constructor has an assert name != NULL and the name is only currently NULL for peer-to-peer proxies that I don't think anyone uses? So it should be asserting. Anyway, for now at least see if we get an earlier assertion failure.

* glib/dbus-gvalue-utils.c: Put in a couple of assertions for apparently broken code to be sure the tests fail and someone will fix them...

```
glib/dbus-gvalue-utils.c | 15 ++++++
1 file changed, 15 insertions(+)
```

commit 4da76dd7d5845f05efd553037750ddef7d1962
Author: Robert McQueen <robot101@debian.org>
Date: 2006-04-28 21:11:28 +0000

2005-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gproxy.c: Fix properties so that they can be given in any order, making it easier for people who inherit from this object.

```
glib/dbus-gproxy.c | 29 ++++++-----
1 file changed, 22 insertions(+), 7 deletions(-)
```

commit 08cd19641ff8c6fcb1f8e5821e6c9dc81f11cd6f
Author: Robert McQueen <robot101@debian.org>
Date: 2006-04-28 21:02:16 +0000

2005-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Patch from Jakub Stachowski to fix leaking of memory from within pointer arrays and lists. Fixes bug #6300.

```
glib/dbus-gvalue-utils.c | 62
+++++
1 file changed, 56 insertions(+), 6 deletions(-)
```

commit 96fb792c6a5661afeef76ece3cd7defc9ffc8db8
Author: Robert McQueen <robot101@debian.org>
Date: 2006-04-28 20:36:56 +0000

2005-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue.c: Patch from Jakub Stachowski to fix a leak
in
generating struct signatures. Fixes bug #6083.

glib/dbus-gvalue.c | 9 +++++---
1 file changed, 6 insertions(+), 3 deletions(-)

commit 636fe6ed1b321cd05b72ca39d414ecd1c2cb4b4f
Author: Havoc Pennington <hp@redhat.com>
Date: 2006-02-26 01:06:33 +0000

2006-02-25 Havoc Pennington <hp@redhat.com>

* glib/dbus-glib-tool.c (usage): fix up the usage message,
someone
should make this thing use the new glib options parser

glib/dbus-glib-tool.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 844b4184483ac979eff27319f144e8f330858062
Author: Robert McQueen <robot101@debian.org>
Date: 2006-02-16 00:43:41 +0000

2006-02-16 Robert McQueen <robot101@debian.org>

* dbus/dbus-message.c (dbus_message_iter_get_fixed_array):
Patch from Rob Taylor <rob.taylor@collabora.co.uk> to correct
a bogus
assertion that the next element to read from the iter is fixed in
size. This is not the case when you are at the end of the iter,
because the next element type is INVALID.

* dbus/dbus-string.c (_dbus_string_init_const_len): Correct a
a bogus assert which means that you may not initialise a 0-length
string unless you provide a non-NULL pointer. This prevented
you from marshalling messages containing zero-length arrays in
some cases.

* glib/dbus-gvalue.c (demarshal_collection_array): Another patch
from Rob to correct bogus asserts when trying to demarshal an
array and get_fixed_array got you 0 elements. Append nothing to
the GArray in this case.

* test/glib/test-dbus-glib.c: Add a test case for round-tripping
an empty array via the glib bindings. Without all of the above
patches, this new test fails.

glib/dbus-gvalue.c | 7 +++++---
1 file changed, 4 insertions(+), 3 deletions(-)

commit 7b33862bac766c00c12f5cb400af06200ae004c4

Author: Robert McQueen <robot101@debian.org>
Date: 2006-02-15 23:45:48 +0000

2006-02-16 Robert McQueen <robot101@debian.org>

- * glib/dbus-gmain.c: Make the previous commit compile.
- * python/_dbus.py, python/matchrules.py: Patch from Ole Andre Ravnaas <ole.andre.ravnaas@collabora.co.uk> to allow you to specify sender_keyword="foo", path_keyword="bar" when adding a signal listener, so that you can bind to signals generically but still do something useful in your callback.
- * python/dbus_bindings.pyx: Demarshal the byte type as unsigned chars so that they're not cast to chars and made negative. Thanks to Jakub Stachowski for reporting this and testing the fix.

glib/dbus-gmain.c | 8 ++++----
1 file changed, 4 insertions(+), 4 deletions(-)

commit 8d98ce884210745d7070e4438167ece2be0e357b
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-02-15 21:42:54 +0000

2006-02-15 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-glib.h:
- * glib/dbus-gmain.h:
(dbus_g_connection_open): new method for opening
a connection to an arbitrary address in the glib bindings
- * ChangeLog: checkin last entry which doesn't seem to be committed

glib/dbus-gmain.c | 37 +++++++++++++++++++++++++++++++++++++++
1 file changed, 37 insertions(+)

commit 6fd3dffe4561b37bc75c13f6296e91b65224ce4d
Author: Robert McQueen <robot101@debian.org>
Date: 2006-02-13 22:30:11 +0000

2006-02-13 Robert McQueen <robot101@debian.org>

- * glib/dbus-binding-tool-glib.c, glib/dbus-gmain.c,
glib/dbus-gsignature.c, glib/dbus-gtype-specialized.c,
glib/dbus-gtype-specialized.h, glib/dbus-gvalue-utils.c,
glib/dbus-gvalue-utils.h, glib/dbus-gvalue.c:
Patch from Rob Taylor <rob.taylor@collabora.co.uk> to add a big
missing piece of the glib bindings jigsaw puzzle. This modifies
the existing specialised types to have N type parameters (rather
than the current 1 or 2 for arrays and dictionaries
respectively).
You can then use this to get a glib type to represent any

arbitrary
D-Bus struct type using `dbus_g_type_get_struct`. The only implementation of these types is with `GValueArrays` as before, but it's now possible to store these in arrays, emit them in signals, etc.

```
glib/dbus-binding-tool-glib.c | 26 ++++
glib/dbus-gmain.c             | 4 +-
glib/dbus-gsignature.c        | 25 +++-
glib/dbus-gtype-specialized.c | 296
+++++-----
glib/dbus-gtype-specialized.h | 43 ++++++
glib/dbus-gvalue-utils.c      | 184 ++++++
glib/dbus-gvalue-utils.h      | 2 +-
glib/dbus-gvalue.c            | 148 ++++++
8 files changed, 695 insertions(+), 33 deletions(-)
```

commit eef27dfbd6dc73baa13c78220f98297881071368
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-27 16:40:53 +0000

2006-01-27 Robert McQueen <robot101@debian.org>

* `glib/dbus-binding-tool-glib.[ch]`: Patch based on Ricardo Kekki's patch to use an annotation `org.freedesktop.DBus.GLib.ClientCSymbol` when generating the client-side methods, instead of overloading `CSymbol` which broke everything horribly. My apologies.

```
glib/dbus-binding-tool-glib.c | 45
+++++-----
glib/dbus-binding-tool-glib.h | 1 +
2 files changed, 19 insertions(+), 27 deletions(-)
```

commit 879e2b697e2a0fd439a38ea4d4422cd8c01fefb5
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-27 15:40:36 +0000

2006-01-27 Robert McQueen <robot101@debian.org>

* `glib/dbus-gtype-specialized.[ch]`, `glib/dbus-gvalue-utils.c`:
Patch by me and Rob Taylor to add a `simple_free` function to D-Bus map and collection types, which allows those types which can be freed with a `GDestroyNotify` (such as `GHashTables` and `GArrays`, but not `GPtrArrays`) to be stored as the values in hashtables.

* `test/glib/test-dbus-glib.c`, `test/glib/test-service-glib.{c,xml}`:
Patch by Rob Taylor to add nested dicts to the glib tests to check

the above code works, and appears not to leak when called repeatedly.

```
glib/dbus-gtype-specialized.c | 33 ++++++-----
glib/dbus-gtype-specialized.h | 4 +++-
glib/dbus-gvalue-utils.c      | 50
+++++-----
3 files changed, 77 insertions(+), 10 deletions(-)
```

```
commit f467138770f3b5b771e772e7f59780a502540cb6
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-27 15:15:16 +0000
```

2006-01-27 Robert McQueen <robot101@debian.org>

```
* glib/dbus-gvalue.c (demarshal_valuearray): Patch from Rob
Taylor
to free a D-Bus allocated string with dbus_free () instead of
g_free ().
```

```
glib/dbus-gvalue.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 0c87b127b878d41d8a134fd5dee5d41a2434ef41
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-27 15:06:54 +0000
```

2006-01-27 Iain Holmes <iain@openedhand.com>

```
* glib/dbus-gproxy.c (dbus_g_proxy_dispose): Protect the dispose
method from being called multiple times.
```

```
glib/dbus-gproxy.c | 5 +++++
1 file changed, 5 insertions(+)
```

```
commit 5ad59f0f48a27d35d58e86c2cf2aa8c7d025248b
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-19 02:54:07 +0000
```

2006-01-19 Robert McQueen <robot101@debian.org>

```
* glib/dbus-binding-tool-glib.c: Patch from Rob Taylor
<rob.taylor@collabora.co.uk> to add support for generating
bindings
to arrays that are represented as GPtrArrays rather than GArrays
(ie
size-variable things, such as strings, objects, structs, etc).
```

```
glib/dbus-binding-tool-glib.c | 32 ++++++-----
1 file changed, 26 insertions(+), 6 deletions(-)
```

```
commit 0dffd5ab28874cc6453e8907f1f4cb95785748cb
```

Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-05 20:22:06 +0000

2006-01-05 Robert McQueen <robot101@debian.org>

* dbus/dbus-glib.h, glib/dbus-gproxy.c: Patch from Ricardo Kekki <ricardo.kekki@movial.fi> to make it possible to inherit from DBusGProxy, by splitting the DBusGProxy struct into a public part and a private part, and moving the setting of the DBusGProxyManager into a connection property, allowing proper GObject construction.

glib/dbus-gproxy.c | 327
+++++
1 file changed, 196 insertions(+), 131 deletions(-)

commit b356f998530ffe9762b4b0f5fd7f236eceb044db
Author: Robert McQueen <robot101@debian.org>
Date: 2006-01-05 20:03:38 +0000

2006-01-05 Robert McQueen <robot101@debian.org>

* glib/dbus-binding-tool-glib.c: Patch from Ricardo Kekki <ricardo.kekki@movial.fi> to make dbus-binding-tool heed C symbol name annotations when generating glib client bindings.

glib/dbus-binding-tool-glib.c | 42
+++++
1 file changed, 37 insertions(+), 5 deletions(-)

commit 8029bd07868baa11810b0b3ee56a8daefc0588da
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-11-30 19:32:25 +0000

* dbus/dbus-auth.c, dbus/dbus-connection.c, dbus/dbus-keyring.c, dbus/dbus-server-debug-pipe.c, glib/dbus-binding-tool-glib.c, glib/dbus-glib-tool.c, glib/dbus-gparser.c, glib/dbus-gproxy.c, test/test-segfault.c, test/test-utils.c, test/glib/test-dbus-glib.c, tools/dbus-cleanup-sockets.c, tools/dbus-launch.c, tools/dbus-tree-view.c, tools/dbus-viewer.c: Various cleanup of dead code and compiler warnings (patch from Kjartan Maraas <kmaraas@gnome.org>)

glib/dbus-binding-tool-glib.c | 2 --
glib/dbus-glib-tool.c | 4 ----
glib/dbus-gparser.c | 11 +-----
glib/dbus-gproxy.c | 2 --
4 files changed, 2 insertions(+), 17 deletions(-)

commit c5753442aacba39a3e0948aa77e523eb3fe866b
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-11-30 18:48:09 +0000

* glib/dbus-gmain.c (connection_setup_add_watch): plugged a leak
(patch from Carlos Garnacho Parro <carlosg at gnome.org>

glib/dbus-gmain.c | 1 +
1 file changed, 1 insertion(+)

commit 5d3c9bd0cd7e98df0be8351e3c210a38a978f52a
Author: Robert McQueen <robot101@debian.org>
Date: 2005-11-27 16:55:09 +0000

2005-11-27 Carlos Garcia Campos <carlosgc@gnome.org>

* glib/dbus-gobject.c: Append a GValue instead of a basic type in
method return message for property getters

```
glib/dbus-gobject.c |          41 ++++++-----  
--  
glib/dbus-gvalue.c |           8 ++++----  
glib/dbus-gvalue.h |           1 +  
3 files changed, 36 insertions(+), 14 deletions(-)
```

commit 9f2aae9a744e45c0646b36396e846031ec4ffe7f
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-11-22 20:37:00 +0000

* configure.in: Add test/name-test/Makefile to the generated
Makefile list

* dbus/dbus-shared.h (#define DBUS_NAME_FLAG_ALLOW_REPLACEMENT):
New flag which replaces DBUS_NAME_FLAG_PROHIBIT_REPLACEMENT
(#define DBUS_NAME_FLAG_DO_NOT_QUEUE): New flag for specifying
not to queue an owner if it can't be the primary owner

* bus/bus.h: Add new internal BusOwner struct

* bus/driver.c (bus_driver_handle_hello): Send flags (0 for
default)
to bus_registry_ensure and don't set the prohibit_replacement
flag

since they are now set per BusOwner and not per name.
(bus_driver_handle_list_queued_owners): bus method
(ListQueuedOwners)
that returns the list of connections in a name's connection queue

* bus/services.c (struct BusService): remove prohibit_replacement
field
(struct BusOwner): new struct for keeping track of queued

connections
and their associated flags for the queue
(struct BusRegistry): add a BusOwner memory pool
(bus_registry_new): initialize the BusOwner memory pool
(bus_registry_unref): free the BusOwner memory pool
(_bus_service_find_owner_link): new internal method for
searching the queue for a specific connection
(bus_owner_set_flags): new method for adding setting the flags
on a
bus owner
(bus_owner_new): new method that creates a BusOwner object
from the
pool and sets its flags
(bus_owner_ref, bus_owner_unref): ref counting for BusOwner
objects
(bus_registry_ensure): Add the flags parameter
(bus_registry_acquire_service): Switch from using raw connections
to
using the BusOwner struct
Add new state machine for dealing with the new set of flags
(bus_registry_set_service_context_table, struct
OwnershipCancelData,
cancel_ownership, free_ownership_cancel_data,
add_cancel_ownership_to_transaction, struct OwnershipRestoreData,
restore_ownership, free_ownership_restore_data,
add_restore_ownership_to_transaction): Switch to using BusOwner
instead of raw connections
(bus_service_add_owner): Add flags parameter
Switch to using BusOwner instead of raw connections
Add state machine for dealing with the new set of flags
(bus_service_swap_owner): Swaps the first and second owners in
the
queue. Used to make sure proper signals are sent when a
service
looses
or gains primary ownership. We never insert an owner at the
top of the
queue. Instead we insert it in the second position and then
swap.
(bus_service_remove_owner): Remove the owner from the queue
sending
out the NameLost and NameOwnerChanged signals if the we were the
primary owner
(bus_service_get_primary_owners_connection): New method that
extracts
the connection from the primary owner
(bus_service_get_primary_owner): Returns the BusOwner instead
of the
connection
(bus_service_get_allow_replacement): Changed from the old
bus_service_get_prohibit_replacement method. Checks the flags
of the

primary owner and returns if it can be replaced or not
(bus_service_set_prohibit_replacement): removed
(bus_service_has_owner): returns TRUE if and owner with
the specified connection exists in the queue

* dbus/dbus-bus.c (dbus_bus_connection_get_unique_name):
New helper
method that only compiles if tests are enabled. Allows us to
get the
unique name of a connection so we can check it against the
queue when
doing regression tests

* bus/activation.c (bus_activation_send_pending_auto_activate),
bus/dispatch.c (bus_dispatch),
bus/driver.c (bus_driver_handle_get_service_owner,
bus_driver_handle_get_connection_unix_user,
bus_driver_handle_get_connection_unix_process_id,
bus_driver_handle_get_connection_selinux_security_context),
bus/signals.c (connection_is_primary_owner):
use bus_service_get_primary_owners_connection instead of
bus_service_get_primary_owner

* dbus/dbus-sysdeps.c (_dbus_connect_unix_socket,
_dbus_listen_unix_socket): Calculate the length of the socket
path and use that instead of using a fixed length which was
causing socket names to contain many trailing Nul bytes.

* dbus/dbus-glib-lowlevel.h, glib/dbus-gobject.c
(dbus_g_method_get_sender): New method for extracting the sender
from a DBusGMethodInvocation
(dbus_g_method_return_get_reply): changed name to
dbus_g_method_get_reply
(dbus_g_method_return_send_reply): changed name to
dbus_g_method_send_reply

* doc/dbus-specification.xml: New docs that describe how the new
queueing system works and talks about the changes to the how
we specify socket names

* glib/examples/example-service.c,
glib/examples/example-signal-emitter.c,
glib/examples/statemachine/statemachine-server.c:
Changed the RequestName flags to the new system

* test/name-test/ (test-names.c, run-test.sh, Makefile.am): New
regression test suite for testing various states of the new
queueing system

```
glib/dbus-gobject.c | 28
+++++-----
glib/examples/example-service.c | 2 +-

```

```
glib/examples/example-signal-emitter.c | 2 +-
glib/examples/statemachine/statemachine-server.c | 2 +-
4 files changed, 28 insertions(+), 6 deletions(-)
```

commit 57b6ef4e8648c0c7f9b059690c4a35e73932669f

Author: Robert McQueen <robot101@debian.org>

Date: 2005-11-15 19:34:32 +0000

2005-11-15 Robert McQueen <robot101@debian.org>

* dbus/dbus-glib-lowlevel.h, glib/dbus-gobject.c: Patch from Rob Taylor to add two methods, `dbus_g_method_return_get_reply` and `dbus_g_method_return_send_reply`, to allow you to get the reply message from a `DBusGMethodInvocation`, append arbitrary stuff to it, and send it. The GLib bindings can't marshal a return value of something like `a(s)` if the array is empty - ultimately they should be made to heed the signature of the out arguments as the Python bindings now can, but this is a workable interim solution which might have other applications.

```
glib/dbus-gobject.c | 34 ++++++
1 file changed, 34 insertions(+)
```

commit cf87837198ac181b16965dcc8fb3f65bc2efa8bd

Author: Robert McQueen <robot101@debian.org>

Date: 2005-10-29 17:02:58 +0000

2005-10-29 Robert McQueen <robot101@debian.org>

* glib/Makefile.am, glib/examples/Makefile.am, glib/examples/statemachine/Makefile.am: Merge patch from Ubuntu by Daniel Stone to replace explicit calls to `libtool` with `$(LIBTOOL)`.

* test/python/.cvsignore: Add `run-with-tmp-session-bus.conf`.

* tools/dbus-monitor.1, tools/dbus-monitor.c: Merge `dbus-monitor` patch from Ubuntu by Daniel Silverstone to allow specifying match rules on the command line.

```
glib/Makefile.am | 2 +-
glib/examples/Makefile.am | 4 +--
glib/examples/statemachine/Makefile.am | 4 +--
3 files changed, 5 insertions(+), 5 deletions(-)
```

commit 3a8319b792df4ce260f8cf1e0f95fadd76942d75

Author: Ross Burton <ross@openedhand.com>
Date: 2005-10-27 16:35:43 +0000

Cleanups

```
glib/dbus-gobject.c |          1 +  
1 file changed, 1 insertion(+)
```

commit 29f0ea31b256784b4e75cdedcb822f30fe804dd3
Author: Ross Burton <ross@openedhand.com>
Date: 2005-10-25 15:57:13 +0000

Add const keywords

```
glib/dbus-gmain.c |          2 +-  
glib/dbus-gobject.c |         2 +-  
2 files changed, 2 insertions(+), 2 deletions(-)
```

commit f4b2bdf28c5e589d9161e27bf89e636c8226169f
Author: Ross Burton <ross@openedhand.com>
Date: 2005-10-25 08:54:57 +0000

Add a NoReply annotation

```
glib/dbus-binding-tool-glib.c |   53  
+++++-----  
glib/dbus-binding-tool-glib.h |    3 +-  
2 files changed, 39 insertions(+), 17 deletions(-)
```

commit adc1cd1de84b057cc858b1226522d2fde96871a0
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-10-18 04:38:04 +0000

- * glib/dbus-gvalue-utils.c (hash_free_from_gtype): handle gdouble
and G_TYPE_VALUE_ARRAY (DBUS_TYPE_STRUCT)
(gvalue_from_hash_value, hash_value_from_gvalue): handle gdouble
- * glib/dbus-gvalue.c (dbus_gvalue_to_signature): add missing
DBUS_STRUCT_BEGIN_CHAR and DBUS_STRUCT_END_CHAR charaters
when constructing struct signatures
- * python/_dbus.py (Bus): handle private connections using the
private keyword in the constructor. defaults to private=False
(Bus::close): new method to close a connection to the bus
- * python/dbus_bindings.pyx (Connection::close): renamed method
was previously called disconnect
(bus_get): now supports getting a private connection
- * python/proxies.py (ProxyMethod::__call__): check if ignore_reply

keyword is set to True. if it is, execute the method without waiting
for a reply
(ProxyObject::_introspect_execute_queue): new method for executing all the pending methods that were waiting for the introspect to finish. this is called when introspect either succeeds or fails
(ProxyObject::_introspect_error_handler): call queued methods

```
glib/dbus-gvalue-utils.c | 17 ++++++
glib/dbus-gvalue.c       | 4 +++-
2 files changed, 20 insertions(+), 1 deletion(-)
```

commit 8c16af02c159982b451289945835a86ff56805c2
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-10-05 20:43:46 +0000

* glib/dbus-gvalue.c (marshal_variant): call _dbus_gvalue_marshal instead of marshal basic so we can handle recursive types in a variant

* test/glib/test-dbus-glib.c: Add test for marshaling recursive types in variants

* test/glib/test-service-glib.c, test-service-glib.xml
(my_object_echo_variant [EchoVariant],
my_object_process_variant_of_array_of_ints123
[ProcessVariantOfArrayOfInts123]):
Add two test methods

* python/introspect_parser.py: New module for parsing introspect data.

* python/dbus_bindings.pyx:
(various places): when throwing errors fix to use errormsg instead of message local variable because Pyrex can get confused with other message variables (initial patch by Robert McQueen <robert.mcqueen at collabora.co.uk>)
(MessageIter::parse_signature_block): new method for getting the next block in a signature.
(MessageIter::append_strict): new method for appending values strictly using the passed in signature instead of guessing at the type
(MessageIter::append_dict, append_struct, append_array): use signatures to marshal children if the signature is available

* python/exceptions.py (IntrospectionParserException): new exception

* python/proxies.py (ProxyMethod::__call__): Marshal args with introspected signatures if available, else we fall back to the old way of doing things.
(ProxyObject::_introspect_reply_handler): parse introspection data

* python/service.py (ObjectType::_reflect_on_method): Properly terminate <method> if there are no args in the reflection data

* test/python/test-client.py: add tests for talking with the GLib test server. This gives us better coverage for introspection since python to python will always generate arguments as variants. It also allows us to test the robustness of the GLib bindings and interlanguage communications.

glib/dbus-gvalue.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit b489f57506320ba3e29088dc612e27112c6a1854
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-09-26 22:26:38 +0000

* glib/dbus-glib-tool.c: removed extra comma at the end of the DBusBindingOutputMode enum which was causing a warning. #include <time.h> so using time_t is explicitly defined

glib/dbus-glib-tool.c | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)

commit a32ca0058020e7e23d72d9ad69fe881ba8b98e93
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-09-06 22:38:54 +0000

* Released 0.50

* Patch from Steve Grubb:

- bus/activation.c (bus_activation_service_reload_test): clean up some indentation
- dbus/dbus-keyring.c (_dbus_keyring_reload): fix conditional
- dbus/dbus-message-factory.c (generate_special): fix a couple of buffer overflows in the test suite. This is non critical because it can not be exploited and this code is only run when doing a make check.

* Patch from Yaakov Selkowitz: Build fixes for Cygwin

- configure.in: Don't check and link against kdeccore, only qt headers
- dbus/Makefile.am: Add -no-undefined to libdbus_1_la_LDFLAGS
- gcj/org/freedesktop/dbus/Makefile.am:

```
add libdbus_gcj_1_la_LDFLAGS = -no-undefined
- glib/Makefile.am: Add -no-undefined to libdbus_glib_1_la_LDFLAGS
and $(DBUS_GLIB_LIBS) to dbus_binding_tool_LDADD
- qt/Makefile.am: Add -no-undefined to libdbus_qt_1_la_LDFLAGS
- tools/Makefile.am: Add platform extentions to binaries
(i.e. .exe on windows)
```

```
* configure.in:
- Make it so if no suitable version of python is found we only
disable building python instead of exiting the configure script
- Require version 2.4 of glib for glib bindings
- Up version to 0.50
```

```
* python/__init__.py: Sync version with libdbus to (0,50,0)
```

```
glib/Makefile.am | 6 +++--
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
commit 23fb630c1e907187c8d7aa5539b30a34e7cc9632
Author: Colin Walters <walters@verbum.org>
Date: 2005-08-26 02:00:37 +0000
```

```
2005-08-25 Colin Walters <walters@verbum.org>
```

```
* glib/dbus-gproxy.c (dbus_g_proxy_call): Doc update, thanks
to Ryan Lortie for the suggestion.
```

```
glib/dbus-gproxy.c | 3 +-
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
commit 8e13d06500c5b4df219891fc4b5b8242cb9f5cae
Author: Ross Burton <ross@openedhand.com>
Date: 2005-08-17 17:30:45 +0000
```

```
Unref message and protect against NULL
```

```
glib/dbus-gproxy.c | 5 ++++
1 file changed, 4 insertions(+), 1 deletion(-)
```

```
commit 03dc49be2e41245f107e71194ebef428d4d4ccf7
Author: Colin Walters <walters@verbum.org>
Date: 2005-08-16 21:26:12 +0000
```

```
2005-08-16 Colin Walters <walters@verbum.org>
```

```
* glib/dbus-gobject.c (dbus_set_g_error): Don't lose if the
DBusError message is NULL.
```

```
glib/dbus-gobject.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 20ab99c264aa730de304f637a7e4eacec3e1686d
```

Author: Colin Walters <walters@verbum.org>
Date: 2005-08-04 15:49:30 +0000

2005-08-04 Colin Walters <walters@verbum.org>

```
* glib/dbus-gvalue-utils.h
(_dbus_g_type_specialized_builtins_init)
(dbus_g_type_is_fixed, dbus_g_type_fixed_get_size)
(dbus_gvalue_set_from_pointer, dbus_g_hash_table_value_foreach)
(dbus_g_hash_table_insert_values,
dbus_g_hash_table_insert_steal_values)
(dbus_gtype_is_valid_hash_key, dbus_gtype_is_valid_hash_value)
(dbus_g_hash_func_from_gtype, dbus_g_hash_free_from_gtype)
(dbus_g_hash_equal_from_gtype, dbus_gvalue_stor,
dbus_gvalue_take):
* glib/dbus-gvalue.h (dbus_g_value_types_init)
(dbus_gvalue_demarshal, dbus_gvalue_demarshal_variant)
(dbus_gvalue_demarshal_message, dbus_gvalue_marshal):
```

Prefix name with _ to ensure they're not exported. All callers updated.

```
* glib/dbus-gvalue.c (typecode_to_gtype)
(dbus_typecode_maps_to_basic, basic_typecode_to_gtype)
(signature_iter_to_g_type_dict)
(signature_iter_to_g_type_array)
(dbus_gtype_from_signature_iter, dbus_gtype_from_signature)
(dbus_gtypes_from_arg_signature):
Move to dbus-gsignature.c.
```

```
* glib/dbus-binding-tool-glib.c
(dbus_binding_tool_output_glib_server): Call
dbus_g_type_specialized_builtins_init instead of
dbus_g_value_types_init.
(dbus_binding_tool_output_glib_client): Ditto.
```

```
* glib/Makefile.am (DBUS_GLIB_INTERNALS): Add dbus-gsignature.c
and dbus-gsignature.h
```

```
* test/glib/test-service-glib.c (my_object_rec_arrays): Delete
unused variable.
```

```
glib/Makefile.am          |    5 +-
glib/dbus-binding-tool-glib.c |   36 ++++---
glib/dbus-gmain.c         |   23 +++-
glib/dbus-gobject.c       |   35 ++++---
glib/dbus-gproxy.c        |   13 +-
glib/dbus-gsignature.c    |  189 ++++++
glib/dbus-gsignature.h    |   19 ++++
glib/dbus-gvalue-utils.c  |   46 ++++----
glib/dbus-gvalue-utils.h  |   28 ++---
glib/dbus-gvalue.c        |  237
```



```
+++++-----
glib/dbus-gvalue.h      |    21 +--
11 files changed, 350 insertions(+), 302 deletions(-)
```

```
commit f19bb3b8b599f006e51a1ff81447ac6470c249fc
Author: Colin Walters <walters@verbum.org>
Date: 2005-08-03 23:21:49 +0000
```

```
2005-08-03 Colin Walters <walters@verbum.org>
```

```
* glib/dbus-gobject.c: Add tests on hardcoded object info;
this should
catch any incompatible changes accidentally made.
```

```
glib/dbus-gobject.c |    150
+++++
1 file changed, 150 insertions(+)
```

```
commit f902eead6b209181f7a26b4cb1a967661ac86117
Author: Colin Walters <walters@verbum.org>
Date: 2005-08-01 16:12:53 +0000
```

```
2005-08-01 Colin Walters <walters@verbum.org>
```

```
Patch from Joe Markus Clarke:
```

```
* glib/dbus-gidl.c (property_info_unref, arg_info_unref): Fix
use-after-free.
```

```
glib/dbus-gidl.c |     4 +--
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
commit db1be2d67b4239496940080f44c7e05eed42a610
Author: Colin Walters <walters@verbum.org>
Date: 2005-08-01 16:07:19 +0000
```

```
2005-08-01 Colin Walters <walters@verbum.org>
```

```
Patch from Joe Markus Clarke:
```

```
* glib/dbus-gvalue.c (dbus_g_value_types_init):
* glib/dbus-gvalue-utils.c
(dbus_g_type_specialized_builtins_init)
* glib/dbus-gobject.c (write_interface):
```

```
Don't use C99 style initializers (bug #3933).
```

```
glib/dbus-gobject.c |     2 +-
glib/dbus-gvalue-utils.c |    7 +++----
glib/dbus-gvalue.c |     8 ++++----
3 files changed, 8 insertions(+), 9 deletions(-)
```

commit 04045b587a62e4b5cafbad0b81e18761c28a41e5
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-07-31 02:09:15 +0000

2005-07-30 Havoc Pennington <hp@redhat.com>

* fix a bunch of Doxygen warnings and mistakes

```
glib/dbus-glib.c      |          4 +---  
glib/dbus-gobject.c  |          2 +-  
glib/dbus-gproxy.c   |          4 +---  
3 files changed, 5 insertions(+), 5 deletions(-)
```

commit f91e8613d913ee943afa5bbb6cdafb2ce062c6ef
Author: Ross Burton <ross@openedhand.com>
Date: 2005-07-25 19:16:22 +0000

Remove matchers when all proxies are removed

```
glib/dbus-gproxy.c | 6 ++++++  
1 file changed, 6 insertions(+)
```

commit 2c755e157a9b41451efd5fd88e74c37501e21127
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-24 18:04:23 +0000

2005-07-24 Colin Walters <walters@verbum.org>

* glib/dbus-gvalue.c (signature_iter_to_g_type_array): Don't
require
typedata; recursive arrays won't have it.

* test/glib/test-dbus-glib.c:
* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml: Add recursive arrays tests.

```
glib/dbus-gvalue.c | 5 -----  
1 file changed, 5 deletions(-)
```

commit 6bf40ec5a63f7703070f951130d8e78cb044ec66
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-07-15 20:28:05 +0000

* glib/Makefile.am: Add make-dbus-glib-error-switch.sh to
EXTRA_DIST
so distcheck doesn't fail

* glib/examples/Makefile.am: Add example-service.xml and
example-signal-emitter.xml to EXTRA_DIST so distcheck doesn't
fail

* glib/examples/statemachine/Makefile.am: Add statemachine.xml and

statemachine-server.xml to EXTRA_DIST so distcheck doesn't fail

* python/Makefile.am: Prepend \$(srcdir)/ to source files so the compiler looks in the right places during distcheck

```
glib/Makefile.am | 2 +-
glib/examples/Makefile.am | 2 ++
glib/examples/statemachine/Makefile.am | 2 +-
3 files changed, 4 insertions(+), 2 deletions(-)
```

commit a447f257bd0b557b661be74f2d63ba0d31366a46
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-07-15 19:15:05 +0000

* glib/example/Makefile.am: Fix a typo which cause make distcheck to fail

```
glib/examples/Makefile.am | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit ce277d89860aaf05064d11eb24878cbf41326ff2
Author: Ross Burton <ross@openedhand.com>
Date: 2005-07-15 13:25:48 +0000

Don't leak messages when calling methods

```
glib/dbus-gproxy.c | 1 +
1 file changed, 1 insertion(+)
```

commit 877add82e329e1de4ab7b4d99084362e250fa34c
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-12 17:57:04 +0000

2005-07-12 Colin Walters <walters@verbum.org>

* glib/examples/statemachine/Makefile.am
(statemachine-server-glue.h)
(statemachine-glue.h):
* glib/examples/Makefile.am (example-service-glue.h)
(example-signal-emitter-glue.h):
* glib/Makefile.am (dbus-glib-error-switch.h):
Add libtool --mode=execute so we use the built library instead
of any installed one.

```
glib/Makefile.am | 2 +-
glib/examples/Makefile.am | 4 +++-
glib/examples/statemachine/Makefile.am | 4 +++-
3 files changed, 5 insertions(+), 5 deletions(-)
```

commit 31ca9e74c599c17f802698ad8184f9ae654983c5
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-11 16:12:49 +0000

2005-07-11 Colin Walters <walters@verbum.org>

```
* glib/dbus-gvalue.c (struct _DBusGValue): Delete.
(dbus_g_value_types_init): Remove assertion.
(dbus_g_value_get_g_type, dbus_g_value_open)
(dbus_g_value_iterator_get_values, dbus_g_value_get_signature)
(dbus_g_value_copy, dbus_g_value_free): Delete unimplemented
functions related to DBusGValue. Now we marshal/demarshal
structures as GValueArray.
(dbus_gtype_from_signature_iter): Return G_TYPE_VALUE_ARRAY for
structures.
(signature_iter_to_g_type_array): Don't call
signature_iter_to_g_type_struct.
(signature_iter_to_g_type_struct): Delete.
(dbus_gvalue_to_signature): Delete.
(dbus_gvalue_to_signature): New function with same name as other
one; we can convert structures to signatures.
(demarshal_valuearray): New function.
(get_type_demarshaller): Use it.
(demarshal_recurse): Delete.
(marshal_proxy): New function.
(marshal_map): Warn if we can't determine signature from type.
(marshal_collection_ptrarray): Ditto.
(marshal_collection_array): Ditto.
(get_type_marshaller): Use marshal_valuearray.
(marshal_recurse): Delete.
(_dbus_gvalue_test): Add some tests.

* dbus/dbus-glib.h (struct _DBusGValueIterator):
(dbus_g_value_get_g_type, DBUS_TYPE_G_VALUE)
(dbus_g_value_open, dbus_g_value_iterator_get_value)
(dbus_g_value_iterator_get_values, dbus_g_value_iterator_recurse)
(dbus_g_value_free): Remove prototypes.

* glib/dbus-binding-tool-glib.c
(dbus_g_type_get_lookup_function):
Handle
G_TYPE_VALUE_ARRAY.

* glib/examples/example-service.c:
* glib/examples/example-client.c: Implement GetTuple.

* test/glib/test-dbus-glib.c:
* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml: Add structure tests.

glib/dbus-binding-tool-glib.c | 1 +
glib/dbus-gvalue.c | 358
+++++-----
glib/examples/example-client.c | 32 +++-
glib/examples/example-service.c | 21 +-

```

4 files changed, 209 insertions(+), 203 deletions(-)

commit 5c39fa43d8eb87284abe7d009349523ad5398db0

Author: Colin Walters <walters@verbum.org>

Date: 2005-07-10 22:54:18 +0000

2005-07-10 Colin Walters <walters@verbum.org>

- * doc/TODO: Knock off some GLib items with this patch.
- * glib/dbus-gvalue-utils.c (`_dbus_gtype_can_signal_error`) (`_dbus_gvalue_signals_error`): New functions.
- * glib/dbus-gvalue-utils.h: Prototype them.
- * glib/dbus-gobject.c (`arg_iterate`): Update to handle return vals and change to not output const/retval flags for input args. All callers updated.
(`invoke_object_method`): Refactor to handle return values. Add some more comments in various places. Remove debug `g_print`.
- * glib/dbus-binding-tool-glib.h (`DBUS_GLIB_ANNOTATION_RETURNVAL`): New.
- * glib/dbus-binding-tool-glib.c (`dbus_g_type_get_marshall_name`): Handle `G_TYPE_NONE`.
(`compute_gsignature`): New function; refactored from code from `compute_marshall` and `compute_marshall_name`. Enhance to handle return values and async ops more cleanly. Update for async ops returning `NONE` instead of `BOOLEAN`.
(`compute_marshall`, `compute_marshall_name`): Call `compute_gsignature` and output appropriate string.
(`generate_glue`): Handle return value annotation. Also don't dump constness flag for input arguments.
- * glib/Makefile.am (`DBUS_GLIB_INTERNALS`): New variable; contains files shared between installed library and utilities.
(`libdbus_glib_la_SOURCES`): Move some stuff into `DBUS_GLIB_INTERNALS`.
(`libdbus_gtool_la_SOURCES`): Suck in `DBUS_GLIB_INTERNALS` so the binding tool can access gtype utility functions.
- * test/glib/test-service-glib.c:
- * test/glib/test-service-glib.xml:
- * test/glib/test-dbus-glib.c: Add some tests for return values.

```
glib/Makefile.am          | 19 +-
glib/dbus-binding-tool-glib.c      | 324
+++++-----
glib/dbus-binding-tool-glib.h      | 1 +
glib/dbus-gobject.c          | 242 ++++++-----
```

```
glib/dbus-gvalue-utils.c      | 54 ++++++
glib/dbus-gvalue-utils.h      | 4 +
glib/examples/statemachine/.cvsignore | 1 +
7 files changed, 479 insertions(+), 166 deletions(-)
```

commit 00de44640de45aa6068ca9d1b42deb801b430320
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-09 18:54:45 +0000

2005-07-09 Colin Walters <walters@verbum.org>

* glib/dbus-gparser.c (parse_annotation): Add annotations to argument if available, not method.

* glib/dbus-gobject.c (arg_iterate): More verbose warnings. (invoke_object_method): First, remove some redundant GValues (object_value, error_value) in favor of working on array directly. Second, rework constness to be less buggy. Now we iterate directly over the argument metadata instead of parallel iterating over output signature and metadata.

* glib/dbus-glib-tool.h: Add INVALID_ANNOTATION error.

* glib/dbus-binding-tool-glib.c (generate_glue): Barf on const annotation on input args.

```
glib/dbus-binding-tool-glib.c | 17 ++++++++--
glib/dbus-glib-tool.h         | 3 +-
glib/dbus-gobject.c           | 54
+++++++-----
glib/dbus-gparser.c           | 7 +++--
4 files changed, 52 insertions(+), 29 deletions(-)
```

commit 91dc6d18c4ccc69fef3c12e6f628f7458c1d4511
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-09 17:52:51 +0000

2005-07-09 Colin Walters <walters@verbum.org>

* glib/dbus-binding-tool-glib.h (DBUS_GLIB_ANNOTATION_CONST): Define.

* glib/dbus-binding-tool-glib.c (generate_glue): Handle Const annotation.

* glib/dbus-gobject.c (arg_iterate): Update to parse constval too.

(method_dir_signature_from_object_info): Handle arg_iterate change.

(write_interface): Ditto.

(lookup_object_info): Don't barf if format_version is > 0.

(invoke_object_method): Handle arg constness.

- * glib/dbus-gidl.c (struct ArgInfo): Add annotations.
 (arg_info_new): Create.
 (arg_info_unref): Destroy.
 (arg_info_get_annotations, arg_info_get_annotation)
 (arg_info_add_annotation): New functions.
- * glib/dbus-gidl.h: Prototype them.
- * glib/dbus-gparser.c (parse_annotation): Allow annotations in
 args, disallow them in properties.
 (parse_annotation): Handle arg annotations.
- * test/glib/test-service-glib.xml:
- * test/glib/test-service-glib.c: Update to make some methods
 const.

```

glib/dbus-binding-tool-glib.c |    6 ++++++
glib/dbus-binding-tool-glib.h |    1 +
glib/dbus-gidl.c              |   30 ++++++
glib/dbus-gidl.h              |    7 ++++++
glib/dbus-gobject.c           |   48
+++++-----
glib/dbus-gparser.c           |    5 ++++-
6 files changed, 89 insertions(+), 8 deletions(-)

```

```

commit 52438e8f3f67fa96ffd9607bad34968875d90226
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-09 01:46:51 +0000

```

2005-07-08 Colin Walters <walters@verbum.org>

- * test/glib/test-service-glib.xml:
- * test/glib/test-service-glib.c:
- * test/glib/test-dbus-glib.c: Test a{sv}.
- * glib/examples/statemachine/statemachine.c:
- * glib/examples/statemachine/statemachine-server.c:
- * glib/examples/statemachine/statemachine-client.c: Fix some
 bugs,
 add progress bar, etc.
- * glib/dbus-gvalue.c (register_array, register_dict): Delete; not
 needed anymore due to generic array/map marshalling.
 (dbus_g_value_types_init): Don't register basic arrays or the
 string/string hash.
 (dbus_gtype_from_signature_iter): Don't try to recurse into
 variants.
 (dbus_gtype_to_signature): Check collection/map before type
 metadata.
 (demarshal_garray_basic): Renamed to demarshal_collection_array.
 (demarshal_ghashtable): Renamed to demarshal_map; fix to use new

generic map creation/append functions instead of hash table specifically.
(get_type_demarshaller): Handle maps.
(demarshal_collection): Dispatch on collection type to either demarshal_collection_ptrarray or demarshal_collection_array.
(get_type_marshall): Handle maps.
(marshal_collection): Dispatch collection type to either marshal_collection_ptrarray or marshal_collection_array.
(_dbus_gvalue_test): New test.

* glib/dbus-gvalue-utils.c (unset_and_free_g_value): New function.

(hash_free_from_gtype): Use it to free GValues.
(hashtable_append): New function.
(ptrarray_append): Fix prototype.
(slist_append): Ditto.
(_dbus_gvalue_utils_test): Extend tests.

* glib/dbus-gtype-specialized.c

(dbus_g_type_specialized_init_append): Renamed from dbus_g_type_specialized_collection_init_append. Remove const from value, since we steal it.

(dbus_g_type_specialized_map_append): New function.

* glib/dbus-gtype-specialized.h: Update prototypes. Add DBusGTypeSpecializedMapAppendFunc.

* glib/dbus-gtest.c (dbus_glib_internal_do_not_use_run_tests): Run _dbus_gvalue_test.

* glib/dbus-gtest.h: Prototype it.

```
glib/dbus-gtest.c           |    4 +
glib/dbus-gtest.h           |    1 +
glib/dbus-gtype-specialized.c |   13 +-
glib/dbus-gtype-specialized.h |   15 +-
glib/dbus-gvalue-utils.c    |  116 ++++++-
glib/dbus-gvalue.c          |   402
+++++++-----
glib/examples/statemachine/statemachine-client.c |  192 ++++++++--
glib/examples/statemachine/statemachine-server.c |   27 ++
glib/examples/statemachine/statemachine.c       |   44 +--
9 files changed, 555 insertions(+), 259 deletions(-)
```

commit 6758d07f4dd3bee533488381c8cf236d9d99a5f0

Author: Ross Burton <ross@openedhand.com>

Date: 2005-07-08 17:02:42 +0000

Make async bindings work again


```
glib/dbus-binding-tool-glib.c | 31 ++++++-----  
1 file changed, 13 insertions(+), 18 deletions(-)
```

```
commit 8068477b92f2ee5ca19623e033e335d4b16d8de5  
Author: Colin Walters <walters@verbum.org>  
Date: 2005-07-08 16:25:28 +0000
```

```
2005-07-08 Colin Walters <walters@verbum.org>
```

```
* tools/Makefile.am: Kill of print-introspect in favor of using  
dbus-send --print-reply=literal.  
* tools/print-introspect.c: Deleted.
```

```
* test/glib/test-service-glib.xml:  
* test/glib/test-service-glib.c (my_object_get_objs): New test  
for "ao".
```

```
* test/glib/test-dbus-glib.c (echo_received_cb): Free echo data.  
(main): Test GetObjs.
```

```
* glib/examples/statemachine/Makefile.am:  
* glib/examples/statemachine/sm-marshal.list:  
* glib/examples/statemachine/statemachine-client.c:  
* glib/examples/statemachine/statemachine-server.c:  
* glib/examples/statemachine/statemachine-server.xml:  
* glib/examples/statemachine/statemachine.c:  
* glib/examples/statemachine/statemachine.h:  
* glib/examples/statemachine/statemachine.xml:
```

```
New example.
```

```
* glib/examples/example-service.c (main): Move invocation  
of dbus_g_object_type_install_info earlier, to emphasize it  
should only be done once.
```

```
* glib/examples/example-signal-emitter.c (main): Ditto.
```

```
* glib/examples/Makefile.am (SUBDIRS): Include statemachine.
```

```
* glib/dbus-gvalue.h (dbus_gtype_to_signature)  
(dbus_gvalue_marshal): Update prototypes.
```

```
* glib/dbus-gvalue.c: Update all marshalling functions to take  
const GValue instead of GValue.  
(signature_iter_to_g_type_array): Return a GPtrArray for nonfixed  
types.  
(dbus_gvalue_to_signature): Update for dbus_gtype_to_signature  
change.  
(dbus_gtype_to_signature): Handle generic collecitons and maps.  
Return a newly-allocated string.  
(demarshal_proxy, demarshal_object_path, demarshal_object)
```

```

        (demarshal_strv, demarshal_ghashtable): Set error, don't assert
if
we get the wrong types from message.
(get_type_demarshaller): New function, extracted from
dbus_gvalue_demarshal.
(demarshal_collection): New function, demarshals generic
collection.
(dbus_gvalue_demarshal): Just invoke result of
get_type_demarshaller.      Throw error if we don't have one.
(marshal_garray_basic): Abort on OOM.
(get_type_marshaller): New function, extracted from
dbus_gvalue_marshal.
(collection_marshal_iterator, marshal_collection): New functions;
implements generic marshalling for an iterable specialized
collection.
(dbus_gvalue_marshal): Just invoke result of get_type_marshaller.

* glib/dbus-gvalue-utils.c (gvalue_from_ptrarray_value): Handle
G_TYPE_STRING.
(ptrarray_value_from_gvalue): Ditto.
(ptrarray_append, ptrarray_free): New functions.
(slist_constructor, slist_iterator, slist_copy_elt, slist_copy)
(slist_append, slist_end_append, slist_free): New functions.
(dbus_g_type_specialized_builtins_init): Add append fuctions
for GPtrArray and GSList.  Register GSList.
(test_specialized_hash, _dbus_gvalue_utils_test): New functions.

* glib/dbus-gtype-specialized.h
(DBusGTypeSpecializedAppendContext):
New.
(dbus_g_type_specialized_collection_init_append)
(dbus_g_type_specialized_collection_append)
(dbus_g_type_specialized_collection_end_append): Prototype.
(DBusGTypeSpecializedCollectionVtable): Add append_func and
end_append_func.

* glib/dbus-gtype-specialized.c
(dbus_g_type_specialized_collection_init_append)
(dbus_g_type_specialized_collection_append)
(dbus_g_type_specialized_collection_end_append): New functions.
(dbus_g_type_map_value_iterate): Take const GValue.
(dbus_g_type_collection_value_iterate): Ditto.

* glib/dbus-gtest.c (dbus_glib_internal_do_not_use_run_tests):
Run
_dbus_gvalue_utils_test.

* glib/dbus-gtest.h: Prototype it.

* glib/dbus-gproxy.c (dbus_g_proxy_manager_filter): Avoid
using uninitialized owner_list.
(dbus_g_proxy_begin_call_internal): Move return_if_fail to

```

public API.
(dbus_g_proxy_end_call_internal): Update to use error set
from dbus_gvalue_demarshal instead of setting it here.
(dbus_g_proxy_begin_call): Move return_if_fail here.

* glib/dbus-gobject.c (write_interface): Update for
dbus_gtype_to_signature returning new string.

* configure.in: Add glib/examples/statemachine.

```

glib/dbus-gobject.c          |  6 +-
glib/dbus-gproxy.c          | 31 +-
glib/dbus-gtest.c           |  7 +
glib/dbus-gtest.h           |  1 +
glib/dbus-gtype-specialized.c | 44 +-
glib/dbus-gtype-specialized.h | 32 +-
glib/dbus-gvalue-utils.c    | 261 ++++++++
glib/dbus-gvalue-utils.h    |  1 -
glib/dbus-gvalue.c          | 388 ++++++++
glib/dbus-gvalue.h          |  4 +-
glib/examples/Makefile.am   |  2 +
glib/examples/example-service.c |  4 +-
glib/examples/example-signal-emitter.c |  4 +-
glib/examples/statemachine/.cvsignore | 15 +
glib/examples/statemachine/Makefile.am | 35 ++
glib/examples/statemachine/sm-marshal.list | 1 +
glib/examples/statemachine/statemachine-client.c | 540
+++++++
glib/examples/statemachine/statemachine-server.c | 202 ++++++
glib/examples/statemachine/statemachine-server.h | 37 ++
glib/examples/statemachine/statemachine-server.xml | 14 +
glib/examples/statemachine/statemachine.c | 351 ++++++++
glib/examples/statemachine/statemachine.h | 77 +++
glib/examples/statemachine/statemachine.xml | 33 ++
23 files changed, 1983 insertions(+), 107 deletions(-)

```

```

commit 8b39e9d32cb505d4f9550ca24627020e5edd9681
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-06 21:27:45 +0000

```

2005-07-06 Colin Walters <walters@verbum.org>

```

* dbus/dbus-glib.h (DBusGPendingCall, DBusGPendingCallNotify)
(DBUS_TYPE_G_PENDING_CALL, dbus_g_pending_call_get_g_type)
(dbus_g_pending_call_ref, dbus_g_pending_call_unref): Delete.
(dbus_g_pending_call_set_notify, dbus_g_pending_call_cancel):
Delete in favor of dbus_g_proxy_begin_call and
dbus_g_proxy_cancel_call.
(DBusGProxyCall, DBusGProxyCallNotify): New.
(dbus_g_proxy_begin_call): Change prototype to take callback,
user
data, and destroy function. This replaces

```

```

dbus_g_pending_call_set_notify.
(dbus_g_proxy_cancel_call): Prototype.
(DBusGAsyncData): Delete, shouldn't be needed anymore.

* glib/dbus-gproxy.c (struct _DBusGProxy): Add call_id_counter
and
pending_calls map.
(struct _DBusGProxyManager): Add bus_proxy member, which is an
internal proxy for calls to the bus. Remove
pending_nameowner_calls, now the internal proxy keeps track.
(dbus_g_proxy_manager_unref): Unref bus proxy, remove reference
to
pending_nameowner_calls.
(got_name_owner_cb): Update prototype, and use
dbus_g_proxy_end_call.
(got_name_owner_cb): Remove reference to pending_nameowner_calls.
(dbus_g_proxy_manager_register): Delete directly libdbus code in
favor of using internal proxy.
(dbus_g_proxy_manager_unregister): Update to use
dbus_g_proxy_cancel_call for any pending GetNameOwner call.
(dbus_g_proxy_init): Initialize pending calls map.
(dbus_g_proxy_constructor): New.
(dbus_g_proxy_class_init): Add get/set property functions,
constructor, and add NAME, PATH, and INTERFACE properties.
(cancel_pending_call): New function.
(dbus_g_proxy_dispose): Iterate over any outstanding calls and
cancel them.
(dbus_g_proxy_set_property, dbus_g_proxy_get_property): New.
(GPendingNotifyClosure): New structure.
(d_pending_call_notify, d_pending_call_free): Moved here from
dbus-glib.c.
(DBUS_G_VALUE_ARRAY_COLLECT_ALL): Moved around to satisfy
function
ordering.
(manager_begin_bus_call): New internal function for talking to
internal bus proxy.
(dbus_g_proxy_new): Construct object using GObject properties.
(dbus_g_proxy_begin_call_internal): Update to take user data,
etc.
Create closure of same, and insert call into map of pending
calls.
(dbus_g_proxy_end_call_internal): Take call id instead of pending
call. Look up pending call in current set. Remove it when we've
completed.
(dbus_g_pending_call_end, dbus_g_proxy_end_call_internal):
Delete.
(dbus_g_proxy_begin_call): Change API to take callback, user
data,
and destroy function directly.
(dbus_g_proxy_end_call): Update to take DBusGProxyCall.
(dbus_g_proxy_call): Invoke with NULL callback.
(dbus_g_proxy_cancel_call): New function, replaces

```

dbus_g_pending_call_cancel.

- * glib/dbus-gparser.c (validate_signature): Fix call to dbus_set_g_error.
- * glib/dbus-gobject.c (dbus_g_object_type_dbus_metadata_quark): New quark for attaching metadata to GType.
 - (info_hash): Delete.
 - (lookup_object_info): Look up using quark.
 - (dbus_g_object_type_install_info): Check that a type is classed, not that it's an object. Also just install type data using quark instead of using global hash.
- * glib/dbus-glib.c (dbus_g_pending_call_ref) (dbus_g_pending_call_unref, dbus_pending_call_get_g_type) (GPendingNotifyClosure): Delete.
 - (d_pending_call_notify, d_pending_call_free): Move to dbus-gproxy.c.
 - (dbus_g_pending_call_set_notify, dbus_g_pending_call_cancel): Delete.
- * glib/dbus-binding-tool-glib.c (generate_client_glue): Disable async client method generation until we can fix it...
- * tools/dbus-viewer.c (load_child_nodes): Use dbus_g_proxy_call. (load_from_service_thread_func): Ditto.
- * tools/dbus-names-model.c (struct NamesModel): Hold DBusGProxyCall.
 - (have_names_notify): Update prototype, use dbus_g_proxy_cancel_call.
 - (names_model_reload): Update for new dbus_g_proxy_begin_call API.
- * tools/dbus-monitor.c (filter_func): Update for print_message API change.
- * test/glib/test-dbus-glib.c: Add more tests for async invocations. Update many begin_call/end_call pairs to just use dbus_g_proxy_call.
- * tools/dbus-send.c (main): Add --print-reply=literal mode. This allows us to dump print-introspect.c.
- * tools/dbus-print-message.h (print_message): Add literal argument to print_message which is intended to allow printing arguments without metadata like "string=".
- * tools/dbus-print-message.c (print_iter): Add literal argument. (print_message): Allow printing string messages literally.

```
glib/dbus-binding-tool-glib.c |    2 +
glib/dbus-glib.c              |   140 -----
glib/dbus-gobject.c          |    44 ++--
glib/dbus-gparser.c          |     2 +-
glib/dbus-gproxy.c           |   555
+++++-----
5 files changed, 390 insertions(+), 353 deletions(-)
```

commit 462a36e4898da053ba749857511f2d634dbea665
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-05 16:39:56 +0000

2005-07-05 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (marshal_dbus_message_to_g_marshaller):
Remove value refcount leak, original patch from Jorn Baayen
<jorn@openedhand.com>. Also remove useless extra value in
favor
of prepending to value array directly.

```
glib/dbus-gproxy.c |    7 +++----
1 file changed, 3 insertions(+), 4 deletions(-)
```

commit d80b7960d7d189d7c33e6d230fdbbf8b69546f60
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-02 06:05:23 +0000

2005-07-02 Colin Walters <walters@verbum.org>

* glib/dbus-gmain.c (_dbus_gmain_test): Fix test.

```
glib/dbus-gmain.c |    4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit c7f14e3c337601053b08dc07a186be81063fc005
Author: Colin Walters <walters@verbum.org>
Date: 2005-07-01 15:44:12 +0000

2005-07-01 Colin Walters <walters@verbum.org>

Patch from Jonathan Matthew <jonathan@kaolin.hn.org>

* glib/dbus-gvalue.c (basic_typecode_to_gtype): Fix return type.
(dbus_g_value_types_init): Marshal G_TYPE_CHAR as DBUS_TYPE_BYTE,
G_TYPE_LONG as DBUS_TYPE_INT32, G_TYPE_ULONG as DBUS_TYPE_UINT32,
and G_TYPE_FLOAT as DBUS_TYPE_DOUBLE.

```
glib/dbus-gvalue.c |   33 +++++-----
1 file changed, 31 insertions(+), 2 deletions(-)
```

commit 5c82c5ce0eba4ec3bbb0454f3987f56de968bf95

Author: Colin Walters <walters@verbum.org>
Date: 2005-06-30 18:22:09 +0000

2005-06-30 Colin Walters <walters@verbum.org>

- * test/glib/test-dbus-glib.c:
- * test/glib/test-service-glib.c:
- * test/glib/test-service-glib.xml: Update tests for new error setting bits, also add async tests (patch from Ross Burton).

- * test/glib/Makefile.am (test_service_glib_LDADD): Add DBUS_GLIB_THREADS_LIBS.

- * glib/dbus-gproxy.c (get_name_owner) (dbus_g_pending_call_end_valist): Ditto.

- * glib/dbus-gobject.c (error_metadata): New mapping from GError domain (GQuark) to DBusGErrorInfo.
(gerror_domaincode_to_dbus_error_name): Attempt to look up error quark in error_metadata. Take message interface as default error message interface.
(gerror_to_dbus_error_message): Pass message interface.
(dbus_set_g_error): Resurrected.
(dbus_g_error_info_free): New function.
(dbus_g_object_type_install_info): Use g_type_class_ref instead of _peek to actually create the object class if it hasn't been created yet.
(dbus_g_error_domain_register): New function.

- * glib/dbus-gmain.c (dbus_g_bus_get): Switch to dbus_set_g_error.

- * glib/dbus-gparser.c (validate_signature): Ditto.

- * dbus/dbus-glib.h (dbus_g_error_set): Delete.
(dbus_g_error_domain_register): Prototype.

- * glib/dbus-glib.c (dbus_g_error_set): Delete.
Update tests.

```
glib/dbus-glib.c      |      45 +-----  
glib/dbus-gmain.c    |       3 +-  
glib/dbus-gobject.c  |     127  
+++++  
glib/dbus-gparser.c  |       3 +-  
glib/dbus-gproxy.c   |       6 +--  
5 files changed, 134 insertions(+), 50 deletions(-)
```

commit f879e3e832d76a7b3b02cc7d25a3e5b9ef195f42
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-29 23:52:35 +0000

2005-06-29 Colin Walters <walters@verbum.org>

* dbus/dbus-glib.h: Delete DBUS_TYPE_G_PROXY_ARRAY. Add DBUS_TYPE_G_OBJECT_PATH.

* glib/dbus-gvalue.c (dbus_g_value_types_init): Remove
marshallers
for G_TYPE_OBJECT and DBUS_TYPE_G_PROXY_ARRAY (the latter should
be handled more generically). Add DBUS_TYPE_G_OBJECT_PATH.
(dbus_g_object_path_get_g_type): New function.
(dbus_gtype_from_signature_iter): Map DBUS_TYPE_G_OBJECT_PATH
to DBUS_TYPE_G_OBJECT_PATH by default.
(demarshal_proxy): Remove unused name variable.
(demarshal_object_path, marshal_object_path): New functions.
(demarshal_proxy_array, marshal_proxy_array): Delete.

* glib/dbus-binding-tool-glib.c (dbus_g_type_get_c_name): Map
DBUS_TYPE_G_OBJECT_PATH to char *.
(dbus_g_type_get_lookup_function): Map builtin
DBUS_TYPE_G_OBJECT_PATH.

* test/glib/test-dbus-glib.c
* test/glib/test-service-glib.c (my_object_objpath):
Adapt tests to new object path marshalling.

```
glib/dbus-binding-tool-glib.c |    5 +-  
glib/dbus-gvalue.c           |   164  
+++++-----  
2 files changed, 68 insertions(+), 101 deletions(-)
```

commit 1f87d5d3363d1e23208c993eabb3cb1f0341b4d9
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-29 17:02:33 +0000

2005-06-29 Colin Walters <walters@verbum.org>

Patch from Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c (invoke_object_method): Unset object
value in all cases, not only in async case.

```
glib/dbus-gobject.c |    4 ++--  
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit 49fce903db4124277de6aba92f7c03f24d84e598
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-29 16:58:59 +0000

2005-06-29 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (struct _DBusGProxy): Add new member
name_call for keeping track of any outgoing GetNameOwner call.
Also add for_owner and associated.

(struct _DBusGProxyManager): Add owner_names, which is hash table that maps a base name to a list of names it owns (that we're interested in). Add pending_nameowner_calls which is a list of all outstanding GetNameOwner; avoids us having to iterate over every proxy. Add unassociated_proxies which keeps track of name proxies with no associated name owner.
(dbus_g_proxy_manager_unref): Destroy owner_names.
(struct DBusGProxyNameOwnerInfo): New struct for keeping track of name refcounts.
(find_name_in_info, name_owner_foreach)
(dbus_g_proxy_manager_lookup_name_owner, insert_nameinfo)
(dbus_g_proxy_manager_monitor_name_owner)
(dbus_g_proxy_manager_unmonitor_name_owner)
(unassociate_proxies, dbus_g_proxy_manager_replace_name_owner): New functions; they manipulate the owner_names mapping.
(got_name_owner_cb): New function.
(get_name_owner): New function, extracted from dbus_g_proxy_new_for_name_owner.
(dbus_g_proxy_manager_register): For now we need to keep track of all NameOwnerChanged. Also if the proxy is for a name, if we don't already know the name owner, queue a new GetNameOwner request and add it to our list of unassociated proxies.

Otherwise

inc the refcount.

(dbus_g_proxy_manager_unregister): If this proxy is for a name, cancel any pending GetNameOwner call, etc.

(dbus_g_proxy_manager_filter): Handle NameOwnerChanged. Also use the owner_names mapping to look up the current names for the signal source, and dispatch to any proxies for that name.

(dbus_g_proxy_new): Initialize new members.

(dbus_g_proxy_new_for_name): Delete unused proxy variable.

(dbus_g_proxy_new_for_name_owner): Use get_name_owner.

(dbus_g_pending_call_end_valist): New function, extracted from dbus_g_proxy_end_call_internal. Useful when we don't have a

proxy

but want to use the GLib infrastructure. Also note how many arguments in reply were over.

(dbus_g_pending_call_end): New function, just call dbus_g_pending_call_end_valist.

(dbus_g_proxy_end_call_internal): Just call dbus_g_pending_call_end_valist.

* glib/dbus-gobject.c (_dbus_gobject_lookup_marshall):

Fix lookup

of builtin marshaller for STRING_STRING_STRING.

* test/glib/test-dbus-glib.c:

* test/glib/test-service-glib.c:

* test/glib/test-service-glib.xml:

Extend tests to cover name proxies, destruction of owner proxies, etc.

```
* glib/examples/example-signal-recipient.c
(dbus_g_proxy_new_for_name_owner): Create a name proxy.

* tools/dbus-send.c (main): Print D-BUS error name in addition
to message.
```

```
glib/dbus-gobject.c | 14 +-
glib/dbus-gproxy.c | 780
+++++-----
glib/examples/example-signal-recipient.c | 9 +-
3 files changed, 689 insertions(+), 114 deletions(-)
```

```
commit f2f505e8027bd4ef84b051ad9cd84ab86524b530
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-27 18:20:20 +0000
```

```
2005-06-27 Colin Walters <walters@verbum.org>
```

```
* test/glib/test-dbus-glib.c:
* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml:
Test hash table signal emitting.
```

```
* glib/dbus-gobject.c (_dbus_gobject_lookup_marshallers): Convert
types to their fundamental basis types, since this is what
marshallers operate on. Also add an entry for VOID__BOXED.
(dbus_g_object_register_marshallers_array): Convert to
fundamental.
```

```
glib/dbus-gobject.c | 28 +++++-----
1 file changed, 21 insertions(+), 7 deletions(-)
```

```
commit f0480850590afa233d7e1ad2b0c8250621c2bb9f
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-26 17:02:09 +0000
```

```
2005-06-26 Colin Walters <walters@verbum.org>
```

```
* glib/dbus-glib.c (dbus_set_g_error): Delete.
(dbus_g_error_set): New public function from its ashes; used by
both service-side method implementation and GLib bindings
internals.
(dbus_g_error_has_name, dbus_g_error_get_name): New function.
(_dbus_glib_test): Add some tests.
```

```
* test/glib/test-dbus-glib.c (main): Test dbus_g_error_has_name.
```

```
* test/glib/test-service-glib.c (my_object_throw_error): Use
dbus_g_error_set.
```

```
* glib/dbus-gobject.c (gerror_to_dbus_error_message): Handle
errors thrown by dbus_g_error_set.
```

- * glib/dbus-gmain.c (dbus_g_bus_get): Change to dbus_g_error_set.
- * glib/dbus-gparser.c (validate_signature): Ditto.
- * glib/dbus-gproxy.c (dbus_g_proxy_new_for_name_owner) (dbus_g_proxy_end_call_internal): Ditto.
- * glib/Makefile.am: Generate dbus-glib-error-switch.h, which converts DBUS_ERROR_x to DBUS_GERROR_x. (libdbus_glib_1_la_SOURCES, BUILT_SOURCES, CLEANFILES): Add it.
- * doc/TODO: Remove error TODO.
- * doc/dbus-tutorial.xml: Update with documentation about error handling.
- * dbus/make-dbus-glib-error-enum.sh: Tighten up regexp to make sure we only change DBUS_ERROR to DBUS_GERROR, not all ERROR to GERROR. Also add DBUS_GERROR_REMOTE_EXCEPTION.

```

glib/.cvsignore          |          1 +
glib/Makefile.am         |           8 +++
glib/dbus-glib.c         |          109
+++++-----
glib/dbus-gmain.c       |           2 +-
glib/dbus-gobject.c     |          15 +++--
glib/dbus-gparser.c     |           2 +-
glib/dbus-gproxy.c      |           6 +-
glib/make-dbus-glib-error-switch.sh | 29 ++++++++
8 files changed, 150 insertions(+), 22 deletions(-)

```

```

commit ce03d4c9e9be4724a3255085b5ab5e51d9b9c196
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-22 23:35:32 +0000

```

2005-06-22 Colin Walters <walters@verbum.org>

Patch from Ross Burton <ross@openedhand.com>

- * glib/dbus-gobject.c (dbus_g_method_return): Free out_sig.

```

glib/dbus-gobject.c |          1 +
1 file changed, 1 insertion(+)

```

```

commit 6d152f08fa90bdc14d75b9091cccd5422faf3f11
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-21 01:18:25 +0000

```

2005-06-20 Colin Walters <walters@verbum.org>

- * configure.in: Add glib/examples.

```

* glib/Makefile.am: Add examples/

* glib/examples/.cvsignore
* glib/examples/Makefile.am
* glib/examples/example-client.c
* glib/examples/example-service.c
* glib/examples/example-service.xml
* glib/examples/example-signal-emitter.c
* glib/examples/example-signal-emitter.xml
* glib/examples/example-signal-recipient.c:
New files; GLib binding examples, ported from
python/examples.

```

```

glib/Makefile.am          |    2 +
glib/examples/.cvsignore  |   17 ++++
glib/examples/Makefile.am |   34 ++++++++
glib/examples/example-client.c |  107 ++++++++
glib/examples/example-service.c |  138
+++++++
glib/examples/example-service.xml |   19 ++++
glib/examples/example-signal-emitter.c |  132
+++++++
glib/examples/example-signal-emitter.xml |   13 +++
glib/examples/example-signal-recipient.c |  103
+++++++
9 files changed, 565 insertions(+)

```

```

commit ad05d7aa3b298fdc6c214550944e9aa1464a1e43
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-21 00:30:20 +0000

```

2005-06-20 Colin Walters <walters@verbum.org>

```

* dbus/dbus-glib.h:
* glib/dbus-gproxy.c: Rename dbus_g_proxy_invoke to
dbus_g_proxy_call.

* glib/dbus-binding-tool-glib.c:
* doc/dbus-tutorial.xml:
* test/glib/test-dbus-glib.c: Update for rename.

```

```

glib/dbus-binding-tool-glib.c |    2 +-
glib/dbus-gproxy.c           |   18 ++++++-----
2 files changed, 11 insertions(+), 9 deletions(-)

```

```

commit 17ba4df65fdf4a7b65eedd2a4f10e616dd1b186c
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-20 19:54:21 +0000

```

2005-06-20 Colin Walters <walters@verbum.org>

Patch suggested by Ross Burton <ross@openedhand.com>

- * glib/dbus-gobject.c (export_signals): Free signal name.
- (g_value_init): Use G_VALUE_NOCOPY_CONTENTS to plug memory leak. Add a bit of documentation.
- (dbus_g_method_return_error): Free context, and note we do so.

glib/dbus-gobject.c | 6 ++++-
1 file changed, 5 insertions(+), 1 deletion(-)

commit 01b89e61b5b98ad983babb506c9856aeade2562
Author: Murray Cumming <murrayc@murrayc.com>
Date: 2005-06-19 15:31:25 +0000

2005-06-18 Murray Cumming <murrayc@murrayc.com>

- * dbus/dbus-glib.h:
- * glib/dbus-gobject.c:
- * glib/dbus-gproxy.c:
- * glib/dbus-gvalue.c: Predeclare structs as typedef struct _Something Something instead of typedef struct Something Something, so we can redeclare the prototypes. Other GNOME libraries do this already.

glib/dbus-gobject.c | 2 +-
glib/dbus-gproxy.c | 8 ++++----
glib/dbus-gvalue.c | 2 +-
3 files changed, 6 insertions(+), 6 deletions(-)

commit da18702f63a9f33d4f929113b6361062a86443a6
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-17 14:29:48 +0000

2005-06-17 Colin Walters <walters@verbum.org>

- * glib/dbus-gproxy.c (dbus_g_proxy_emit_remote_signal): Don't spew warnings if we get malformed remote signals.

- * glib/dbus-gobject.c (propsig_iterate): New function.
- (lookup_object_info): New function, extracted from lookup_object_and_method.
- (introspect_properties, introspect_signals): Delete; these are merged into write_interface.
- (write_interface): Write out signals and properties here; dump the org.gtk.object stuff and use the interface given in the introspection data blob. Also fix up property XML.
- (lookup_values): New function.
- (introspect_interfaces): Gather a mapping from interface to a list of its methods, signals, and properties, then write out each interface.

(lookup_object_and_method): Use lookup_object_info.
 (struct DBusGSignalClosure): Add interface.
 (dbus_g_signal_closure_new): Add interface. Don't dup signame;
 we can just use the constant data.
 (dbus_g_signal_closure_finalize): Don't free signal name.
 (signal_emitter_marshall): Use interface from signal closure.
 (export_signals): Only export signals mentioned in introspection
 blob.
 (dbus_g_connection_register_g_object): Warn if we have no
 introspection data for an object.
 (funcsig_equal): Remove unused variable.
 (dbus_g_object_register_marshall): Take varargs instead of
 list.
 (dbus_g_object_register_marshall_array): New function,
 extracted from old dbus_g_object_register_marshall.

* glib/dbus-binding-tool-glib.c (struct DBusBindingToolCData):
 Add
 signals and property data.
 (write_quoted_string): New function, extracted from
 generate_glue.
 (generate_glue): Write signals and properties to introspection
 blob.

* dbus/dbus-glib.h (struct DBusGObjectInfo): Include
 exported_signals and exported_properties.
 (dbus_g_object_register_marshall): Update prototype.
 (dbus_g_object_register_marshall_array): Prototype.

* test/glib/test-dbus-glib.c: Extend testing to cover new
 signals.

* test/glib/test-service-glib.c: Add new test signals and method
 to emit them.

* test/glib/test-service-glib.xml: Add some test signals.

* test/glib/Makefile.am (BUILT_SOURCES): Add my-object-marshal.c
 and my-object-marshal.h
 (test_service_glib_SOURCES, test_dbus_glib_SOURCES): Add
 my-object-marshal.c.
 (my-object-marshal.c, my-object-marshal.h): Implement.

* test/glib/.cvsignore: Update.

* doc/TODO: Remove two GLib TODO items fixed by this
 patch.

```

glib/dbus-binding-tool-glib.c | 88 +++++--
glib/dbus-gobject.c          | 547
+++++-----
glib/dbus-gproxy.c          | 3 +
  
```

3 files changed, 394 insertions(+), 244 deletions(-)

commit 5c64ac1e7abcb7013c640d8aa6621319341f228d
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-16 19:50:24 +0000

2005-06-16 Colin Walters <walters@verbum.org>

- * glib/dbus-binding-tool-glib.c:
- * glib/dbus-gobject.c:
- * glib/dbus-gproxy.c: Add Nokia copyright; Patch from Ross Burton, for his GLib bindings work.

```
glib/dbus-binding-tool-glib.c | 1 +
glib/dbus-gobject.c          | 1 +
glib/dbus-gproxy.c          | 1 +
3 files changed, 3 insertions(+)
```

commit 87556105e7b984f8e56fc75d5c5ede8a69d63d12
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-16 19:45:49 +0000

2005-06-16 Colin Walters <walters@verbum.org>

- * glib/dbus-gobject.c (funcsig_hash, funcsig_equal): Use n_params to iterate instead of walking to G_TYPE_INVALID.

Patch based on a patch from Ryan Gammon.

```
glib/dbus-gobject.c | 11 ++++++-----
1 file changed, 6 insertions(+), 5 deletions(-)
```

commit b274594d1f15e23be62d3ae2693524652ff58752
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-16 04:38:11 +0000

2005-06-16 Colin Walters <walters@verbum.org>

- * glib/dbus-gvalue.c (marshal_basic): Marshal NULL string values as the empty string (#2948).

```
glib/dbus-gvalue.c | 2 ++
1 file changed, 2 insertions(+)
```

commit 702f09f4f45225f603a6d17ae77c3360c428855a
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-14 15:55:10 +0000

2005-06-14 Colin Walters <walters@verbum.org>

- * glib/dbus-binding-tool-glib.c:
- * glib/dbus-gobject.c:

* glib/dbus-gvalue.c: Fix indentation and brace style.

```
glib/dbus-binding-tool-glib.c | 74 ++++++-----
glib/dbus-gobject.c           | 190
+++++++-----
glib/dbus-gvalue.c           | 11 +--
3 files changed, 145 insertions(+), 130 deletions(-)
```

commit ceb44c57e8ebc78ef420597bae484f7713c2e8bf
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-14 15:49:43 +0000

2005-06-14 Ross Burton <ross@burtonini.com>.

* glib/dbus-glib.h: Make DBusGMethodInvocation
a private structure. Rearrange prototypes a bit.

* glib/dbus-gproxy.c (dbus_g_proxy_invoke): Add
documentation for first_arg_type.

* glib/dbus-gobject.c: Move DBusGMethodInvocation
here, add documentation. Move dbus_g_method_return
and dbus_g_method_return_error into public API
section.

```
glib/dbus-gobject.c | 119
+++++++-----
glib/dbus-gproxy.c | 1 +
2 files changed, 71 insertions(+), 49 deletions(-)
```

commit 72b52186a73aac9f4d0ael868fel dedcedf3a6e6
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-14 14:23:56 +0000

2005-06-14 Colin Walters <walters@verbum.org>

* glib/dbus-gobject.c (_dbus_gobject_lookup_marshaller):
Add missing return statements, noticed by Ross Burton.

```
glib/dbus-gobject.c | 6 +++++
1 file changed, 6 insertions(+)
```

commit 40617899dc08750ab5a8cd0fb176dc2111a95420
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-13 15:36:31 +0000

2005-06-13 Ross Burton <ross@burtonini.com>.

* glib/dbus-gobject.c: Handle errors on message
demarshalling by sending error message back.

* glib/dbus-gvalue.c: Initialize return variables.


```
glib/dbus-gobject.c |      11 ++++++---
glib/dbus-gvalue.c  |       8 ++++----
2 files changed, 12 insertions(+), 7 deletions(-)
```

```
commit fc23d39a292cee7d4f9ee9a1df0c0141c69d0942
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-13 15:34:05 +0000
```

2005-06-13 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Fix thinko in last patch.

```
glib/Makefile.am |     2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit 7c42065c67e99716130a5dec4e493bce3ccc9aaa
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-13 14:56:51 +0000
```

2005-06-13 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Move dbus-gtype-specialized.c and dbus-gtype-specialized.h into a `_HEADERS` variable, install them.

```
glib/Makefile.am |     8 ++++++--
1 file changed, 6 insertions(+), 2 deletions(-)
```

```
commit 8bcbafe775d6bc63290c5c73f138c1511b104c70
Author: Colin Walters <walters@verbum.org>
Date: 2005-06-13 03:01:17 +0000
```

2005-06-12 Colin Walters <walters@verbum.org>

Async signals and various bugfixes and testing by Ross Burton <ross@burtonini.com>.

* glib/dbus-gvalue.h: (struct DBusBasicGValue): Delete.
(dbus_gvalue_genmarshal_name_from_type)
(dbus_gvalue_ctype_from_type): Moved to dbus-binding-tool-glib.c.
(dbus_gtype_to_dbus_type): Renamed to dbus_gtype_from_signature.
(dbus_g_value_types_init, dbus_gtype_from_signature)
(dbus_gtype_from_signature_iter, dbus_gtype_to_signature)
(dbus_gtypes_from_arg_signature): New function prototypes.
(dbus_gvalue_demarshal): Take context and error arguments.
(dbus_gvalue_demarshal_variant): New function.
(dbus_gvalue_demarshal_message): New function.
(dbus_gvalue_store): Delete.

* glib/dbus-gvalue.c:

File has been almost entirely rewritten; now we special-case

more types such as `DBUS_TYPE_SIGNATURE`, handle arrays and hash tables correctly, etc. Full support for recursive values is not yet complete.

- * `glib/dbus-gproxy.c` (`dbus_g_proxy_class_init`): Change last argument of `signal` to `G_TYPE_POINTER` since we now pass a structure.
- (`lookup_g_marshaller`): Delete in favor of `_dbus_gobject_lookup_marshaller`.
- (`marshal_dbus_message_to_g_marshaller`): Use `_dbus_gobject_lookup_marshaller` and `dbus_gvalue_demarshal_message` to handle remote signal callbacks.
- (`dbus_g_proxy_new_from_proxy`): New function; creates a new `DBusGProxy` by copying an existing one.
- (`dbus_g_proxy_get_interface`, `dbus_g_proxy_set_interface`)
- (`dbus_g_proxy_get_path`): New functions.
- (`dbus_g_proxy_marshal_args_to_message`): New function; factored out of existing code.
- (`DBUS_G_VALUE_ARRAY_COLLECT_ALL`): Collect all arguments from a `varargs` array.
- (`dbus_g_proxy_begin_call_internal`): New function.
- (`dbus_g_proxy_end_call_internal`): New function.
- (`dbus_g_proxy_begin_call`): Take `GTypes` instead of `DBus` types as arguments; simply invoke `dbus_g_proxy_begin_call_internal` after collecting args into value array.
- (`dbus_g_proxy_end_call`): Take `GTypes` instead of `DBus` types; invoke `dbus_g_proxy_end_call_internal`.
- (`dbus_g_proxy_invoke`): Simply invoke `begin_call_interanl` and `end_call_internal`.
- (`dbus_g_proxy_call_no_reply`): Take `GTypes` instead of `DBus` types.
- (`array_free_all`): New function.
- (`dbus_g_proxy_add_signal`): Take `GTypes`.

- * `glib/dbus-gobject.h`:
 - (`_dbus_glib_marshal_dbus_message_to_gvalue_array`): Delete.
 - (`_dbus_gobject_get_path`, `_dbus_gobject_lookup_marshaller`): Prototype.

- * `glib/dbus-gobject.c`: Add a global `marshal_table` hash which stores mappings from type signatures to `marshallers`. Change lots of invocations of `dbus_gtype_to_dbus_type` to `dbus_gtype_to_signature`.
 - (`_dbus_glib_marshal_dbus_message_to_gvalue_array`): Delete.
 - (`introspect_signals`): Fix test for `query.return_type`.
 - (`set_object_property`): Update invocation of `dbus_gvalue_demarshal`.
 - (`invoke_object_method`): Many changes. Handle asynchronous invocations. Convert arguments with `dbus_gvalue_demarshal_message`. Handle errors. Use `DBusSignatureIter` instead of `strlen` on args. Handle all arguments generically. Special-case variants.

(dbus_g_method_return, dbus_g_method_return_error): New function.
 (DBusGSignalClosure): New structure, closes over signal information.
 (dbus_g_signal_closure_new): New function.
 (dbus_g_signal_closure_finalize): New function.
 (signal_emitter_marshall): New function; is special marshaller which emits signals on bus.
 (export_signals): New function; introspects object signals and connects to them.
 (dbus_g_object_type_install_info): Take GType instead of GObjectClass.
 (dbus_g_connection_register_g_object): Invoke export_signals.
 (dbus_g_connection_lookup_g_object): New function.
 (DBusGFuncSignature) New structure; used for mapping type signatures tomarshallers.
 (funcsig_hash): New function; hashes DBusGFuncSignature.
 (funcsig_equal): New function; compares DBusGFuncSignature.
 (_dbus_gobject_lookup_marshall): New function.
 (dbus_g_object_register_marshall): New function; used to register a marshaller at runtime for a particular signature.

* glib/dbus-gmain.c (_dbus_gmain_test): Add various tests.

* glib/dbus-binding-tool-glib.h: Add DBUS_GLIB_ANNOTATION_ASYNC which notes a server method implementation should be asynchronous.

* glib/dbus-binding-tool-glib.c
 (dbus_binding_tool_output_glib_server): Call dbus_g_value_types_init.
 (write_formal_parameters): Use dbus_gtype_from_signature. Handle variants specially.
 (dbus_g_type_get_lookup_function): Turn GType into an invocation of a lookup function.
 (write_args_for_direction): Use dbus_g_type_get_lookup_function.
 (write_untyped_out_args): New method; write output arguments.
 (write_formal_declarations_for_direction): Function for writing prototypes.
 (write_formal_parameters_for_direction): Function for writing implementations.
 (write_typed_args_for_direction): Function for writing arguments prefixed with GTypes.
 (write_async_method_client): Write out async version of method.

* glib/dbus-binding-tool-glib.c: Include dbus-gvalue-utils.h.
 (dbus_g_type_get_marshal_name): Move mapping from GType to marshal name into here.
 (dbus_g_type_get_c_name): Move into here.
 (compute_marshall): Convert signature to type with dbus_gtype_from_signature, use dbus_g_type_get_marshal_name.
 (compute_marshall_name): Ditto.

(compute_marshall): Handle async signal annotations.
(gather_marshallers): Return if we don't have a known prefix.
(generate_glue): Collect introspection blob here, and write all of the blob at the end. This allows an object with multiple interfaces to work.
Mark async methods in introspection blob.

* glib/Makefile.am (libdbus_glib_1_la_SOURCES): Add dbus-gtype-specialized.c, dbus-gtype-specialized.h, dbus-gvalue-utils.h, dbus-gvalue-utils.c.

* dbus/dbus-glib.h: Don't include dbus-protocol.h; this avoids people accidentally using DBUS_TYPE_* which should not be necessary anymore.
Do include dbus-gtype-specialized.h, which are utilities for GLib container types.
Add various #defines for types such as DBUS_TYPE_G_BOOLEAN_ARRAY.
(DBusGValueIterator, DBusGValue): Define, not fully used yet.
(dbus_g_value_get_g_type): Type for recursive value.
(dbus_g_value_open, dbus_g_value_iterator_get_value)
(dbus_g_value_iterator_get_values, dbus_g_value_iterator_recurse)
(dbus_g_value_free): Prototypes.
(dbus_g_object_register_marshall, dbus_g_proxy_new_from_proxy): Prototype.
(dbus_g_proxy_set_interface): Prototype.
(dbus_g_proxy_begin_call, dbus_g_proxy_end_call)
(dbus_g_proxy_call_no_reply): Take GLib types instead of Dbus types.
(dbus_g_proxy_get_path, dbus_g_proxy_get_interface): Accessors.
(DBusGAsyncData, DBusGMethodInvocation): Structures for doing async invocations.
(dbus_g_method_return, dbus_g_method_return_error): Prototypes.

* doc/dbus-tutorial.xml: Update GLib section.

* tools/dbus-viewer.c (load_child_nodes): Update for new invocation type of dbus_g_proxy_end_call.
(load_from_service_thread_func): Ditto.

* tools/print-introspect.c (main): Ditto.

* tools/dbus-names-model.c (have_names_notify)
(names_model_reload, names_model_set_connection)
Use GTypes.

* python/Makefile.am (INCLUDES): Define DBUS_COMPILATION, needed since Python bindings use GLib bindings.

* test/glib/Makefile.am (INCLUDES): Define DBUS_COMPILATION.
Add --prefix argument.

* tools/Makefile.am: Define DBUS_COMPILATION. Remove
unneded --ignore-unsupported arg.

* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml:
* test/glib/test-dbus-glib.c: Add many more tests.

```
glib/Makefile.am          |    6 +-
glib/dbus-binding-tool-glib.c | 642 ++++++++-----
glib/dbus-binding-tool-glib.h |    1 +
glib/dbus-gmain.c         |   29 +
glib/dbus-gobject.c       |  599 ++++++++-----
glib/dbus-gobject.h       |   10 +-
glib/dbus-gproxy.c        |  789 ++++++++-----
glib/dbus-gtype-specialized.c |  441 ++++++++
glib/dbus-gtype-specialized.h |   104 +++
glib/dbus-gvalue-utils.c   |   715 ++++++++
glib/dbus-gvalue-utils.h   |    70 ++
glib/dbus-gvalue.c         |  1508
+++++++
glib/dbus-gvalue.h         |    55 +-
13 files changed, 4185 insertions(+), 784 deletions(-)
```

commit 4bc350fedd536bc187768a7bfa2455aae7474f74
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-05-16 21:27:03 +0000

* glib/dbus-gmain.c (io_handler_dispatch): fix deadlock when using
recursive g_main_loops

* python/_dbus.py (class Bus): add the ProxyObjectClass
alias for ProxyObject to make it easier for the Twisted
networking framework to integrate dbus.

* python/proxies.py (class ProxyObject): add the ProxyMethodClass
alias for ProxyMethod to make it easier for the Twisted
networking framework to integrate dbus.

```
glib/dbus-gmain.c |    8 +-----
1 file changed, 1 insertion(+), 7 deletions(-)
```

commit 3c7dcbf8f47d3061fccdc315716c90a0b69a470f
Author: Colin Walters <walters@verbum.org>
Date: 2005-05-11 19:02:31 +0000

2005-05-11 Ross Burton <ross@burtonini.com>

* glib/dbus-glib-tool.c: Add --prefix argument.
* glib/dbus-binding-tool-glib.h: Add prefix argument.

```
* glib/dbus-binding-tool-glib.c (compute_marshall_name):  
Add prefix argument.  
(generate_glue): Pass prefix argument down.  
(dbus_binding_tool_output_glib_server): Pass prefix to  
glib-genmarshal.
```

```
glib/dbus-binding-tool-glib.c | 16 ++++++++-----  
glib/dbus-binding-tool-glib.h | 2 +-  
glib/dbus-glib-tool.c | 8 ++++++-  
3 files changed, 18 insertions(+), 8 deletions(-)
```

```
commit 55263b490f4dba73b1590939be2b087856487054  
Author: Colin Walters <walters@verbum.org>  
Date: 2005-05-03 17:45:29 +0000
```

```
2005-05-03 Ross Burton <ross@burtonini.com>
```

```
* glib/dbus-gobject.c (dbus_g_connection_register_g_object):  
Return if we get an error during registration. Set up a  
weak reference on object to unregister if object is destroyed.  
(unregister_gobject): New function.
```

```
glib/dbus-gobject.c | 18 ++++++++-----  
1 file changed, 14 insertions(+), 4 deletions(-)
```

```
commit 260elfecf1690bb12c5bf7d5ddf793d79a1efed5  
Author: David Zeuthen <david@fubar.dk>  
Date: 2005-04-19 03:35:19 +0000
```

```
2005-04-18 David Zeuthen <davidz@redhat.com>
```

```
* glib/dbus-gmain.c (io_handler_destroy_source):  
(timeout_handler_destroy_source, connection_setup_free):  
Also unref the source to avoid memory leaks.
```

```
glib/dbus-gmain.c | 3 +++  
1 file changed, 3 insertions(+)
```

```
commit 56288b6e3a9e5f8bd8920c5f7dac2a2b64842b15  
Author: Havoc Pennington <hp@redhat.com>  
Date: 2005-04-13 14:10:21 +0000
```

```
2005-04-13 Havoc Pennington <hp@redhat.com>
```

```
* glib/dbus-gmain.c (message_queue_dispatch): only dispatch one  
message at a time to avoid monopolizing the main loop, bug  
#2953 from Benjamin Otte
```

```
glib/dbus-gmain.c | 6 +++---  
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
commit ca2ab9616ed7c6e936e495e64303f77956cb6b82
```

Author: Havoc Pennington <hp@redhat.com>
Date: 2005-04-09 23:50:58 +0000

2005-04-09 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message-util.c (_dbus_message_test): fix signedness warning
- * glib/dbus-glib-tool.c (main): fix warning
- * glib/dbus-binding-tool-glib.c (generate_glue): fix warning
- * dbus/dbus-connection.c (dbus_connection_read_write_dispatch): add a new function that can be used in simple applications that don't have a main loop and are willing to block

```
glib/dbus-binding-tool-glib.c |    3 +++  
glib/dbus-glib-tool.c         |    1 +  
2 files changed, 4 insertions(+)
```

commit b34961c0e5a75ff9f70e0ba439a96d993456ef92
Author: David Zeuthen <david@fubar.dk>
Date: 2005-04-06 17:36:47 +0000

2005-04-05 David Zeuthen <davidz@redhat.com>

Fix https://bugs.freedesktop.org/show_bug.cgi?id=2889

- * glib/dbus-gmain.c:
(io_handler_destroy_source): Remove from list of IO handlers of the ConnectionSetup object
(timeout_handler_destroy_source): -do- for timeout handlers
(io_handler_source_finalized): Don't remove from list since we now do that in io_handler_destroy_source(). Renamed from io_handler_source_destroyed
(timeout_handler_source_destroyed): -do- for timeout handlers
(connection_setup_free): It is now safe to iterate over all IO and timeout handlers as the _destroy_source removes them from the list synchronously

```
glib/dbus-gmain.c |   14 ++++++-----  
1 file changed, 6 insertions(+), 8 deletions(-)
```

commit 681e7deee9a01c1f315b0066b87204e0da207d99
Author: Colin Walters <walters@verbum.org>
Date: 2005-03-17 17:48:29 +0000

2005-03-17 Colin Walters <walters@verbum.org>

- * bus/print-introspect.c: Move to tools/.
- * bus/run-with-tmp-session-bus.sh: Ditto.

- * glib/Makefile.am (dbus-glib-bindings.h): Move generation to tools/Makefile.am.
- * test/glib/run-test.sh: Update to handle move of run-with-tmp-session-bus.sh.
- * test/glib/test-service-glib.c: Update to handle move of dbus-glib-bindings.h.
- * tools/print-introspect.c: Moved here from bus/, and ported to GLib bindings.
- * tools/run-with-tmp-session-bus.sh: Moved here from bus/.
- * tools/Makefile.am: Generate dbus-glib-bindings.h and dbus-bus-introspect.xml here.
- * tools/.cvsignore, glib/.cvsignore, bus/.cvsignore: Update.

```
glib/.cvsignore | 1 -  
glib/Makefile.am | 8 -----  
2 files changed, 9 deletions(-)
```

```
commit 1cc2659deb9f108f449ad3ca8f18bff743a4cd81  
Author: Colin Walters <walters@verbum.org>  
Date: 2005-03-12 20:07:21 +0000
```

```
2005-03-12 Colin Walters <walters@verbum.org>
```

- * bus/driver.c (write_args_for_direction): New function, parses a type signature into arguments and outputs to XML.
(bus_driver_handle_introspect): Use it instead of hardcoding XML for certain signatures.
- * bus/Makefile.am (dbus-bus-introspect.xml): Add dependency on dbus-daemon.
- * glib/dbus-glib-tool.c (main): Parse ignore_unsupported argument, pass it to dbus_binding_tool_output_glib_client.
- * glib/dbus-binding-tool-glib.c
(generate_client_glue): Protect against multiple inclusion.
(dbus_binding_tool_output_glib_client): Add G_BEGIN_DECLS/G_END_DECLS.
- * glib/dbus-binding-tool-glib.c (compute_client_method_name): Change to just take iface prefix directly.
(write_formal_parameters): Clarify error message.
(check_supported_parameters): New function; checks to see type

signatures of method parameters are supported.
(generate_client_glue): Handle ignore_unsupported flag.
(dbus_binding_tool_output_glib_client): Handle ignore_unsupported parameter.

* glib/Makefile.am (dbus-glib-bindings.h): Pass --ignore-unsupported by default until glib bindings support arrays.

```
glib/Makefile.am          |    2 +-  
glib/dbus-binding-tool-glib.c |   64  
+++++-----  
glib/dbus-binding-tool-glib.h |    2 +-  
glib/dbus-glib-tool.c      |    6 +++-  
4 files changed, 63 insertions(+), 11 deletions(-)
```

commit c5247484ea45ef9b6b3449a71e3ea80b86de03ad
Author: Colin Walters <walters@verbum.org>
Date: 2005-03-12 16:33:00 +0000

2005-03-11 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Generate dbus-glib-bindings.h and install it.

* bus/print-introspect.c: New file; prints introspection data for a given name and object path.

* bus/run-with-tmp-session-bus.sh: New file, refactored from test/glib/run-test.sh. Creates a temporary session bus and runs another program.

* test/glib/run-test.sh: Refactor to invoke run-with-tmp-session-bus.sh.

* bus/driver.c (bus_driver_handle_introspect): Fix to print new introspection format. Also change to use DBUS_TYPE_x_AS_STRING macros instead of hardcoding.

* glib/.cvsignore, bus/.cvsignore, test/glib/.cvsignore: Update.

```
glib/.cvsignore |    1 +  
glib/Makefile.am |    8 +++++++  
2 files changed, 9 insertions(+)
```

commit 10dc75239d071c49622972d061ff7546ed169bc5
Author: Colin Walters <walters@verbum.org>
Date: 2005-03-09 17:09:11 +0000

2005-03-09 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (dbus_g_proxy_invoke): New method; calls

to this are generated for client-side wrappers. Invokes a D-BUS method and returns reply values.

* glib/dbus-binding-tool-glib.c (write_args_sig_for_direction):
New
function; writes signature string for argument direction.
(write_args_for_direction): Change to pass input values directly
instead of via address, and fix indentation.
(generate_client_glue): Change to invoke dbus_g_proxy_invoke.
Also
make generated wrappers inlineable.

* dbus/dbus-message.c (dbus_message_iter_get_fixed_array): Add
note about using dbus_type_is_fixed.

* dbus/dbus-marshall-basic.c (_dbus_type_is_fixed): Moved to
dbus/dbus-signature.c as dbus_type_is_fixed.

All callers updated.

* dbus/dbus-signature.c (dbus_type_is_fixed): Moved here
from dbus/dbus-marshall-basic.c:_dbus_type_is_fixed.

* dbus/dbus-signature.h: Prototype.

* glib/dbus-binding-tool-glib.c (compute_marshall_name): Fix
error printf code.

* test/glib/test-dbus-glib.c (main): Be sure to clear error as
appropriate instead of just freeing it.
(main): Free returned strings using g_free.

* test/glib/Makefile.am (test-service-glib-glue.h)
(test-service-glib-bindings.h): Add dependency on
dbus-binding-tool.

* glib/dbus-gvalue.c (MAP_BASIC): Refactored from MAP_BASIC_INIT;
simply maps a simple D-BUS type to GType.
(dbus_dbus_type_to_gtype): Function which maps D-BUS type to
GType.
(dbus_gvalue_init): Just invoke dbus_dbus_type_to_gtype and
initialize the value with it.
(dbus_gvalue_binding_type_from_type): Unused, delete.
(dbus_gvalue_demarshal): Switch to hardcoding demarshalling for
various types instead of unmarshalling to value data directly.
Remove can_convert boolean.
(dbus_gvalue_marshal): Remove duplicate initialization; switch to
returning directly instead of using can_convert boolean.
(dbus_gvalue_store): New function; not related to D-BUS per-se.
Stores a GValue in a pointer to a value of its corresponding C
type.

* glib/dbus-gvalue.h: Remove dbus_gvalue_binding_type_from_type,
add dbus_gvalue_store.

```
glib/dbus-binding-tool-glib.c | 73 ++++++-----  
glib/dbus-gproxy.c           | 205 ++++++  
glib/dbus-gvalue.c           | 254  
+++++  
glib/dbus-gvalue.h           | 4 +-  
4 files changed, 392 insertions(+), 144 deletions(-)
```

commit 61b05621beeb5a929bf68b95fdb9cb63516c1c7e
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-03-07 21:10:46 +0000

- * NEWS: Update for 0.31
- * configure.in: Release 0.31
add LT_CURRENT, LT_REVISION, LT_AGE for easy soname bumping
- * qt/Makefile.am: fixed build
- * dbus/Makefile.am: soname bump for libdbus
- * glib/Makefile.am: soname bump for libdbus-glib

```
glib/Makefile.am | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

commit be6ef9488ddc1b9fd3fc303cccf6e6618450be89
Author: Colin Walters <walters@verbum.org>
Date: 2005-02-27 17:38:12 +0000

2005-02-27 Colin Walters <walters@verbum.org>

* glib/dbus-gidl.c (property_info_get_type, arg_info_get_type):
Change return value to const char * instead of int so we can do
full signatures.

(struct PropertyInfo, struct ArgInfo): Store char *.
(property_info_new, arg_info_new): Update parameters, strdup.
(property_info_unref, arg_info_unref): Free.

* glib/dbus-gidl.h: Update prototypes.

* glib/dbus-gparser.c (basic_type_from_string): Delete.
(validate_signature): New function, just validates signature and
sets GError.

(parse_property, parse_arg): Invoke validate_signature. Store
signature instead of just type code.

* glib/dbus-gvalue.c (base_type_from_signature): New utility
function to return a primary type for a signature, dropping
information about types in container types.

```

(dbus_gvalue_genmarshal_name_from_type)
(dbus_gvalue_binding_type_from_type)
(dbus_gvalue_ctype_from_type): Update to take full signature
instead of type code.
(dbus_gtype_to_dbus_type): Moved here from glib/dbus-gobject.c.

* glib/dbus-gvalue.h: Update prototypes for above.

* glib/dbus-gobject.c (gtype_to_dbus_type): Moved to
glib/dbus-gvalue.c as dbus_gtype_to_dbus_type.
(introspect_properties, introspect_signals, write_interface):
Update to handle signatures, and remove usage of
_dbus_gutils_type_to_string.
(handle_introspect): Print out type codes instead of e.g.
"string"
in hardcoded introspection XML; also use x_AS_STRING constants
instead of hardcoding in string.

* glib/dbus-glib-tool.c (pretty_print): Handle signature change
to string. Remove usage of _dbus_gutils_type_to_string.

* glib/dbus-gutils.c (_dbus_gutils_type_to_string): Delete.

* glib/dbus-gutils.h (_dbus_gutils_type_to_string): Update for
deletion.

* glib/dbus-binding-tool-glib.c (compute_marshall)
(compute_marshall_name, generate_glue): Handle signature change
to string.
(write_formal_parameters, write_args_for_direction): Ditto, and
remove FIXME.

* tools/dbus-tree-view.c (type_to_string): Delete.
(info_set_func_text): Update to print full signatures.

* test/glib/test-service-glib.xml: Change types to new
introspection format.

```

```

glib/dbus-binding-tool-glib.c | 20 ++++++-----
glib/dbus-gidl.c             | 18 ++++++-----
glib/dbus-gidl.h             | 8  ++---
glib/dbus-glib-tool.c        | 8  ++---
glib/dbus-gobject.c          | 69
+++++++-----
glib/dbus-gparser.c          | 72
+++++++-----
glib/dbus-gutils.c           | 42 -----
glib/dbus-gutils.h           | 1  -
glib/dbus-gvalue.c           | 71
+++++++-----
glib/dbus-gvalue.h           | 8  ++---
10 files changed, 131 insertions(+), 186 deletions(-)

```

commit 662755210c931da87972398ded13d4651e3d3fec
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2005-02-24 22:01:34 +0000

glib/Makefile.am: added dbus-gobject.h to sources list

so distcheck doesn't fail

glib/Makefile.am | 1 +
1 file changed, 1 insertion(+)

commit 0573676ce932874b07780036eaeaa42a6f3e1628
Author: Colin Walters <walters@verbum.org>
Date: 2005-02-19 23:25:41 +0000

2005-02-19 Colin Walters <walters@verbum.org>

* glib/dbus-binding-tool-glib.c
(dbus_binding_tool_output_glib_server): Fix iochannel
refcounting.

* glib/dbus-glib-tool.c: Include dbus-glib-tool.h, as well
as errno.h and sys/stat.h.
(lose): New function, prints error with
newline and exits.
(lose_gerror): Similar, but takes GError for message.
(main): Add --output argument to specify output file to write to,
instead of always printing to stdout. In this mode, determine
timestamps on source files to see whether any are newer than the
target file. If not, exit. Also convert a number of error
messages to use lose (since it's shorter), and switch to using
g_io_channel_shutdown.

glib/dbus-binding-tool-glib.c | 2 +-
glib/dbus-glib-tool.c | 119
+++++-----
2 files changed, 91 insertions(+), 30 deletions(-)

commit 7ff788981218b324d0eeec16fa4b7bb9ec97d694
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-02-19 16:17:29 +0000

2005-02-19 Havoc Pennington <hp@redhat.com>

* glib/dbus-gobject.c
(_dbus_glib_marshall_dbus_message_to_gvalue_array): add docs

* glib/dbus-glib.c: fix doxygen warnings

* glib/dbus-gparser.c (parse_annotation): error if an annotation
is found on an <arg>

```
glib/dbus-glib.c      |          8 ++++----
glib/dbus-gobject.c   |          8 ++++++--
glib/dbus-gparser.c   |          4 +---
3 files changed, 12 insertions(+), 8 deletions(-)
```

commit 03186401718758948242881cf429bdbb48eb8474
Author: Colin Walters <walters@verbum.org>
Date: 2005-02-18 03:14:33 +0000

2005-02-17 Colin Walters <walters@verbum.org>

```
* glib/dbus-gobject.h: Don't export
  _dbus_glib_marshall_dbus_message_to_gvalue_array.

* glib/dbus-gobject.c
  (_dbus_glib_marshall_dbus_message_to_gvalue_array): Do rename.
  (invoke_object_method): Handle it.

* glib/dbus-gproxy.c (marshal_dbus_message_to_g_marshaller):
  Handle rename.
```

```
glib/dbus-gobject.c |          4 ++--
glib/dbus-gobject.h |          2 +-
glib/dbus-gproxy.c  |          2 +-
3 files changed, 4 insertions(+), 4 deletions(-)
```

commit a6e8deb82f6202628fcb3b9b3a616d0a98598474
Author: Colin Walters <walters@verbum.org>
Date: 2005-02-17 21:19:48 +0000

2005-02-17 Colin Walters <walters@verbum.org>

```
* dbus/dbus-protocol.h (DBUS_SERVICE_ORG_FREEDESKTOP_DBUS):
  Rename to DBUS_SERVICE_DBUS.
  (DBUS_PATH_ORG_FREEDESKTOP_DBUS): Rename to DBUS_PATH_DBUS.
  (DBUS_PATH_ORG_FREEDESKTOP_LOCAL): Rename to DBUS_PATH_LOCAL.
  (DBUS_INTERFACE_ORG_FREEDESKTOP_DBUS): Rename to
  DBUS_INTERFACE_DBUS.
  (DBUS_INTERFACE_ORG_FREEDESKTOP_INTROSPECTABLE): Rename to
  DBUS_INTERFACE_INTROSPECTABLE.
  (DBUS_INTERFACE_ORG_FREEDESKTOP_PROPERTIES): Rename to
  DBUS_INTERFACE_PROPERTIES.
  (DBUS_INTERFACE_ORG_FREEDESKTOP_PEER): Rename to
  DBUS_INTERFACE_PEER.
  (DBUS_INTERFACE_ORG_FREEDESKTOP_LOCAL):
  DBUS_INTERFACE_LOCAL.
```

All other users of those constants have been changed.

```
* bus/driver.c (bus_driver_handle_introspect): Use constants.
```

* glib/dbus-gobject.c (handle_introspect): Use constants.

* doc/dbus-faq.xml, doc/dbus-specification.xml: Update for rename.

```
glib/dbus-gobject.c |      10 +++++-----  
glib/dbus-gproxy.c |       8 +++++-----  
2 files changed, 9 insertions(+), 9 deletions(-)
```

commit b085dcd0556e82c8fe347cc4bedd2e5b9b876a95

Author: Colin Walters <walters@verbum.org>

Date: 2005-02-17 21:11:18 +0000

2005-02-17 Colin Walters <walters@verbum.org>

* glib/dbus-gparser.c (struct Parser): Add in_annotation boolean.
(parse_node, parse_interface, parse_method, parse_signal)
(parse_property, parse_annotation): Lose if we're currently in an
annotation.

(parse_annotation): New function.

(parser_start_element, parser_end_element): Handle annotation.

(parse_method, parse_interface): Remove support for c_name
attribute,

switch to annotations.

* glib/dbus-gidl.h (interface_info_get_binding_names)
(method_info_get_binding_names)
(interface_info_get_binding_name, method_info_get_binding_name)
(interface_info_set_binding_name, method_info_set_binding_name):
Remove.

(interface_info_get_annotations, method_info_get_annotations)

(interface_info_get_annotation, method_info_get_annotation)

(interface_info_add_annotation, method_info_add_annotation):
Prototype.

* glib/dbus-gidl.c (struct InterfaceInfo): Substitute
"annotations"

for "bindings".

(struct MethodInfo): Ditto.

Straightfoward conversion of binding methods into annotation
methods

as prototyped.

* glib/dbus-glib-tool.c (pretty_print): Print annotations.

* glib/dbus-binding-tool-glib.h (DBUS_GLIB_ANNOTATION_C_SYMBOL):
Define.

* glib/dbus-binding-tool-glib.c (gather_marshallers,
generate_glue):

Use new annotation API.

* doc/introspect.dtd: Fix a number of DTD syntax errors. Add annotation element.

* doc/dbus-specification.xml: Discuss introspection annotations, include list of well-known annotations.

* test/glib/test-service-glib.xml: Make validate against new DTD.

```
glib/dbus-binding-tool-glib.c | 10 ++---
glib/dbus-binding-tool-glib.h |  2 +
glib/dbus-gidl.c              | 60 ++++++++-----
glib/dbus-gidl.h              | 24 +++++-----
glib/dbus-glib-tool.c         | 30 +++++-----
glib/dbus-gparser.c           | 88
+++++++-----
6 files changed, 143 insertions(+), 71 deletions(-)
```

commit 87e81b87d7b443142302ecb48e46191a443ae02a
Author: Colin Walters <walters@verbum.org>
Date: 2005-02-17 17:41:24 +0000

2005-02-17 Colin Walters <walters@verbum.org>

This patch is based on initial work from
Paul Kuliniewicz <kuliniew@purdue.edu>.

* glib/dbus-gvalue.c (dbus_gvalue_init): New function; move initialization of GValue from dbus type to here.
(dbus_gvalue_genmarshal_name_from_type): New function; generates a string for the "glib-genmarshal" program from a DBus type.
(dbus_gvalue_binding_type_from_type): New function; turns a DBus type into the C name for it we use in the glib bindings.
(dbus_gvalue_ctype_from_type): New function; maps a DBus type into a glib C type (not GValue).
(dbus_gvalue_demarshal): invoke dbus_gvalue_init.

* glib/dbus-gutils.c (_dbus_gutils_wincaps_to_uscore): Moved here from dbus-gobject.c.

* glib/dbus-gutils.h: Prototype it.

* glib/dbus-gproxy.c: Include new dbus-gobject.h.
(marshal_dbus_message_to_g_marshall): Use new shared function dbus_glib_marshal_dbus_message_to_gvalue_array.

* glib/dbus-gparser.c (parse_interface, parse_method): Handle c_name attribute.
Will be changed once we have annotations.

* glib/dbus-gobject.c: Change info_hash_mutex from GStaticMutex
to
GStaticRWLock. Callers updated.
(wincaps_to_uscore): Move to dbus-gutils.c. Callers updated.
(string_table_next): New function for iterating over
zero-terminated
string value array.
(string_table_lookup): New function; retrieves specific entry in
array.
(get_method_data): New function; look up method data in object
data chunk.
(object_error_domain_prefix_from_object_info)
(object_error_code_from_object_info): New functions, but not
implemented yet.
(method_interface_from_object_info): New function; retrieve
interface name.
(method_name_from_object_info): New function; retrieve method
name.
(method_arg_info_from_object_info): New function; retrieve
argument data.
(arg_iterate): New function; iterates over serialized argument
data.
(method_dir_signature_from_object_info): New function; returns a
GString holding type signature for arguments for just one
direction (input or output).
(method_input_signature_from_object_info)
(method_output_signature_from_object_info): New functions.
(dbus_glib_marshall_dbus_message_to_gvalue_array): New shared
function;
converts dbus message arguments into a GValue array. Used for
both
signal handling and method invocation.
(struct DBusGlibWriteInterfaceData): New utility structure.
(write_interface): New function; generate introspection XML for
an interface.
(introspect_interfaces): New function; gathers all
interface->methods,
generates introspection XML for them.
(handle_introspect): Invoke introspect_interfaces.
(get_object_property): Be sure to zero-initialize stack-allocated
GValue.
(lookup_object_and_method): New function; examines an incoming
message
and attempts to match it up (via interface, method name,
and argument
signature) with a known object and method.
(gerror_domaincode_to_dbus_error_name): New function; converts a
GError domain and code into a DBus error name. Needs GError
data
added to object introspection to work well.
(gerror_to_dbus_error_message): Creates a DBusMessage error
return from

GError.
 (invoke_object_method): New function to invoke an object method looked up via lookup_object_and_method. Parses the incoming message, turns it into a GValue array, then invokes the marshaller specified in the DBusGMethodInfo. Creates a new message with either return values or error message as appropriate.
 (gobject_message_function): Invoke lookup_object_and_method and invoke_object_method.

* glib/dbus-glib-tool.c: Include dbus-binding-tool-glib.h.
 (enum DBusBindingOutputMode): New enum for binding output modes.
 (pretty_print): Print binding names.
 (dbus_binding_tool_error_quark): GError bits.
 (version): Fix typo.
 (main): Create GIOChannel for output. Parse new --mode argument, possible values are "pretty-print", "glib-server", "glib-client". Use mode to invoke appropriate function.

* glib/dbus-gobject.h: Prototype
 dbus_glib_marshall_dbus_message_to_gvalue_array.

* glib/dbus-glib-tool.h: New header, just includes GError bits for now.

* glib/dbus-gidl.c (struct InterfaceInfo): Add bindings hashtable;
 maps binding style to name.
 (struct MethodInfo): Ditto.
 (get_hash_keys, get_hash_key): Utility function, returns keys for a GHashTable.
 (interface_info_new, method_info_new): Initialize bindings.
 (interface_info_unref, method_info_unref): Destroy bindings.
 (method_info_get_binding_names, method_info_get_binding_name)
 (interface_info_get_binding_names,
 interface_info_get_binding_name):
 Functions for retrieving binding names.
 (method_info_set_binding_name, interface_info_set_binding_name):
 Functions for setting binding names.

* glib/dbus-binding-tool-glib.h: New file, has prototypes for glib binding generation.

* glib/dbus-binding-tool-glib.c: New file, implements server-side and client-side glib glue generation.

* glib/Makefile.am (dbus_binding_tool_SOURCES): Add
 dbus-binding-tool-glib.c, dbus-binding-tool-glib.h,
 dbus-glib-tool.h.

* dbus/dbus-glib.h (struct DBusGMethodMarshaller): Remove in favor

of using GClosureMarshal directly.
(struct DBusGObjectInfo): Add n_infos member.

* test/glib/test-service-glib.xml: New file; contains introspection data for MyTestObject used in test-service-glib.c.

* test/glib/test-service-glib.c (enum MyObjectError): New GError enum.
(my_object_do_nothing, my_object_increment, my_object_throw_error)
(my_object_uppercase, my_object_many_args): New test methods.
(main): Use dbus_g_object_class_install_info to include generated object info.

* test/glib/Makefile.am: Generate server-side glue for test-service-glib.c, as well as client-side bindings.

* test/glib/test-dbus-glib.c: Include test-service-glib-bindings.h.
(main): Activate TestSuiteGLibService; test invoke a bunch of its methods using both the dbus_gproxy stuff directly as well as the generated bindings.

```
glib/Makefile.am          |    3 +
glib/dbus-binding-tool-glib.c |   812
+++++
glib/dbus-binding-tool-glib.h |    33 ++
glib/dbus-gidl.c          |    73 ++++
glib/dbus-gidl.h          |    14 +-
glib/dbus-glib-tool.c     |   120 +++++-
glib/dbus-glib-tool.h     |    37 ++
glib/dbus-gobject.c       |   596 ++++++-----
glib/dbus-gobject.h       |    35 ++
glib/dbus-gparser.c       |     8 +
glib/dbus-gproxy.c        |    52 +--
glib/dbus-gutils.c        |    28 ++
glib/dbus-gutils.h        |     1 +
glib/dbus-gvalue.c        |   171 ++++++-
glib/dbus-gvalue.h        |    28 +-
15 files changed, 1889 insertions(+), 122 deletions(-)
```

commit dd22216537d98b06366ade81eb6f63621d4c6379
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-02-16 04:37:27 +0000

2005-02-15 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_dispatch): always

complete a pending call, don't run filters first.

* glib/dbus-gproxy.c (dbus_g_proxy_end_call): change to use
dbus_pending_call_steal_reply

* dbus/dbus-pending-call.c (dbus_pending_call_block): just call
_dbus_connection_block_pending_call
(dbus_pending_call_get_reply): change to steal_reply and return a
ref

* dbus/dbus-connection.c
(dbus_connection_send_with_reply_and_block): port to work in
terms
of DBusPendingCall
(_dbus_connection_block_pending_call): replace block_for_reply
with this

glib/dbus-gproxy.c | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)

commit 3ce5325735aae6b1d4e6a28b45465a586710d23e
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-02-12 20:27:45 +0000

2005-02-12 Havoc Pennington <hp@redhat.com>

* tools/dbus-tree-view.c (info_set_func_text): display more
details on args

* bus/driver.c (bus_driver_handle_list_services): list the bus
driver

* glib/dbus-gparser.c (parse_arg): generate an arg name if none
is supplied

* glib/dbus-gidl.c (signal_info_get_n_args): new function
(method_info_get_n_args): new function

glib/dbus-gidl.c | 12 ++++++++
glib/dbus-gidl.h | 2 ++
glib/dbus-gparser.c | 13 ++++++++
3 files changed, 26 insertions(+), 1 deletion(-)

commit 89113d257f1f75cea5d0f39bbca4785afc784bf2
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-02-10 23:47:54 +0000

2005-02-10 Havoc Pennington <hp@redhat.com>

* dbus/dbus-object-tree.c (handle_default_introspect_and_unlock):
change to be _and_unlock instead of _unlocked

* dbus/dbus-connection.c
(_dbus_connection_send_preallocated_unlocked_no_update): rename
to
have no_update so we can find this bug quickly in future

glib/dbus-gthread.c | 2 ++
1 file changed, 2 insertions(+)

commit 116d931394d42412595e64185cd9cb97444cd653
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-02-05 04:15:57 +0000

2005-02-04 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c (dbus_g_proxy_disconnect_signal): use
g_quark_try_string() so it actually can return 0
(dbus_g_proxy_connect_signal): ditto

2005-02-04 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c (dbus_g_proxy_emit_remote_signal): fix a
bogus warning
(tristring_from_message): assert cleanly on null path/interface
(should not be possible though I decided later)
(dbus_g_proxy_dispose): move proxy manager unregistration here
(DBUS_G_PROXY_DESTROYED): add this macro, and use it in a bunch
of
g_return_if_fail() checks

glib/dbus-gproxy.c | 59
+++++-----
1 file changed, 46 insertions(+), 13 deletions(-)

commit 1fdc099a4ed729871597c501a9015ec2855bebc2
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-31 23:17:18 +0000

2005-01-31 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c: rewrite how signals work again, this time I
think it's sort of right

glib/dbus-gproxy.c | 334
+++++-----
1 file changed, 194 insertions(+), 140 deletions(-)

commit 463fb506e9661849d9838aedbdf4b7febe8e132e
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-31 02:55:12 +0000

2005-01-30 Havoc Pennington <hp@redhat.com>

```
* tools/dbus-names-model.c: dynamically watch NameOwnerChanged

* autogen.sh: change to autotools 1.9

* glib/dbus-gproxy.c: completely change how signals work
(dbus_g_proxy_add_signal): new function to specify signature of a
signal
(dbus_g_proxy_emit_received): marshal the dbus message to
GValues,
and g_warning if the incoming message has the wrong signature.
```

```
glib/Makefile.am      |   11 ++
glib/dbus-gmarshal.c  |   89 ++++++++
glib/dbus-gmarshal.h  |   21 +++
glib/dbus-gmarshal.list |    1 +
glib/dbus-gproxy.c    |  345
+++++-----
5 files changed, 390 insertions(+), 77 deletions(-)
```

```
commit c39a68d373c1f73a3bb7d0f45287ceelcd8eff55
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-30 23:06:32 +0000
```

```
2005-01-30 Havoc Pennington <hp@redhat.com>
```

```
* glib/dbus-glib.c (dbus_g_pending_call_set_notify): new function
(dbus_g_pending_call_cancel): new function

* dbus/dbus-glib.h: move GType decls for connection/message here;
* dbus/dbus-glib.c: move all the g_type and ref/unref stuff in
here, just kind of rationalizing how we handle all that

* tools/dbus-names-model.c: new file for a tree model listing the
services on a bus

* tools/dbus-tree-view.c (model_new): use proper typing on the
model rows
```

```
glib/dbus-glib.c      |   335
+++++-----
glib/dbus-gmain.c     |   172 -----
glib/dbus-gproxy.c    |    32 +----
3 files changed, 338 insertions(+), 201 deletions(-)
```

```
commit 967f6acd6c220b0be691edd18d9a636f4d16fbaf
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-30 20:06:52 +0000
```

```
2005-01-30 Havoc Pennington <hp@redhat.com>
```

```
* glib/dbus-gmain.c: add a custom GSource back that just checks
```

whether the message queue has anything in it; otherwise, there are cases where we won't see messages in the queue since there was no IO visible to the glib main loop

* dbus/dbus-connection-internal.h (_DBUS_DEFAULT_TIMEOUT_VALUE): increase default message timeout to 25 seconds

glib/dbus-gmain.c | 93
+++++-----
1 file changed, 86 insertions(+), 7 deletions(-)

commit af3c2731c74f014c0713a52d66027a9428c1ddaf
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-30 19:33:29 +0000

2005-01-30 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c: rewrite the main loop stuff to avoid the custom source, seems to be a lot easier to understand and work better.

glib/dbus-gmain.c | 653
+++++-----
1 file changed, 313 insertions(+), 340 deletions(-)

commit 32587c38401f16a9477f7ea7868120645d072462
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-30 18:25:14 +0000

2005-01-30 Havoc Pennington <hp@redhat.com>

I think this main loop thing is conceptually broken, but here are some band aids. I'll maybe rewrite it in a minute.

* glib/dbus-gmain.c (add_timeout): timeout stuff doesn't use the custom GSource, so don't pass it in; confusing (gsource_server_finalize, gsource_connection_finalize): add finalize handlers that remove all the watches.

glib/dbus-gmain.c | 90
+++++-----
1 file changed, 74 insertions(+), 16 deletions(-)

commit 4b7acbbfacaa37ce399df1ce616920c09750ae60
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-30 07:44:08 +0000

2005-01-30 Havoc Pennington <hp@redhat.com>

* glib/dbus-gobject.c (introspect_properties): fix the XML

generated

* dbus/dbus-message.c (dbus_message_unref): add an in_cache flag which effectively detects the use of freed messages

* glib/dbus-gobject.c (handle_introspect): modify and return the reply message instead of the incoming message

* dbus/dbus-object-tree.c (handle_default_introspect_unlocked): gee, maybe it should SEND THE XML instead of just making a string and freeing it again ;-)

* tools/dbus-print-message.c (print_message): improve printing of messages

* configure.in: add debug-glib.service to the output

```
glib/dbus-gobject.c |      10 +++++-----
glib/dbus-gparser.c |       2 ++
2 files changed, 7 insertions(+), 5 deletions(-)
```

commit 71181795196712bbb64d721464f42432997c7d0c

Author: Havoc Pennington <hp@redhat.com>

Date: 2005-01-30 05:18:44 +0000

2005-01-30 Havoc Pennington <hp@redhat.com>

dbus-viewer introspected and displayed the bus driver

* dbus/dbus-object-tree.c
(object_tree_test_iteration): add tests for a handler registered on "/"

* dbus/dbus-object-tree.c
(_dbus_decompose_path): fix to handle path "/" properly
(run_decompose_tests): add tests for path decomposition

* glib/dbus-gutils.c (_dbus_gutils_split_path): fix to handle "/" properly

* glib/dbus-gobject.c (handle_introspect): fix quotes

* test/glib/run-test.sh: support launching the bus, then running dbus-viewer

* test/glib/test-service-glib.c (main): put in a trivial gobject subclass and register it on the connection

* bus/driver.c (bus_driver_handle_introspect): implement introspection of the bus driver service

* dbus/dbus-protocol.h: add #defines for the XML namespace,

identifiers, doctype decl

* bus/driver.c (bus_driver_handle_get_service_owner): handle attempts to get owner of DBUS_SERVICE_ORG_FREEDESKTOP_DBUS by returning the service unchanged.
(bus_driver_handle_message): remove old check for reply_serial in method calls, now the message type deals with that
(bus_driver_handle_message): handle NULL interface

* glib/dbus-gproxy.c (dbus_g_proxy_get_bus_name): new function

* glib/dbus-gloader-expat.c (description_load_from_string): allow -1 for len

* tools/dbus-viewer.c: add support for introspecting a service on a bus

* glib/dbus-gproxy.c (dbus_g_pending_call_ref): add
(dbus_g_pending_call_unref): add

```
glib/dbus-gidl.c      | 14 ++++++++
glib/dbus-gidl.h      | 3 +++
glib/dbus-gloader-expat.c | 4 +++
glib/dbus-gobject.c  | 16 ++++++-----
glib/dbus-gproxy.c   | 62
+++++-----
glib/dbus-gutils.c   | 20 ++++++-----
6 files changed, 98 insertions(+), 21 deletions(-)
```

commit 3593b814632b482b541ccf3adb2b31887028b234
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-29 20:12:21 +0000

2005-01-29 Havoc Pennington <hp@redhat.com>

* tools/dbus-tree-view.c: add support for displaying properties.
(run dbus-viewer with an introspect xml file as arg, then resize the window so the tree elements show up, not sure what that is)

* glib/dbus-gobject.c (handle_introspect): return org.freedesktop.Properties and org.freedesktop.Introspectable interfaces when we are introspected.

* doc/dbus-specification.xml: allow empty interface name when Get/Set a property

```
glib/dbus-gobject.c | 28 ++++++-----
1 file changed, 26 insertions(+), 2 deletions(-)
```

commit 426cf6b944282d16db0d88b230627793df52f1ad
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-29 19:52:19 +0000

2005-01-29 Havoc Pennington <hp@redhat.com>

- * glib/Makefile.am: rename dbus-glib-tool to dbus-binding-tool; though it uses glib, it could be extended for any binding in principle
- * glib/dbus-gobject.c (gobject_message_function): change to the new way properties work
- * dbus/dbus-protocol.h: add the new interfaces
- * doc/dbus-specification.xml: document the introspection format, Introspectable interface, and add an org.freedesktop.Properties interface.
- * glib/dbus-gparser.c: add support for a <property> element
- * glib/dbus-gidl.c: add PropertyInfo
- * glib/dbus-gobject.c (handle_introspect): put the outermost <node> outside the signal and property descriptions.
(introspect_properties): export properties as <property> rather than as method calls

```
glib/.cvsignore      |    2 +-
glib/Makefile.am     |    6 +-
glib/dbus-gidl.c     |  100 ++++++
glib/dbus-gidl.h     |  112 ++++++-----
glib/dbus-glib-tool.c |   31 +++++-
glib/dbus-gobject.c  |  104 ++++++-----
glib/dbus-gparser.c  |   190
+++++
7 files changed, 434 insertions(+), 111 deletions(-)
```

commit 54c56014a0d2dab88f1fa3bd17a7c1fd2768c4e9
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-28 03:06:55 +0000

2005-01-27 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-arch-deps.h.in: add 16/32-bit types
- * configure.in: find the right type for 16 and 32 bit ints as well
as 64
- * dbus/dbus-protocol.h (DBUS_TYPE_INT16, DBUS_TYPE_UINT16): add the 16-bit types so people don't have to stuff them in 32-bit or byte arrays.

```
glib/dbus-gparser.c |    4 ++++
```

```
glib/dbus-gutils.c |      4 ++++
glib/dbus-gvalue.c |     29 ++++++-----
3 files changed, 31 insertions(+), 6 deletions(-)
```

```
commit 459e046b1ca4bdd06d85458fd4f213354833d1ed
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-21 05:06:10 +0000
```

2005-01-21 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c: don't put the GLib bindings in the same toplevel doxygen group as the low-level API stuff

* dbus/dbus.h: note that libdbus is the low-level API

```
glib/dbus-gmain.c |    12 ++++++-----
1 file changed, 7 insertions(+), 5 deletions(-)
```

```
commit 8cecb5bad77434129fc751c2ace45c389f46ef0d
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-18 20:42:15 +0000
```

2005-01-18 Havoc Pennington <hp@redhat.com>

* Throughout, grand renaming to strip out the use of "service", just say "name" instead (or "bus name" when ambiguous). Did not change the internal code of the message bus itself, only the programmer-facing API and messages.

* doc/dbus-specification.xml: further update the message bus section

* bus/config-parser.c (all_are_equiv): fix bug using freed string in error case

```
glib/dbus-gproxy.c |   171
+++++-----
1 file changed, 88 insertions(+), 83 deletions(-)
```

```
commit 40c882bcb60faeee24e3e887f3503e9ecb878166
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-17 19:49:52 +0000
```

2005-01-17 Havoc Pennington <hp@redhat.com>

* dbus/dbus-types.h: hardcode dbus_bool_t to 32 bits

* Throughout: modify DBUS_TYPE_BOOLEAN to be a 32-bit type instead of an 8-bit type. Now dbus_bool_t is the type to use whenever you are marshaling/unmarshaling a boolean.

```
glib/dbus-gvalue.c | 4 +++-
1 file changed, 3 insertions(+), 1 deletion(-)
```

```
commit c8c3d8a243f6b284b3aa46352afb9fb913f09f78
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-16 15:51:55 +0000
```

```
2005-01-16 Havoc Pennington <hp@redhat.com>
```

```
* Add and fix docs according to Doxygen warnings throughout
source.
```

```
* dbus/dbus-marshal-recursive.c
(_dbus_type_reader_array_is_empty): change this to just call
array_reader_get_array_len() and make it static
```

```
* dbus/dbus-message.c (dbus_message_iter_get_element_type):
rename
from get_array_type
(dbus_message_iter_init_append): rename from append_iter_init
```

```
* dbus/dbus-marshal-recursive.c
(_dbus_type_reader_get_element_type): rename from
_dbus_type_reader_get_array_type
```

```
glib/dbus-glib.c | 2 +-
glib/dbus-gobject.c | 2 +-
2 files changed, 2 insertions(+), 2 deletions(-)
```

```
commit e116119209a507a5ba305e890f897d31e5b626e1
Author: Havoc Pennington <hp@redhat.com>
Date: 2005-01-15 07:15:38 +0000
```

```
2005-01-15 Havoc Pennington <hp@redhat.com>
```

```
* Land the new message args API and type system.
```

```
This patch is huge, but the public API change is not
really large. The set of D-BUS types has changed somewhat,
and the arg "getters" are more geared toward language bindings;
they don't make a copy, etc.
```

```
There are also some known issues. See these emails for details
on this huge patch:
http://lists.freedesktop.org/archives/dbus/2004-December/001836.html
http://lists.freedesktop.org/archives/dbus/2005-January/001922.html
```

```
* dbus/dbus-marshal-*: all the new stuff
```

```

* dbus/dbus-message.c: basically rewritten

* dbus/dbus-memory.c (check_guards): with "guards" enabled, init
freed blocks to be all non-nul bytes so using freed memory is
less
likely to work right

* dbus/dbus-internals.c (_dbus_test_oom_handling): add
DBUS_FAIL_MALLOC=N environment variable, so you can do
DBUS_FAIL_MALLOC=0 to skip the out-of-memory checking, or
DBUS_FAIL_MALLOC=10 to make it really, really, really slow and
thorough.

* qt/message.cpp: port to the new message args API
(operator<<): use str.utf8() rather than str.unicode()
(pretty sure this is right from the Qt docs?)

* glib/dbus-gvalue.c: port to the new message args API

* bus/dispatch.c, bus/driver.c: port to the new message args API

* dbus/dbus-string.c (_dbus_string_init_const_len): initialize
the
"locked" flag to TRUE and align_offset to 0; I guess we never
looked at these anyhow, but seems cleaner.

* dbus/dbus-string.h (_DBUS_STRING_ALLOCATION_PADDING):
move allocation padding macro to this header; use it to implement
(_DBUS_STRING_STATIC): ability to declare a static string.

* dbus/dbus-message.c (_dbus_message_has_type_interface_member):
change to return TRUE if the interface is not set.

* dbus/dbus-string.[hc]: move the D-BUS specific validation stuff
to dbus-marshal-validate.[hc]

* dbus/dbus-marshal-basic.c (_dbus_type_to_string): move here
from
dbus-internals.c

* dbus/Makefile.am: cut over from dbus-marshal.[hc]
to dbus-marshal-*. [hc]

* dbus/dbus-object-tree.c (_dbus_decompose_path): move this
function here from dbus-marshal.c

glib/dbus-gobject.c |          2 +-
glib/dbus-gproxy.c  |          7 +-
glib/dbus-gutils.c  |         24 ++++---
glib/dbus-gvalue.c  |         186
+++++-----
4 files changed, 156 insertions(+), 63 deletions(-)

```

commit 62f29fd0fea8e1eaa3e61db44ddd1eeba765f153
Author: Havoc Pennington <hp@redhat.com>
Date: 2004-11-13 07:07:46 +0000

2004-11-13 Havoc Pennington <hp@redhat.com>

- * test/glib/test-profile.c: fix this thing up a bit
- * dbus/dbus-message.c (dbus_message_new_empty_header): increase preallocation sizes by a fair bit; not sure if this will be an overall performance win or not, but it does reduce reallocs.
- * dbus/dbus-string.c (set_length, reallocate_for_length): ignore the test hack that forced constant realloc if asserts are disabled, so we can profile sanely. Sprinkle in some `_DBUS_UNLIKELY()` which are probably pointless, but before I noticed the real performance problem I put them in.
(`_dbus_string_validate_utf8`): micro-optimize this thing a little bit, though callgrind says it didn't help; then special-case `ascii`, which did help a lot; then be sure we detect nul bytes as invalid, which is a bugfix.
(`align_length_then_lengthen`): add some more `_DBUS_UNLIKELY` superstition; use `memset` to nul the padding instead of a manual loop.
(`_dbus_string_get_length`): inline this as a macro; it showed up in the profile because it's used for loop tests and so forth

glib/dbus-gmain.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 0ab857b2916731f92be856c6821d5c9950716f89
Author: Olivier Andrieu <oliv__a@users.sourceforge.net>
Date: 2004-10-07 09:56:01 +0000

- * dbus/dbus-sysdeps.c (`_dbus_file_get_contents`): fix an incorrect format string.
- * glib/dbus-dbus-gmain.c (`dbus_g_bus_get`): do not mangle NULL pointer
(bug #1540, Leonardo Boiko).

glib/dbus-gmain.c | 9 ++++-----
1 file changed, 4 insertions(+), 5 deletions(-)

commit 4b50c510605d5dda629c9d493a61611bb382a55d
Author: Havoc Pennington <hp@redhat.com>
Date: 2004-08-10 03:06:59 +0000

2004-08-09 Havoc Pennington <hp@redhat.com>

* COPYING: switch to Academic Free License version 2.1 instead of 2.0, to resolve complaints about patent termination clause.

```
glib/dbus-gidl.c      | 2 +-
glib/dbus-gidl.h      | 2 +-
glib/dbus-glib-tool.c | 2 +-
glib/dbus-glib.c      | 2 +-
glib/dbus-gloader-expat.c | 2 +-
glib/dbus-gmain.c     | 2 +-
glib/dbus-gobject.c   | 2 +-
glib/dbus-gparser.c   | 2 +-
glib/dbus-gparser.h   | 2 +-
glib/dbus-gproxy.c    | 4 ++--
glib/dbus-gtest-main.c | 2 +-
glib/dbus-gtest.c     | 2 +-
glib/dbus-gtest.h     | 2 +-
glib/dbus-gthread.c   | 2 +-
glib/dbus-gtool-test.h | 2 +-
glib/dbus-gutils.c    | 2 +-
glib/dbus-gutils.h    | 2 +-
17 files changed, 18 insertions(+), 18 deletions(-)
```

```
commit 301a3547a2efd9784fa3ab253373fa689c9bd1c3
Author: Olivier Andrieu <oliv__a@users.sourceforge.net>
Date: 2004-07-29 08:00:45 +0000
```

* bus/config-loader-libxml.c: complete the implementation of libxml

backend for config file loader. Doesn't work with full OOM test yet.

* configure.in: change error when selecting libxml into a warning.

* test/data/invalid-config-files: add two non-well-formed XML files.

* glib/Makefile.am: libdbus_gtool always uses expat, not libxml.

* dbus/dbus-transport-unix.c (unix_handle_watch): do not disconnect in case of DBUS_WATCH_HANGUP, several do_reading() may be necessary to read all the buffer. (bug #894)

* bus/activation.c (bus_activation_activate_service): fix a potential assertion failure (bug #896). Small optimization in the case of auto-activation messages.

* dbus/dbus-message.c (verify_test_message, _dbus_message_test): add

test case for byte-through-vararg bug (#901). patch by Kimmo Hämäläinen.

glib/Makefile.am | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit ef7e145899ce58a5df564127783f014d8bed233f
Author: Havoc Pennington <hp@redhat.com>
Date: 2004-06-20 15:28:14 +0000

2004-06-20 Havoc Pennington <hp@redhat.com>

* dbus/dbus-glib-error-enum.h: autogenerate the GError enum codes from the dbus error names

* glib/dbus-glib.h: move to subdir dbus/ since it's included as dbus/dbus-glib.h and that breakage is now visible due to including dbus/dbus-glib.h in dbus-glib-lowlevel.h

* glib/dbus-glib.h: s/gproxy/g_proxy/

* dbus/dbus-shared.h: new header to hold stuff shared with binding APIs

* dbus/dbus-protocol.h (DBUS_ERROR_*): move errors here rather than dbus-errors.h

* glib/dbus-glib.h (dbus_set_g_error): move to dbus-glib-lowlevel.h

* glib/dbus-glib.h: remove dbus/dbus.h from here; change a bunch of stuff to enable this

* dbus/dbus-glib-lowlevel.h: put dbus/dbus.h here

* a bunch of other changes with the same basic "separate glib bindings from dbus.h" theme

```
glib/Makefile.am      |    6 +-
glib/dbus-gidl.c      |    2 +-
glib/dbus-glib-tool.c |    2 +-
glib/dbus-glib.c      |   67 ++++++
glib/dbus-glib.h      |  159 -----
glib/dbus-gmain.c     |  121 ++++++
glib/dbus-gobject.c   |   17 +-
glib/dbus-gproxy.c    |  237
+++++-----
glib/dbus-gtest.c     |    4 +
glib/dbus-gtest.h     |    9 +-
glib/dbus-gthread.c   |    3 +-
glib/dbus-gtool-test.h |    4 +-
glib/dbus-gutils.c    |    2 +-

```



```
glib/dbus-gutils.h | 15 +++
14 files changed, 342 insertions(+), 306 deletions(-)
```

```
commit 9426e419227a721a8b7b04661ec133cb93c389e5
Author: Olivier Andrieu <oliv__a@users.sourceforge.net>
Date: 2004-06-05 16:32:00 +0000
```

```
2004-06-05 Olivier Andrieu <oliv__a@users.sourceforge.net>
```

```
* dbus/dbus-connection.h, dbus/dbus-connection.c: have object
path
registration functions take the path argument as char* instead of
char**.
```

```
* dbus/dbus-marshal.h, dbus/dbus-marshal.c
(_dbus_decompose_path):
split off the path decomposition part of
_dbus_demarshal_object_path. Some misc. fixes to silence compiler
warnings.
```

```
* glib/dbus-gobject.c, test/test-service.c: update accordingly.
```

```
glib/dbus-gobject.c | 16 +++-----
1 file changed, 3 insertions(+), 13 deletions(-)
```

```
commit b2a4e237f907acd5127f85d0e2110b32d8a7ea86
Author: Kristian Hogsberg <krh@redhat.com>
Date: 2004-06-02 13:13:14 +0000
```

```
2004-06-02 Kristian Höggsberg <krh@redhat.com>
```

```
* glib/dbus-gproxy.c, glib/dbus-gmain.c, dbus/dbus-string.c,
dbus/dbus-object-tree.c, dbus/dbus-message.c: add comments to
quiet doxygen.
```

```
* Doxyfile.in: remove deprecated options.
```

```
* dbus/dbus-message-handler.c, dbus/dbus-message-handler.h,
glib/test-thread.h, glib/test-thread-client.c,
glib/test-thread-server.c, glib/test-profile.c,
glib/test-dbus-glib.c: remove these unused files.
```

```
glib/dbus-gmain.c | 12 ++-
glib/dbus-gproxy.c | 32 +++++-
glib/test-dbus-glib.c | 50 -----
glib/test-profile.c | 223 -----
```

```
glib/test-thread-client.c | 95 -----
glib/test-thread-server.c | 248
```

```
glib/test-thread.h | 1 -
7 files changed, 36 insertions(+), 625 deletions(-)
```

commit 05a2332400f9e23d84e7376b9908f19a59868d5d
Author: Havoc Pennington <hp@redhat.com>
Date: 2004-06-01 03:02:26 +0000

2004-05-31 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gidl.c (method_info_add_arg): keep args sorted with "in" before "out"
- * glib/dbus-gobject.c (dbus_type_to_string): move to dbus-gutils.c
- * glib/dbus-glib-tool.c (main): set up to have a --self-test option that runs the tests, and start filling in some code including for starters just dumping the interfaces to stdout
- * glib/Makefile.am (INCLUDES): define DBUS_LOCALEDIR
- * test/data/valid-introspection-files/lots-of-types.xml: test of an example introspection file
- * glib/dbus-gparser.c (parser_check_doctype): doctype should be "node" (I think...)

```
glib/Makefile.am      |    9 +-  
glib/dbus-gidl.c     |   25 +++++  
glib/dbus-glib-tool.c |  280  
+++++-----  
glib/dbus-glib.h      |    5 +-  
glib/dbus-gobject.c  |   36 +-----  
glib/dbus-gparser.c  |    4 +-  
glib/dbus-gutils.c   |   34 ++++++  
glib/dbus-gutils.h   |    4 +-  
8 files changed, 338 insertions(+), 59 deletions(-)
```

commit c02ba470d7a212d7c51777c467eae808dd954f2f
Author: Havoc Pennington <hp@redhat.com>
Date: 2004-05-28 22:30:04 +0000

2004-05-28 Havoc Pennington <hp@redhat.com>

- * test/glib/test-service-glib.c (main): remove unused variable
- * glib/dbus-gidl.c (base_info_ref): fix a silly compiler warning
- * dbus/dbus-auth.h (enum): remove AUTHENTICATED_WITH_UNUSED_BYTES from the enum, no longer in use.
- * dbus/dbus-sysdeps.h: include config.h so DBUS_VA_COPY actually works right.

* dbus/dbus-message.c: add various `_dbus_return_val_if_fail` for whether `error_name` passed in is a valid error name.

glib/dbus-gidl.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

commit 1400b16528f4d0d8f6a60b199e5799eadb0ff0ea
Author: Michael Meeks <michael@ximian.com>
Date: 2004-05-28 13:10:35 +0000

2004-05-28 Michael Meeks <michael@ximian.com>

* glib/dbus-gvalue.c (`dbus_gvalue_marshal`,
`dbus_gvalue_demarshal`):
fix no int64 case.

* dbus/dbus-string.c (`_dbus_string_parse_basic_type`): impl.

* dbus/dbus-message.c (`_dbus_message_iter_get_basic_type`),
(`_dbus_message_iter_get_basic_type_array`): impl.
drastically simplify ~all relevant `_get` methods to use these.
(`_dbus_message_iter_append_basic_array`),
(`dbus_message_iter_append_basic`): impl
drastically simplify ~all relevant `_append` methods to use these.

* dbus/dbus-message-builder.c (`parse_basic_type`)
(`parse_basic_array`, `lookup_basic_type`): impl.
(`_dbus_message_data_load`): prune scads of duplicate /
cut & paste coding.

* dbus/dbus-marshal.c (`_dbus_demarshal_basic_type_array`)
(`_dbus_demarshal_basic_type`): implement,
(`demarshal_and_validate_len/arg`): beef up debug.
(`_dbus_marshal_basic_type`, `_dbus_marshal_basic_type_array`): impl.

glib/dbus-gvalue.c | 4 ++++
1 file changed, 4 insertions(+)

commit 7866137fa3b028e7a8e22f03998754c0967e1501
Author: Olivier Andrieu <oliv__a@users.sourceforge.net>
Date: 2004-04-15 22:08:04 +0000

2004-04-15 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/driver.c (`bus_driver_handle_get_service_owner`):
implement a `GetServiceOwner` method.
* doc/dbus-specification.xml: document it.
* dbus/dbus-errors.h: add a 'ServiceHasNoOwner' error.

* glib/dbus-gproxy.c (`dbus_gproxy_new_for_service_owner`):
implement, using the bus `GetServiceOwner` method.

* test/glib/test-dbus-glib.c:
use dbus_gproxy_new_for_service_owner so that we can receive the
signal.

glib/dbus-gproxy.c | 55
++++
1 file changed, 55 insertions(+)

commit e03a715a21237d70233d40a393958ec7d3d4027d
Author: Michael Meeks <michael@ximian.com>
Date: 2004-04-13 11:47:17 +0000

2004-04-13 Michael Meeks <michael@ximian.com>

* glib/dbus-gobject.c (handle_introspect): split out
(introspect_properties): this.
(handle_introspect): implement this.

glib/dbus-gobject.c | 123
++++-----
1 file changed, 82 insertions(+), 41 deletions(-)

commit e7e5d9377e54b9d26a59402197d26bb575c15976
Author: Michael Meeks <michael@ximian.com>
Date: 2004-03-29 13:24:50 +0000

2004-03-29 Michael Meeks <michael@ximian.com>

* glib/dbus-gobject.c (set_object_property): split out /
re-work, use the property type, and not the message type(!)
(get_object_property): ditto.

* glib/dbus-gvalue.c (dbus_gvalue_demarshal),
(dbus_gvalue_marshal): make this code re-usable, needed
for signals too, also on both proxy and server side.
Re-write for more efficiency / readability.

glib/Makefile.am | 4 +-
glib/dbus-gobject.c | 187
+++-----
glib/dbus-gvalue.c | 115 +++++
glib/dbus-gvalue.h | 16 +++++
4 files changed, 142 insertions(+), 180 deletions(-)

commit 82d9c846687c428fde8e517ca38d69642de9a66a
Author: Richard Hult <richard@imendio.com>
Date: 2003-12-02 10:44:21 +0000

2003-12-02 Richard Hult <richard@imendio.com>

* Update AFL version to 2.0 throughout the source files to
reflect

the update that was done a while ago.

```
glib/dbus-gidl.c      | 2 +-
glib/dbus-gidl.h      | 2 +-
glib/dbus-glib-tool.c | 2 +-
glib/dbus-glib.h      | 2 +-
glib/dbus-gloader-expat.c | 2 +-
glib/dbus-gmain.c     | 2 +-
glib/dbus-gobject.c   | 2 +-
glib/dbus-gparser.c   | 2 +-
glib/dbus-gparser.h   | 2 +-
glib/dbus-gproxy.c    | 2 +-
glib/dbus-gtest-main.c | 2 +-
glib/dbus-gtest.c     | 2 +-
glib/dbus-gtest.h     | 2 +-
glib/dbus-gthread.c   | 2 +-
glib/dbus-gtool-test.h | 2 +-
glib/dbus-gutils.c    | 2 +-
glib/dbus-gutils.h    | 2 +-
glib/test-profile.c   | 2 +-
18 files changed, 18 insertions(+), 18 deletions(-)
```

```
commit 64858892747bea931226f39dad9e063475bf3c89
Author: Mikael Hallendal <micke@imendio.com>
Date: 2003-11-27 01:25:49 +0000
```

2003-11-26 Mikael Hallendal <micke@imendio.com>

- * bus/*.ch:
- * dbus/*.ch:
- * glib/*.ch: Made ref functions return the pointer

```
glib/dbus-gidl.c      | 24 ++++++-----
glib/dbus-gidl.h      | 12 +++++-----
glib/dbus-gmain.c     | 4 +++-
glib/dbus-gparser.c   | 4 +++-
glib/dbus-gparser.h   | 2 +-
glib/dbus-gproxy.c    | 6 ++++-
6 files changed, 35 insertions(+), 17 deletions(-)
```

```
commit f130073ef0a107d0b57695720f5a94d0e17ce38f
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-10-21 05:46:51 +0000
```

2003-10-20 Havoc Pennington <hp@redhat.com>

hmm, make check is currently not passing.

- * doc/dbus-specification.xml: add requirement that custom type names follow the same rules as interface names.

- * dbus/dbus-protocol.h: change some of the byte codes, to avoid

duplication and allow 'c' to be 'custom'; dict is now 'm' for 'map'

to * doc/dbus-specification.xml: update type codes to match dbus-protocol.h, using the ASCII byte values. Rename type NAMED CUSTOM. Add type OBJECT_PATH to the spec.

2003-10-17 Havoc Pennington <hp@redhat.com>

* bus/driver.c (create_unique_client_name): use "." as separator in base service names instead of '-'

* dbus/dbus-string.c (_dbus_string_get_byte): allow getting nul byte at the end of the string

* dbus/dbus-internals.h (_DBUS_LIKELY, _DBUS_UNLIKELY): add optimization macros since string validation seems to be a slow point.

* doc/dbus-specification.xml: restrict valid service/interface/member/error names. Add test suite code for the name validation.

* dbus/dbus-string.c: limit service/interface/member/error names to [0-9][A-Z][a-z]_

* dbus/dbus-connection.c (dbus_connection_dispatch): add missing format arg to verbose spew

* glib/dbus-gproxy.c (dbus_gproxy_call_no_reply): if not out of memory, return instead of g_error

* test/test-service.c (path_message_func): support emitting a signal on request

* dbus/dbus-bus.c (init_connections_unlocked): only fill in activation bus type if DBUS_BUS_ACTIVATION was set; default to assuming the activation bus was the session bus so that services started manually will still register.
(init_connections_unlocked): fix so that in OOM situation we get the same semantics when retrying the function

* test/test-service.c (main): change to use path registration, to test those codepaths; register with DBUS_BUS_ACTIVATION rather than DBUS_BUS_SESSION

glib/dbus-gobject.c | 4 +++-
glib/dbus-gproxy.c | 12 ++++++++--
2 files changed, 13 insertions(+), 3 deletions(-)

commit 756f88a96232ae730faca81feeb30fcd07458df9

Author: Havoc Pennington <hp@redhat.com>
Date: 2003-10-17 16:23:19 +0000

2003-10-16 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gtest-main.c: bracket with #ifdef DBUS_BUILD_TESTS
- * Makefile.am (GCOV_DIRS): remove "test", we don't care about test coverage of the tests (coverage-report.txt): don't move the .da and .bbg files around

glib/dbus-gtest-main.c | 5 +++++
1 file changed, 5 insertions(+)

commit aa0c367ff5214ddb2578e5b290cfae6aa472a0d
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-10-16 06:34:50 +0000

2003-10-16 Havoc Pennington <hp@redhat.com>

- * bus/connection.c (bus_pending_reply_expired): either cancel or execute, not both (bus_connections_check_reply): use unlink, not remove_link, as we don't want to free the link; fixes double free mess
- * dbus/dbus-pending-call.c (dbus_pending_call_block): fix in case where no reply was received
- * dbus/dbus-connection.c (_dbus_pending_call_complete_and_unlock): fix a refcount leak
- * bus/signals.c (match_rule_matches): add special cases for the bus driver, so you can match on sender/destination for it.
- * dbus/dbus-sysdeps.c (_dbus_abort): print backtrace if DBUS_PRINT_BACKTRACE is set
- * dbus/dbus-internals.c: add pid to assertion failure messages
- * dbus/dbus-connection.c: add message type code to the debug spew
- * glib/dbus-gproxy.c (gproxy_get_match_rule): match rules want sender=foo not service=foo
- * dbus/dbus-bus.c (dbus_bus_get): if the activation bus is the session bus but DBUS_SESSION_BUS_ADDRESS isn't set, use DBUS_ACTIVATION_ADDRESS instead
- * bus/activation.c: set DBUS_SESSION_BUS_ADDRESS, DBUS_SYSTEM_BUS_ADDRESS if appropriate

* bus/bus.c (bus_context_new): handle OOM copying bus type into context struct

* dbus/dbus-message.c (dbus_message_iter_get_object_path): new function
(dbus_message_iter_get_object_path_array): new function (half finished, disabled for the moment)

* glib/dbus-gproxy.c (dbus_gproxy_end_call): properly handle DBUS_MESSAGE_TYPE_ERROR

* tools/dbus-launch.c (babysit): support DBUS_DEBUG_OUTPUT to avoid redirecting stderr to /dev/null
(babysit): close stdin if not doing the "exit_with_session" thing

* dbus/dbus-sysdeps.c (_dbus_become_daemon): delete some leftover debug code; change DBUS_DEBUG_OUTPUT to only enable stderr, not stdout/stdin, so things don't get confused

* bus/system.conf.in: fix to allow replies, I modified .conf instead of .conf.in again.

glib/dbus-gproxy.c | 26 ++++++-----
1 file changed, 20 insertions(+), 6 deletions(-)

commit 33689cc9f76ff14f77ba9b9d6e17094991fc2ba1
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-10-12 05:59:39 +0000

2003-10-12 Havoc Pennington <hp@pobox.com>

Added test code that 1) starts an actual bus daemon and
2) uses
DBusGProxy; fixed bugs that were revealed by the test. Lots
more testing possible, but this is the basic framework.

* glib/dbus-gproxy.c (dbus_gproxy_manager_unregister): remove empty proxy lists from the proxy list hash

* dbus/dbus-message.c (dbus_message_iter_get_args_valist): add a couple of return_if_fail checks

* dbus/dbus-pending-call.c (_dbus_pending_call_new): use
dbus_new0
to allocate, so everything is cleared to NULL as it should be.

* glib/dbus-gmain.c (dbus_connection_setup_with_g_main): pass source as data to dbus_connection_set_timeout_functions() as the timeout functions expected


```
* test/glib/run-test.sh: add a little script to start up a
message
bus and run tests using it

* tools/dbus-launch.1: updates

* tools/dbus-launch.c (main): add --config-file option

* tools/dbus-launch.c (main): remove confusing else if (runprog)
that could never be reached.

* dbus/dbus-message.c (dbus_message_new_method_return)
(dbus_message_new_error, dbus_message_new_signal): set the
no-reply-expected flag on all these. Redundant, but may
as well be consistent.
```

```
glib/dbus-gmain.c | 2 +-
glib/dbus-gproxy.c | 17 ++++++++-----
2 files changed, 13 insertions(+), 6 deletions(-)
```

```
commit 6741526f3dedd835d9f0d7ae4beaa097016b0b9e
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-10-03 03:55:35 +0000
```

2003-10-02 Havoc Pennington <hp@pobox.com>

```
* glib/dbus-gproxy.c (dbus_gproxy_call_no_reply): rename from
dbus_gproxy_oneway_call

* glib/dbus-gmain.c (dbus_connection_setup_with_g_main)
(dbus_server_setup_with_g_main): fix to allow calling them more
than once on the same args
(dbus_bus_get_with_g_main): new function
```

```
glib/dbus-glib.h | 14 +++++----
glib/dbus-gmain.c | 99
+++++-----
glib/dbus-gproxy.c | 8 +-
3 files changed, 97 insertions(+), 24 deletions(-)
```

```
commit b05fc993d8d813ca93fa2c49e4c8b7345613c8e0
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-09-30 02:32:48 +0000
```

2003-09-29 Havoc Pennington <hp@pobox.com>

```
* Merge dbus-object-names branch. To see the entire patch
do cvs diff -r DBUS_OBJECT_NAMES_BRANCHPOINT -r dbus-object-
names,
it's huuuuge though.
To revert, I tagged DBUS_BEFORE_OBJECT_NAMES_MERGE.
```

2003-09-28 Havoc Pennington <hp@pobox.com>

- * HACKING: update to reflect new server

2003-09-26 Seth Nickell <seth@gnome.org>

- * python/dbus.py:
- * python/examples/example-signals.py:

Start implementing some notions of signals. The API is really terrible, but they sort of work (with the exception of being able to filter by service, and to transmit signals *as* a particular service). Need to figure out how to make messages come from the service we registered :-(

- * python/dbus_bindings.pyx.in:

Removed duplicate message_handler callbacks.

2003-09-25 Havoc Pennington <hp@redhat.com>

- * bus/session.conf.in: fix my mess

2003-09-25 Havoc Pennington <hp@pobox.com>

- * bus/session.conf.in: fix security policy, reported by Seth Nickell

2003-09-25 Seth Nickell <seth@gnome.org>

- * python/examples/example-service.py:

Johan notices complete wrong code in example-service, but completely wrong in a way that works exactly the same (!). Johan is confused, how could this possibly work? Example code fails to serve purpose of making things clear. Seth fixes.

2003-09-25 Mark McLoughlin <mark@skynet.ie>

- * doc/dbus-specification.sgml: don't require header fields to be 4-byte aligned and specify that fields should be distinguished from padding by the fact that zero is not a valid field name.

- * doc/TODO: remove re-alignment item and add item to doc the OBJECT_PATH type.

- * dbus/dbus-message.c:
(HeaderField): rename the original member to value_offset and introduce a name_offset member to keep track of where

the field actually begins.
(adjust_field_offsets): remove.
(append_int_field), (append_uint_field),
(append_string_field): don't align the start of the header field to a 4-byte boundary.
(get_next_field): impl finding the next marshalled field after a given field.
(re_align_field_recurse): impl re-aligning a number of already marshalled fields.
(delete_field): impl deleting a field of any type and re-aligning any following fields.
(delete_int_or_uint_field), (delete_string_field): remove.
(set_int_field), (set_uint_field): no need to re-check that we have the correct type for the field.
(set_string_field): ditto and impl re-aligning any following fields.
(decode_header_data): update to take into account that the fields aren't 4-byte aligned any more and the new way to distinguish padding from header fields. Also, don't exit when there is too much header padding.
(process_test_subdir): print the directory.
(_dbus_message_test): add test to make sure a following field is re-aligned correctly after field deletion.

* dbus/dbus-string.[ch]:
(_dbus_string_insert_bytes): rename from insert_byte and allow the insert of multiple bytes.
(_dbus_string_test): test inserting multiple bytes.

* dbus/dbus-marshal.c: (_dbus_marshal_set_string): add warning note to docs about having to re-align any marshalled values following the string.

* dbus/dbus-message-builder.c:
(append_string_field), (_dbus_message_data_load): don't align the header field.

* dbus/dbus-auth.c: (process_test_subdir): print the directory.

* test/break-loader.c: (randomly_add_one_byte): upd. for insert_byte change.

* test/data/invalid-messages/bad-header-field-alignment.message: new test case.

* test/data/valid-messages/unknown-header-field.message: shove a dict in the unknown field.

2003-09-25 Seth Nickell <seth@gnome.org>

* python/dbus.py:

* python/dbus_bindings.pyx.in:

Handle return values.

* python/examples/example-client.py:

* python/examples/example-service.py:

Pass back return values from the service to the client.

2003-09-24 Seth Nickell <seth@gnome.org>

* python/dbus.py:

pass Connect Object methods (when you are sharing an object) up...

in a list of methods to be shared. Sharing all the methods just worked out too weird. You can now create nice Services over the DBus in Python. :-)

* python/dbus_bindings.pyx.in:

Keep references to user_data tuples passed into C functions so Python doesn't garbage collect on us.

Implement MethodReturn and Error subclasses of Message for creating DBusMessage's of those types.

* python/examples/example-client.py:

* python/examples/example-service.py:

Simple example code showing both how create DBus services and objects, and how to use them.

2003-09-23 Havoc Pennington <hp@pobox.com>

* glib/dbus-gproxy.c (dbus_gproxy_manager_filter): implement

2003-09-23 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c (dbus_gproxy_connect_signal): implement
(dbus_gproxy_disconnect_signal): implement
(dbus_gproxy_manager_remove_signal_match): implement
(dbus_gproxy_manager_add_signal_match): implement
(dbus_gproxy_oneway_call): implement

2003-09-23 Havoc Pennington <hp@pobox.com>

* glib/dbus-gproxy.c (struct DBusGProxy): convert to a GObject subclass. This means dropping the transparent thread safety of the

proxy; you now need a separate proxy per-thread, or your own locking on the proxy. Probably right anyway.
(dbus_gproxy_ref, dbus_gproxy_unref): nuke, just use g_object_ref

2003-09-22 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c (dbus_gproxy_manager_get): implement

2003-09-21 Seth Nickell <seth@gnome.org>

First checkin of the Python bindings.

* python/.cvsignore:
* python/Makefile.am:
* python/dbus_bindings.pyx.in:
* python/dbus_h_wrapper.h:

Pieces for Pyrex to operate on, building a dbus_bindings.so python module for low-level access to the DBus APIs.

* python/dbus.py:

High-level Python module for accessing DBus objects.

* configure.in:
* Makefile.am:

Build stuff for the python bindings.

* acinclude.m4:

Extra macro needed for finding the Python C header files.

2003-09-21 Havoc Pennington <hp@pobox.com>

* glib/dbus-gproxy.c (dbus_gproxy_manager_new): start implementing the proxy manager, didn't get very far.

* dbus/dbus-bus.c (dbus_bus_add_match): new
(dbus_bus_remove_match): new

* glib/dbus-gproxy.c (dbus_gproxy_new_for_service): add a path_name argument; adjust the other not-yet-implemented gproxy constructors to be what I think they should be.

2003-09-21 Havoc Pennington <hp@pobox.com>

* dbus/dbus-bus.c (dbus_bus_get): set exit_on_disconnect to TRUE by default for message bus connections.

* dbus/dbus-connection.c (dbus_connection_dispatch): exit if exit_on_disconnect flag is set and we process the disconnected

signal.
(dbus_connection_set_exit_on_disconnect): new function

2003-09-21 Havoc Pennington <hp@pobox.com>

Get matching rules mostly working in the bus; only actually parsing the rule text remains. However, the client side of "signal connections" hasn't been started, this patch is only the bus side.

* dbus/dispatch.c: fix for the matching rules changes

* bus/driver.c (bus_driver_handle_remove_match)
(bus_driver_handle_add_match): send an ack reply from these method calls

* glib/dbus-gproxy.c (dbus_gproxy_begin_call): fix order of arguments, reported by Seth Nickell

* bus/config-parser.c (append_rule_from_element): support eavesdrop=true|false attribute on policies so match rules can be prevented from snooping on the system bus.

* bus/dbus-daemon-1.1.in: consistently use terminology "sender" and "destination" in attribute names; fix some docs bugs; add eavesdrop=true|false attribute

* bus/driver.c (bus_driver_handle_add_match)
(bus_driver_handle_remove_match): handle AddMatch, RemoveMatch messages

* dbus/dbus-protocol.h (DBUS_SERVICE_ORG_FREEDESKTOP_BROADCAST):
get
rid of broadcast service concept, signals are just always
broadcast

* bus/signals.c, bus/dispatch.c, bus/connection.c, bus/bus.c:
mostly implement matching rules stuff (currently only exposed
as signal
connections)

2003-09-21 Mark McLoughlin <mark@skynet.ie>

* doc/dbus-specification.sgml: Change the header field name to be an enum and update the rest of the spec to reference the fields using the conventional name.

* dbus/dbus-protocol.h: update to reflect the spec.

* doc/TODO: add item to remove the 4 byte alignment requirement.

* dbus/dbus-message.c: Remove the code to generalise the

header/body length and serial number header fields as named header fields so we can reference field names using the protocol values.

(append_int_field), (append_uint_field), (append_string_field): Append the field name as a byte rather than four chars.

(delete_int_or_uint_field), (delete_string_field): reflect the fact that the field name and typecode now occupy 4 bytes instead of 8.

(decode_string_field), (decode_header_data): update to reflect protocol changes and move the field specific encoding from decode_string_field() back into decode_header_data().

* dbus/dbus-internals.[ch]: (_dbus_header_field_to_string): Add utility to aid debugging.

* dbus/dbus-message-builder.c:
(append_string_field), (_dbus_message_data_load): Update to reflect protocol changes; Change the FIELD_NAME directive to HEADER_FIELD and allow it to take the field's conventional name rather than the actual value.

* test/data/*/*.message: Update to use HEADER_FIELD instead of FIELD_NAME; Always align the header on an 8 byte boundary *before* updating the header length.

2003-09-15 Havoc Pennington <hp@pobox.com>

* dbus/dbus-pending-call.c: add the get/set object data boilerplate as for DBusConnection, etc. Use generic object data for the notify callback.

* glib/dbus-gparser.c (parse_node): parse child nodes

* tools/dbus-viewer.c: more hacking on the dbus-viewer

* glib/dbus-gutils.c (_dbus_gutils_split_path): add a file to contain functions shared between the convenience lib and the installed lib

* glib/Makefile.am (libdbus_glib_1_la_LDFLAGS): add -export-symbols-regex to the GLib library

* dbus/dbus-object-tree.c
(_dbus_object_tree_dispatch_and_unlock):
fix the locking in here, and add a default handler for Introspect() that just returns sub-nodes.

2003-09-14 Havoc Pennington <hp@pobox.com>

* glib/dbus-gthread.c (dbus_g_thread_init): rename to make g_foo rather than gfoo consistent

* glib/dbus-gproxy.h: delete for now, move contents to dbus-glib.h, because the include files don't work right since we aren't in the dbus/ subdir.

* glib/dbus-gproxy.c (dbus_gproxy_send): finish implementing
(dbus_gproxy_end_call): finish
(dbus_gproxy_begin_call): finish

* glib/dbus-gmain.c (dbus_set_g_error): new

* glib/dbus-gobject.c (handle_introspect): include information about child nodes in the introspection

* dbus/dbus-connection.c (dbus_connection_list_registered): new function to help in implementation of introspection

* dbus/dbus-object-tree.c
(_dbus_object_tree_list_registered_and_unlock): new function

2003-09-12 Havoc Pennington <hp@pobox.com>

* glib/dbus-gidl.h: add common base class for all the foo_info types

* tools/dbus-viewer.c: add GTK-based introspection UI thingy similar to kdcop

* test/Makefile.am: try test srcdir -ef . in addition to test srcdir = ., one of them should work (yeah lame)

* glib/Makefile.am: build the "idl" parser stuff as a convenience library

* glib/dbus-gparser.h: make description_load routines return NodeInfo* not Parser*

* Makefile.am (SUBDIRS): build test dir after all library dirs

* configure.in: add GTK+ detection

2003-09-07 Havoc Pennington <hp@pobox.com>

* Make Doxygen contented.

2003-09-07 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: more updates

2003-09-06 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: partial updates

* bus/dbus-daemon-1.1.in: fix the config file docs for the zillionth time; hopefully I edited the right file this time.

* bus/config-parser.c (append_rule_from_element): support send_type, send_path, receive_type, receive_path

* bus/policy.c: add message type and path to the list of things that can be "firewalled"

2003-09-06 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_register_fallback): add this
(dbus_connection_register_object_path): make this not handle messages to paths below the given path

2003-09-03 Havoc Pennington <hp@pobox.com>

* test/glib/Makefile.am: add this with random glib-linked test programs

* glib/Makefile.am: remove the random test programs from here, leave only the unit tests

* glib/dbus-gobject.c (_dbus_gobject_test): add test for uscore/javacaps conversion, and fix
(get_object_property, set_object_property): change to .NET convention for mapping props to methods, set_FooBar/get_FooBar, since one language has such a convention we may as well copy it. Plus real methods in either getFooBar or get_foo_bar style won't collide with this convention.

2003-09-01 Havoc Pennington <hp@pobox.com>

* glib/dbus-gparser.c: implement

* glib/dbus-gobject.c: start implementing skeletons support

* configure.in: when disabling checks/assert, also define G_DISABLE_ASSERT and G_DISABLE_CHECKS

2003-09-01 Havoc Pennington <hp@pobox.com>

* glib/Makefile.am: rearrange a bunch of files and get "make check" framework set up

2003-08-31 Havoc Pennington <hp@pobox.com>

* fix build with --disable-tests

2003-08-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c: purge DbusMessageHandler

* dbus/dbus-message-handler.c: remove DbusMessageHandler, just use callbacks everywhere

2003-08-30 Havoc Pennington <hp@pobox.com>

* test/data/valid-config-files/system.d/test.conf: change to root for the user so warnings don't get printed

* dbus/dbus-message.c: add dbus_message_get_path, dbus_message_set_path

* dbus/dbus-object-tree.c (do_test_dispatch): add test of dispatching to a path

* dbus/dbus-string.c (_dbus_string_validate_path): add

* dbus/dbus-marshal.c (_dbus_demarshal_object_path): implement (_dbus_marshall_object_path): implement

* dbus/dbus-protocol.h (DBUS_HEADER_FIELD_PATH): new header field to contain the path to the target object
(DBUS_HEADER_FIELD_SENDER_SERVICE): rename DBUS_HEADER_FIELD_SENDER to explicitly say it's the sender service

2003-08-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-tree.c: write tests and fix the discovered bugs

2003-08-29 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-tree.c: modify to allow overlapping paths to be

registered
(struct DbusObjectSubtree): shrink this a lot, since we may have a lot of them
(_dbus_object_tree_free_all_unlocked): implement
(_dbus_object_tree_dispatch_and_unlock): implement

2003-08-29 Havoc Pennington <hp@pobox.com>

* dbus/dbus-internals.h: fix _DBUS_N_GLOBAL_LOCKS

2003-08-28 Havoc Pennington <hp@pobox.com>

purge DbusObjectID

* dbus/dbus-connection.c: port to no ObjectID, create a

DBusObjectTree, rename ObjectTree to ObjectPath in public API

* dbus/dbus-connection.h (struct DBusObjectTreeVTable): delete everything except UnregisterFunction and MessageFunction

* dbus/dbus-marshal.c: port away from DBusObjectID, add DBUS_TYPE_OBJECT_PATH

* dbus/dbus-object-registry.[hc], dbus/dbus-object.[hc], dbus/dbus-objectid.[hc]: remove these, we are moving to path-based object IDs

2003-08-25 Havoc Pennington <hp@pobox.com>

Just noticed that dbus_message_test is hosed, I wonder when I broke that. I thought make check was passing earlier...

* dbus/dbus-object-tree.c: add new "object tree" to match DCOP container tree, will replace most of dbus-object-registry

* dbus/dbus-string.c (_dbus_string_append_printf_valist): fix C99 screwup

2003-08-19 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (decode_string_field): support FIELD_SENDER (dbus_message_is_error): fix this function

* bus/dbus-daemon-1.1: clarify logic on when <deny>/<allow> rules match

* bus/policy.c (bus_client_policy_check_can_receive): fix code to reflect clarified man page (bus_client_policy_check_can_send): ditto

* bus/session.conf.in: fixup

* bus/system.conf.in: fixup

2003-08-18 Havoc Pennington <hp@redhat.com>

* dbus/dbus-hash.c (_dbus_hash_table_insert_two_strings): fix

...

```
glib/.cvsignore      | 6 +-
glib/Makefile.am     | 70 +-
glib/dbus-gidl.c     | 524 +++++
glib/dbus-gidl.h     | 120 +++++
glib/dbus-glib-tool.c | 77 +++
glib/dbus-glib.h     | 121 +++++
glib/dbus-gloader-  
expat.c             | 262 +++++
```

```

glib/dbus-gmain.c      | 100 +++-
glib/dbus-gobject.c   | 790 ++++++
glib/dbus-gparser.c   | 670 ++++++
glib/dbus-gparser.h   | 65 +++
glib/dbus-gproxy.c    | 1249
+++++
glib/dbus-gtest-main.c | 46 ++
glib/dbus-gtest.c     | 77 +++
glib/dbus-gtest.h     | 35 ++
glib/dbus-gthread.c   | 2 +-
glib/dbus-gtool-test.h | 31 ++
glib/dbus-gutils.c    | 96 ++++
glib/dbus-gutils.h    | 40 ++
19 files changed, 4345 insertions(+), 36 deletions(-)

```

```

commit c6510f49417e04522d9cc6a6c51331bef252aba0
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-08-05 13:57:20 +0000

```

2003-08-05 Anders Carlsson <andersca@codefactory.se>

```

* glib/dbus-gmain.c: (watch_fd_new), (watch_fd_ref),
  (watch_fd_unref), (dbus_gsource_check), (dbus_gsource_dispatch),
  (add_watch), (remove_watch), (create_source):
  Refcount fds, fixes some reentrancy issues.

```

```

glib/dbus-gmain.c | 116
+++++-----
1 file changed, 78 insertions(+), 38 deletions(-)

```

```

commit a3c695d62bcfc46bd12f4165610649a604ed6166
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-06-22 19:39:46 +0000

```

2003-06-22 Havoc Pennington <hp@pobox.com>

```

* dbus/dbus-dataslot.c (_dbus_data_slot_allocator_unref)
  (_dbus_data_slot_allocator_alloc): rework these to keep a
  reference count on each slot and automatically manage a global
  slot ID variable passed in by address

* bus/bus.c: convert to new dataslot API

* dbus/dbus-bus.c: convert to new dataslot API

* dbus/dbus-connection.c: convert to new dataslot API

* dbus/dbus-server.c: convert to new dataslot API

* glib/dbus-gmain.c: ditto

* bus/test.c: ditto

```

* bus/connection.c: ditto

```
glib/dbus-gmain.c      | 24 ++++++++-----  
glib/test-thread-server.c | 5 +++--  
2 files changed, 13 insertions(+), 16 deletions(-)
```

commit 1284fea96df990975632f0a5d8d982ec5c974161
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-06-19 22:19:56 +0000

2003-06-19 Anders Carlsson <andersca@codefactory.se>

* glib/dbus-glib.h: Fix so that dbus-glib.h can be used
from C++ (Patch by Miloslav Trmac).

```
glib/dbus-glib.h | 4 +++  
1 file changed, 4 insertions(+)
```

commit 375ff8ffbed8683c1a46fab9c537ccb9e603e1d7
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-06-18 10:48:07 +0000

Add .cvsignore files

```
glib/.cvsignore | 1 +  
1 file changed, 1 insertion(+)
```

commit b4dd1b2cb3e226ece5055bc2f24ef819db7d1b08
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-05-17 17:53:16 +0000

2003-05-17 Havoc Pennington <hp@pobox.com>

* bus/config-parser.c (merge_included): merge in policies from
child configuration file.

* bus/policy.c (bus_policy_merge): function to merge two policies
together

```
glib/test-profile.c | 4 +++-  
1 file changed, 2 insertions(+), 2 deletions(-)
```

commit ae655688473fc5e5d6a279b3e764a713fc73b8ca
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-05-12 02:44:44 +0000

2003-05-11 Havoc Pennington <hp@pobox.com>

* dbus/dbus-marshal.c (_dbus_marshal_validate_arg): fix to avoid
calling _dbus_marshal_validate_arg() for every byte in a byte
array, etc.

```

* dbus/dbus-message-handler.c: use atomic reference counting to
reduce number of locks slightly; the global lock in here sucks

* dbus/dbus-connection.c
(_dbus_connection_update_dispatch_status_and_unlock): variant of
update_dispatch_status that can be called with lock held; then
use
in a couple places to reduce locking/unlocking
(dbus_connection_send): hold the lock over the whole function
instead of acquiring it twice.

* dbus/dbus-timeout.c (_dbus_timeout_new): handle OOM

* bus/connection.c (bus_connections_setup_connection): fix access
to already-freed memory.

* dbus/dbus-connection.c: keep a little cache of linked list
nodes, to avoid using the global linked list alloc lock in the
normal send-message case. Instead we just use the connection lock
that we already have to take.

* dbus/dbus-list.c (_dbus_list_find_last): new function

* dbus/dbus-sysdeps.c (_dbus_atomic_inc, _dbus_atomic_dec):
change to use a struct for the atomic type; fix docs,
they return value before increment, not after increment.

* dbus/dbus-string.c (_dbus_string_append_4_aligned)
(_dbus_string_append_8_aligned): new functions to try to
microoptimize this operation.
(reallocate_for_length): break this out of set_length(), to
improve profile info, and also so we can consider inlining the
set_length() part.

* dbus/dbus-message.c (dbus_message_new_empty_header): init data
strings with some preallocation, cuts down on our calls to
realloc
a fair bit. Though if we can get the "move entire string to empty
string" optimization below to kick in here, it would be better.

* dbus/dbus-string.c (_dbus_string_move): just call
_dbus_string_move_len
(_dbus_string_move_len): add a special case for moving
an entire string into an empty string; we can just
swap the string data instead of doing any reallocs.
(_dbus_string_init_preallocated): new function

glib/test-profile.c |          16 ++++++++-----
1 file changed, 12 insertions(+), 4 deletions(-)

```

```
commit b36dd8b35d932d01407b88e948c5196b49e31aa0
```

Author: Havoc Pennington <hp@redhat.com>
Date: 2003-05-11 07:59:08 +0000

2003-05-11 Havoc Pennington <hp@pobox.com>

Write a "test-profile" that does echo client-server with threads; profile reveals lock contention, memcpy/realloc of buffers, and UTF-8 validation as hot spots. 20% of lock contention eliminated with dbus_atomic_inc/dec implementation on x86. Much remaining contention is global mempool locks for GList and DBusList.

* dbus/dbus-sysdeps.c (_dbus_atomic_inc, _dbus_atomic_dec): add x86 implementation

* dbus/dbus-connection.c (struct DBusConnection): use dbus_atomic_t for the reference count

* dbus/dbus-message.c (struct DBusMessage): declare dbus_atomic_t values as volatile

* configure.in: code to detect ability to use atomic integer operations in assembly, from GLib patch

* dbus/dbus-internals.c (_dbus_verbose_real): call getpid every time, tired of it being wrong in threads and forked processes

* glib/test-profile.c: a little program to bounce messages back and forth between threads and eat CPU

* dbus/dbus-connection.c: add debug spew macros for debugging thread locks; include config.h at top; fix deadlock in dbus_connection_flush()

```
glib/Makefile.am      |          33 +++++---
glib/test-profile.c  |           215
+++++
2 files changed, 237 insertions(+), 11 deletions(-)
```

commit 6f4b867ddd6cef070d65149a4d935b4a964987f1
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-04-29 22:57:13 +0000

2003-04-29 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c: docs cleanups

* dbus/dbus-types.h: add docs on int64 types

* dbus/dbus-memory.c: fix docs to avoid putting private API in public API docs section

```
glib/dbus-gmain.c | 8 ++++++-
```

1 file changed, 7 insertions(+), 1 deletion(-)

commit 7d22e9a86a21e337a31ee42ccdb851ada88c0b40
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-04-25 20:41:49 +0000

2003-04-25 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gmain.c (remove_watch): fix for a crash when watches were toggled without add/remove, fix from Anders Gustafsson

glib/dbus-gmain.c | 4 ++++
1 file changed, 4 insertions(+)

commit 36a6a6bc4edb1b3280e405b0addb9113dc09de5c
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-04-24 19:18:22 +0000

2003-04-24 Havoc Pennington <hp@redhat.com>

- * configure.in: add --enable-checks
- * dbus/dbus-message.c (dbus_message_new): reverse name/service arguments
- * dbus/dbus-connection.c (dbus_connection_preallocate_send): fix to use thread locks.
(dbus_connection_handler_destroyed_locked): move some private functions into proper docs group
- * dbus/dbus-internals.h: add _dbus_return_if_fail, _dbus_return_val_if_fail

Throughout: use dbus_return_if_fail

glib/test-dbus-glib.c | 3 ++-
glib/test-thread-client.c | 2 +-
2 files changed, 3 insertions(+), 2 deletions(-)

commit eb553e370e7651a3104038cbba7ebb7a502118e7
Author: James Willcox <jwillcox@gnome.org>
Date: 2003-04-24 02:22:49 +0000

2003-04-23 James Willcox <jwillcox@gnome.org>

- * glib/dbus-glib.h:
- * glib/dbus-gmain.c: (add_timeout), (wakeup_main), (create_source), (dbus_connection_setup_with_g_main), (dbus_server_setup_with_g_main):
- * glib/test-dbus-glib.c: (main):
- * glib/test-thread-client.c: (main):


```
* glib/test-thread-server.c: (new_connection_callback), (main):
* tools/dbus-monitor.c: (main):
```

```
Added a GMainContext argument to
dbus_connection_setup_with_g_main()
and dbus_server_setup_with_g_main().
```

```
glib/dbus-glib.h          | 6 +++++-
glib/dbus-gmain.c         | 36 ++++++-----
glib/test-dbus-glib.c     | 2 +-
glib/test-thread-client.c | 2 +-
glib/test-thread-server.c | 4 ++--
5 files changed, 31 insertions(+), 19 deletions(-)
```

```
commit 62cc9827deeld9fa052a11ce259acc7d1d2dcb9d
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-04-18 04:18:57 +0000
```

```
2003-04-18 Havoc Pennington <hp@pobox.com>
```

```
* glib/dbus-gmain.c: adapt to watch changes

* bus/bus.c, bus/activation.c, etc.: adjust to watch changes

* dbus/dbus-server.h: remove dbus_server_handle_watch

* dbus/dbus-connection.h: remove dbus_connection_handle_watch

* dbus/dbus-watch.c (dbus_watch_handle): change DBusWatch to work
like DBusTimeout, so we don't need dbus_connection_handle_watch
etc.
```

```
glib/dbus-gmain.c | 9 ++-----
1 file changed, 2 insertions(+), 7 deletions(-)
```

```
commit 723f9251ee8892f05f4c94a9b0cfa864c5e56bae
Author: Alex Larsson <alexl@redhat.com>
Date: 2003-04-10 12:09:00 +0000
```

```
2003-04-10 Alexander Larsson <alexl@redhat.com>
```

```
* bus/.cvsignore:
* glib/.cvsignore:
* test/.cvsignore:
Added files to cvsignore

* dbus/dbus-message.h:
* dbus/dbus-message.c: (dbus_message_iter_get_named):
Make get_named() take two out argument and return a boolean.
(dbus_message_iter_get_args_valist):
Update usage of get_named().
(dbus_message_iter_append_byte):
```

```
Fix typo
(dbus_message_iter_append_named)
Fix typo
(message_iter_test), (check_message_handling_type),
(_dbus_message_test):
More tests.
```

```
glib/.cvsignore |    2 ++
1 file changed, 2 insertions(+)
```

```
commit f8c607ec8949388bd09136d0a6635190910a3e9a
Author: Alex Larsson <alexl@redhat.com>
Date: 2003-04-08 15:52:50 +0000
```

```
2003-04-08 Alexander Larsson <alexl@redhat.com>
```

```
Implemented recursive types, named types and new-style iters
```

```
* bus/driver.c:
* glib/test-thread-client.c: (thread_func):
* glib/test-thread-server.c: (handle_test_message):
* test/test-service.c: (handle_echo):
Update to new api
```

```
* dbus/Makefile.am:
* dbus/dbus-dict.c:
* dbus/dbus-dict.h:
* dbus/dbus.h
Remove DbusDict
```

```
* dbus/dbus-internals.c: (_dbus_type_to_string):
Update for new types.
```

```
* dbus/dbus-marshal.[ch]:
Implement recursive types and the new marshalling format.
Remove hardcoded dict marshalling.
Marshal named types.
```

```
* dbus/dbus-message-builder.c:
Add BYTE_ARRAY.
Remove references to old types
```

```
* dbus/dbus-message.[ch]:
New non-refcounted iter API that supports recursive iters.
Use iters for appending, including support for recursive
iters.
Add byte and named type support.
Update everything to new marshalling formats.
Add tests for new API.
```

```
* dbus/dbus-protocol.h:
Remove old array types.
```

Add types: BYTE, ARRAY, DICT, NAMED

* dbus/dbus-string.c:

* dbus/dbus-sysdeps.c:

Make parse_double locale safe.

* dbus/dbus-test-main.c:

Call setlocale.

* dbus/dbus-test.c:

Kill dict test

* doc/dbus-specification.sgml:

Update spec

* test/data/incomplete-messages/missing-body.message:

* test/data/invalid-messages/bad-boolean.message:

* test/data/invalid-messages/bad-boolean-array.message:

*

test/data/invalid-messages/boolean-array-length-too-long.message-

raw:

* test/data/invalid-messages/boolean-has-no-value.message-raw:

* test/data/invalid-messages/too-short-dict.message:

* test/data/valid-messages/dict-simple.message:

* test/data/valid-messages/dict.message:

* test/data/valid-messages/emptiness.message:

* test/data/valid-messages/lots-of-arguments.message:

* test/data/valid-messages/no-padding.message:

* test/data/valid-messages/recursive-types.message:

Add missing NAME fields

Fix up dicts & arrays

* test/data/invalid-messages/dict-with-nil-value.message:

Removed, this is not invalid anymore.

* test/data/valid-messages/recursive-types.message:

Add new test for deeply recursive types.

glib/test-thread-client.c | 9 ++++++---

glib/test-thread-server.c | 23 ++++++++-----

2 files changed, 17 insertions(+), 15 deletions(-)

commit 4612ff9267417cf8860c233a59fe8ec249ee4a62

Author: Havoc Pennington <hp@redhat.com>

Date: 2003-03-25 04:37:07 +0000

2003-03-24 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_set_fd_nonblocking): move to this file

* dbus/dbus-errors.c (dbus_set_error, dbus_set_error_const):
allow

NULL argument for "message" if the error is a well-known one,
fill in a generic message in this case.

* dbus/dbus-errors.h (DBusResultCode): Kill DBusResultCode in
favor of DBusError

* bus/test.c (bus_test_flush_bus): add

* bus/policy.c (bus_policy_test): test code stub

```
glib/test-dbus-glib.c      | 7 ++++---  
glib/test-thread-client.c | 8 ++++---  
glib/test-thread-server.c | 8 ++++---  
3 files changed, 14 insertions(+), 9 deletions(-)
```

commit 9eb162678713477754e9d88eb68c156d336c80ab

Author: Havoc Pennington <hp@redhat.com>

Date: 2003-03-16 20:16:46 +0000

2003-03-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-string.c (_dbus_string_validate_utf8): oops, unbreak
this. always run the test suite before commit...

* bus/*: adapt to DBusConnection API changes

* glib/dbus-gmain.c: adapt to DBusConnection API changes,
requires renaming stuff to avoid dbus_connection_dispatch name
conflict.

* dbus/dbus-transport.c (_dbus_transport_queue_messages): new
function

* dbus/dbus-message.c (_dbus_message_loader_queue_messages):
separate from _dbus_message_loader_return_buffer()

* dbus/dbus-connection.c (dbus_connection_get_n_messages): remove
this, because it's now always broken to use; the number of
messages in queue vs. the number still buffered by the message
loader is undefined/meaningless. Should use
dbus_connection_get_dispatch_state().
(dbus_connection_dispatch): rename from
dbus_connection_dispatch_message

```
glib/dbus-gmain.c | 64  
+++++-----  
1 file changed, 32 insertions(+), 32 deletions(-)
```

commit 457dd574d5eda1286571f7070860c6feb56f184e

Author: Havoc Pennington <hp@redhat.com>

Date: 2003-03-15 20:47:16 +0000

2003-03-15 Havoc Pennington <hp@pobox.com>

Make it pass the Hello handling test including all OOM codepaths.
Now to do other messages...

- * bus/services.c (bus_service_remove_owner): fix crash when removing owner from an empty list of owners
(bus_registry_ensure): don't leave service in the list of a connection's owned services if we fail to put the service in the hash table.
- * bus/connection.c (bus_connection_preallocate_oom_error): set error flag on the OOM error.
- * dbus/dbus-connection.c (_dbus_connection_new_for_transport): handle _dbus_transport_set_connection failure
- * dbus/dbus-transport-unix.c (_dbus_transport_new_for_fd): modify to create watches up front and simply enable/disable them as needed.
(unix_connection_set): this can now fail on OOM
- * dbus/dbus-timeout.c, dbus/dbus-watch.c: add concept of enabling/disabling a watch or timeout.
- * bus/loop.c (bus_loop_iterate): don't touch disabled watches/timeouts
- * glib/dbus-gmain.c: adapt to enable/disable watches and timeouts

glib/dbus-gmain.c | 44 ++++++-----
1 file changed, 42 insertions(+), 2 deletions(-)

commit 0f748f23b6fd40ab314bdc4d18e631cde958c1d2

Author: Havoc Pennington <hp@redhat.com>

Date: 2003-03-15 02:19:01 +0000

2003-03-14 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-memory.c: add a "detect buffer overwrites on free" cheesy hack
- * dbus/dbus-transport-debug.c: rework this a good bit to be less complicated. hopefully still works.
- * dbus/dbus-server-debug.c (handle_new_client): remove timeout manually
- * glib/dbus-gmain.c (timeout_handler): don't remove timeout after running it

* dbus/dbus-message.c (dbus_message_copy): rename from
dbus_message_new_from_message, fix it up to copy
all the message fields, add test case

* bus/dispatch.c (bus_dispatch_test): add some more test code,
not quite passing yet

glib/dbus-gmain.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit a15287bc97819cca7b61cc18d6d69232bb05117d
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-03-14 01:27:58 +0000

2003-03-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-timeout.c (_dbus_timeout_list_set_functions): handle
out of memory

* dbus/dbus-watch.c (_dbus_watch_list_set_functions): handle out
of memory

* dbus/dbus-connection.h: Make AddWatchFunction and
AddTimeoutFunction return a bool so they can fail on out-of-
memory

* bus/bus.c (bus_context_new): set up timeout handlers

* bus/connection.c (bus_connections_setup_connection): set up
timeout handlers

* glib/dbus-gmain.c: adapt to the fact that set_functions stuff
can fail

* bus/bus.c (bus_context_new): adapt to changes

* bus/connection.c: adapt to changes

* test/watch.c: adapt to DBusWatch changes

* bus/dispatch.c (bus_dispatch_test): started adding this but
didn't finish

glib/dbus-gmain.c | 28 ++++++-----
1 file changed, 17 insertions(+), 11 deletions(-)

commit e0a3fe2914f611ef4d74bbc1ca39794f0f7ae575
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-03-13 00:56:43 +0000

2003-03-12 Havoc Pennington <hp@redhat.com>

Mega-patch that gets the message bus daemon initially handling out-of-memory. Work still needed. Also lots of random moving stuff to DBusError instead of ResultCode.

- * dbus/dbus-list.c (`_dbus_list_length_is_one`): new function
- * dbus/dbus-connection.c (`dbus_connection_send_with_reply_and_block`): use DBusError
- * dbus/dbus-bus.c: adapt to API changes, make it use DBusError

not DBusResultCode

- * dbus/dbus-connection.c (`dbus_connection_send`): drop the result code here, as the only failure possible is OOM.
- * bus/connection.c (`bus_connection_disconnect`): rename `bus_connection_disconnected` as it's a notification only
- * bus/driver.c (`bus_driver_handle_acquire_service`): don't free "name" on `get_args` failure, should be done by `get_args`; don't disconnect client for bad args, just return an error. (`bus_driver_handle_service_exists`): ditto
- * bus/services.c (`bus_services_list`): NULL-terminate returned array
- * bus/driver.c (`bus_driver_send_service_lost`) (`bus_driver_send_service_acquired`): send messages from driver to specific client to the client's unique name, not to the broadcast service.
- * dbus/dbus-message.c (`decode_header_data`): reject messages that contain no name field
(`_dbus_message_get_client_serial`): rename to `dbus_message_get_serial` and make public
(`_dbus_message_set_serial`): rename from `set_client_serial`
(`_dbus_message_set_reply_serial`): make public
(`_dbus_message_get_reply_serial`): make public
- * bus/connection.c (`bus_connection_foreach`): allow stopping iteration by returning `FALSE` from `foreach` function.
- * dbus/dbus-connection.c (`dbus_connection_send_preallocated`) (`dbus_connection_free_preallocated_send`) (`dbus_connection_preallocate_send`): new API for sending a message without possibility of malloc failure.
(`dbus_connection_send_message`): rename to just `dbus_connection_send` (and same for whole function family)

* dbus/dbus-errors.c (dbus_error_free): make this reinit the error

* dbus/dbus-sysdeps.c (_dbus_exit): new function

* bus/activation.c: handle/return errors

* dbus/dbus-errors.h: add more DBUS_ERROR #define

* dbus/dbus-sysdeps.c (_dbus_directory_open)
(_dbus_file_get_contents)
(_dbus_directory_get_next_file): use DBusError instead of
DBusResultCode
(_dbus_result_from_errno): move to this file

```
glib/test-dbus-glib.c      | 20 ++++++++-----  
glib/test-thread-client.c |  9 +++++---  
2 files changed, 19 insertions(+), 10 deletions(-)
```

commit 284daff1f103a303178243ca3e2dee960651fec0
Author: Alex Larsson <alexl@redhat.com>
Date: 2003-02-27 14:22:36 +0000

2003-02-27 Alexander Larsson <alexl@redhat.com>

* glib/Makefile.am:
* configure.in:
Make gthreads-2.0 dependency optional. Don't build thread test if
its not found.

```
glib/Makefile.am |  8 ++++++-  
1 file changed, 7 insertions(+), 1 deletion(-)
```

commit b9589e9908d368cb064d1614e5c52239394414f4
Author: Alex Larsson <alexl@redhat.com>
Date: 2003-02-26 15:52:25 +0000

2003-02-26 Alexander Larsson <alexl@redhat.com>

* configure.in:
Set DBUS_GLIB_THREADS_LIBS for apps using gthread-2.0

* dbus/dbus-connection.c:
* dbus/dbus-connection.h:
Fix _dbus_connection_acquire_io_path and
_dbus_connection_acquire_dispatch.
Add dbus_connection_set_wakeup_main_function and use it when
queueing
incoming and outgoing messages.

* dbus/dbus-dataslot.c:

Threadsafe usage of DBusDataSlotAllocator

* dbus/dbus-message.c: (dbus_message_get_args_iter):
dbus_new can fail.

* dbus/dbus-server-unix.c:
Add todo comment

* glib/dbus-gmain.c:
Implement the new wakeup functions for glib.

* glib/Makefile.am:

* glib/test-thread-client.c:

* glib/test-thread-server.c:

* glib/test-thread.h:

Initial cut at some thread test code. Not really done yet.

```
glib/Makefile.am      | 16 +-  
glib/dbus-gmain.c    | 13 +-  
glib/test-thread-client.c | 89 ++++++  
glib/test-thread-server.c | 246  
+++++  
glib/test-thread.h    | 1 +  
5 files changed, 362 insertions(+), 3 deletions(-)
```

commit c79213c6d3b041ce00bb00f24d4824e4a77f63ef
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-02-26 06:42:57 +0000

2003-02-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c
(dbus_connection_send_message_with_reply_and_block): fix crash
where we ref'd the outgoing message instead of the returned reply

* dbus/dbus-transport-unix.c (do_authentication): check read
watch
at the end of this function, so if we didn't need to read for
authentication, we reinstall it for receiving messages

* dbus/dbus-message.c (dbus_message_new_reply): allow replies to
a NULL sender for peer-to-peer case

* dbus/dbus-transport-unix.c (check_read_watch): handle
!authenticated case correctly

* glib/dbus-gmain.c: add support for DBusServer

* dbus/dbus-server.c: add data slot support

* glib/dbus-gmain.c (dbus_connection_setup_with_g_main): check
return values and handle errors

* dbus/dbus-dataslot.c: factor out the data slot stuff from
DBusConnection

* Doxyfile.in (INPUT): add glib subdir

* glib/dbus-gmain.c (dbus_connection_setup_with_g_main): rename
setup_with_g_main instead of hookup_with_g_main; write docs

```
glib/dbus-glib.h      |    5 +-  
glib/dbus-gmain.c    |   257  
+++++-----  
glib/dbus-gthread.c  |   16 +++  
glib/test-dbus-glib.c |    2 +-  
4 files changed, 238 insertions(+), 42 deletions(-)
```

commit ebase7f90ea6b3d85a9140be3671ea527c3a5727

Author: Joe Shaw <joe@novell.com>

Date: 2003-02-18 22:51:35 +0000

2003-02-18 Joe Shaw <joe@assbarn.com>

* dbus/dbus-auth.c (handle_server_data_stupid_test_mech): Just
get
credentials from our currently running process.
(get_word): Fix a buglet where we were copying the entire length
instead of relative to our position.

* dbus/dbus-hash.c (_dbus_hash_test): Don't try to allocate the
keys on the stack... it's 640k of data.

* dbus/dbus-sysdeps.c (_dbus_read_credentials_unix_socket):
Always
read the credentials byte off the socket, even if we don't have
SO_PEERCREC.
(_dbus_poll): Implement poll() using select() for systems which
don't have it.

* glib/test-dbus-glib.c (main): Print out an error if no
parameters are given.

* test/data/auth/fallback.auth-script: Added. Tests that a
client
can fallback to a secondary auth mechanism if the first fails.

```
glib/test-dbus-glib.c |    7 ++++++  
1 file changed, 7 insertions(+)
```

commit 2655a78cdf1114a686d3c2ee28025db133b287c3

Author: Anders Carlsson <andersca@gnome.org>

Date: 2003-02-16 12:57:26 +0000

2003-02-16 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-auth.c: (client_try_next_mechanism):
Plug a leak.
- * dbus/dbus-threads.c: (dbus_condvar_wait_timeout):
Return TRUE if there's no thread implementation around.
- * glib/dbus-gmain.c: (free_source),
(dbus_connection_hookup_with_g_main):
Make sure to remove the GSource when the connection is finalized.

glib/dbus-gmain.c | 15 ++++++
1 file changed, 15 insertions(+)

commit 4fbaec0ff143d92f96fb60c585282b5cb03bdc76
Author: Havoc Pennington <hp@redhat.com>
Date: 2003-02-16 07:20:54 +0000

2003-02-16 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-connection.c (dbus_connection_set_change_sigpipe):
allow people to avoid setting SIGPIPE to SIG_IGN
(_dbus_connection_new_for_transport): disable SIGPIPE unless
we've been asked not to

glib/.cvsignore | 4 +++
1 file changed, 4 insertions(+)

commit 69a436d7913c1681eea397e768687fab736f9d7c
Author: Alex Larsson <alexl@redhat.com>
Date: 2003-02-15 16:25:08 +0000

2003-02-15 Alexander Larsson <alexl@redhat.com>

- * dbus/dbus-threads.c:
- * dbus/dbus-threads.h:
Add condvars. Remove static mutext from API.
Implement static mutexes by initializing them from threads_init.
- * glib/dbus-gthread.c:
- * qt/dbus-qthread.cpp:
Update with the thread api changes.
- * dbus/dbus-list.c:
- * dbus/dbus-list.h:
Turn StaticMutex into normal mutex + init function.
Export new functions _dbus_list_alloc_link, _dbus_list_free_link,
_dbus_list_append_link, _dbus_list_prepend_link
- * dbus/dbus-sysdeps.c:
- * dbus/dbus-sysdeps.h:

New type `dbus_atomic_t`, and new functions `_dbus_atomic_inc`, `_dbus_atomic_dec`. Only slow fallback implementation at the moment.

* `dbus/dbus-protocol.h`:
Add `DBUS_MESSAGE_LOCAL_DISCONNECT` define

* `dbus/dbus-message.c`:
Make `ref/unref` atomic.
Fix some docs.

* `dbus/dbus-connection-internal.h`:
* `dbus/dbus-connection.c`:
* `dbus/dbus-connection.h`:
Make threadsafe.
Change `_peek` to `_borrow`, `_return` & `_steal_borrowed`.
Change `disconnect` callback to event.
Make `dbus_connection_dispatch_messages` reentrant.

* `dbus/dbus-transport.c`:
Don't ref the connection on calls to the transport implementation.

* `dbus/dbus-message-handler.c`:
Make threadsafe.

* `glib/dbus-gmain.c`:
Don't use `peek_message` anymore

* `test/Makefile.am`:
* `test/debug-thread.c`:
* `test/debug-thread.h`:
Simple thread implementation that asserts() on deadlocks in single-threaded code.

* `test/bus-test.c`:
(main) Call `debug_threads_init`.

* `test/watch.c`:
Use `disconnect` message instead of `disconnect` callback.

* `bus/connection.c`:
* `bus/connection.h`:
Don't call `dbus_connection_set_disconnect_function`. Instead
export
`bus_connection_disconnect`.

* `bus/dispatch.c`:
Call `bus_connection_disconnect` when we get a disconnected message.

```
glib/dbus-gthread.c |          94
+++++-----
2 files changed, 86 insertions(+), 10 deletions(-)
```

```
commit 530cf7689515707b5430b3207da225320cf244c4
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-02-13 21:37:58 +0000
```

```
2003-02-13 Anders Carlsson <andersca@codefactory.se>
```

```
* bus/driver.c: (bus_driver_handle_hello):
* bus/driver.h:
* bus/services.c: (bus_service_lookup):
Reorder message sending so we get a more sane order.

* test/bus-test.c: (message_handler):
Fix tyop.
```

```
glib/test-dbus-glib.c |      3 ---
1 file changed, 3 deletions(-)
```

```
commit 044f4c6ed2170170d833d402079b90277893cd05
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-02-13 20:37:07 +0000
```

```
2003-02-13 Anders Carlsson <andersca@codefactory.se>
```

```
* glib/dbus-gmain.c: (timeout_handler), (add_timeout),
(remove_timeout):
Implement support for timeouts in dbus-glib.
```

```
glib/dbus-gmain.c |    22 ++++++
1 file changed, 22 insertions(+)
```

```
commit 12d8e0d2d6aeb5f29ff8b51cdd4aae857d35d13f
Author: Alex Larsson <alex1@redhat.com>
Date: 2003-02-01 22:02:27 +0000
```

```
2003-02-02 Alexander Larsson <alex1@redhat.com>
```

```
* dbus/dbus-watch.c (dbus_watch_get_flags):
Add note in the docs that ERROR or HANGUP won't be returned
and are assumed always on.

* glib/dbus-gmain.c (add_watch):
Always add IO_ERR | IO_HUP

* dbus/dbus-message.h:
Add semicolon after dbus_message_iter_get_string_array().
Makes qt code build again
```

```
glib/dbus-gmain.c |    5 +----
```

1 file changed, 1 insertion(+), 4 deletions(-)

commit bb581faf87398e807a828f2686474869cd23757d
Author: Richard Hult <rhult@codefactory.de>
Date: 2003-01-26 00:06:30 +0000

2003-01-26 Richard Hult <rhult@codefactory.se>

* glib/dbus-gmain.c (dbus_connection_dispatch): Traverse a copy
of
the file descriptor list, since it can change under us.

glib/dbus-gmain.c | 12 ++++++----
1 file changed, 8 insertions(+), 4 deletions(-)

commit 74cef08b5e2f2596cade4fc1027933d763cd13ae
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-01-25 21:22:57 +0000

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* glib/dbus-gmain.c: (dbus_connection_prepare),
(dbus_connection_check), (dbus_connection_dispatch), (add_watch),
(remove_watch), (dbus_connection_hookup_with_g_main):
Rewrite the glib handling to use its own GSource instead of a
GIOChannel so we can catch messages put in the queue while
waiting
for a reply.

glib/dbus-gmain.c | 193
+++++-----
1 file changed, 128 insertions(+), 65 deletions(-)

commit 6f062c93ca19a73b34688488e76ec055d2bab2a0
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-01-24 23:51:58 +0000

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* bus/connection.c: (bus_connection_foreach):
* bus/connection.h:
Add new bus_connection_foreach function.

* bus/driver.c: (send_one_message),
(bus_driver_broadcast_message):
Add function that broadcasts a message to all clients.

(bus_driver_send_service_created), (bus_driver_handle_hello),
(bus_driver_send_welcome_message),
(bus_driver_handle_list_services), (bus_driver_message_handler):
Implement functions that take care of listing services, and
notifying

clients when new services are created.

```
* bus/services.c: (bus_services_list):
* bus/services.h:
Add new function that returns an array of strings with the
currently
registered services.

* glib/dbus-glib.h:
* glib/dbus-gmain.c:
Update copyright year.
```

```
glib/dbus-glib.h | 2 +-
glib/dbus-gmain.c | 2 +-
2 files changed, 2 insertions(+), 2 deletions(-)
```

```
commit ea868ce1cd9328ffa31cd9419e23498771f8feb2
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-01-21 09:23:18 +0000
```

2003-01-21 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-connection.c: (dbus_connection_send_message):
Add a new client_serial parameter.

(dbus_connection_send_message_with_reply):
Remove a @todo since we've implemented the blocking function.

(dbus_connection_send_message_with_reply_and_block):
New function that sends a message and waits for a reply and
then returns the reply.

* dbus/dbus-connection.h:
Add new functions.

* dbus/dbus-errors.c: (dbus_result_to_string):
* dbus/dbus-errors.h:
Add new DBUS_RESULT.

* dbus/dbus-message-internal.h:
* dbus/dbus-message.c: (_dbus_message_get_reply_serial),
(_dbus_message_set_sender), (dbus_message_write_header),
(dbus_message_new_reply), (decode_header_data),
(_dbus_message_loader_return_buffer), (_dbus_message_test):
* dbus/dbus-message.h:
Add new functions that set the reply serial and sender.
Also marshal and demarshal them correctly and add test.

* dbus/dbus-protocol.h:
Add new DBUS_MESSAGE_TYPE_SENDER.

* glib/dbus-glib.h:
```

```
* glib/dbus-gmain.c: (watch_callback), (free_callback_data),
(add_watch), (remove_watch), (add_timeout), (remove_timeout),
(dbus_connection_hookup_with_g_main):
* glib/test-dbus-glib.c: (main):
Rewrite to use GIOChannel and remove the GSource crack.
```

```
* test/echo-client.c: (main):
* test/watch.c: (check_messages):
Update for changed APIs
```

```
glib/dbus-glib.h      | 10 +--
glib/dbus-gmain.c    | 219
+++++-----
glib/test-dbus-glib.c | 48 +++-----
3 files changed, 99 insertions(+), 178 deletions(-)
```

```
commit f8a3cb6d30e74be85b583ce9ff3482f91cd26b9a
Author: Anders Carlsson <andersca@gnome.org>
Date: 2003-01-07 20:18:23 +0000
```

```
2003-01-07 Anders Carlsson <andersca@codefactory.se>
```

```
* dbus/dbus-connection-internal.h:
* dbus/dbus-connection.c: (_dbus_connection_new_for_transport),
(_dbus_connection_get_next_client_serial),
(dbus_connection_send_message):
* dbus/dbus-internals.h:
* dbus/dbus-marshal.c: (unpack_uint32), (dbus_unpack_int32),
(dbus_pack_int32), (_dbus_marshall_double), (_dbus_marshall_int32),
(_dbus_marshall_uint32), (_dbus_demarshal_double),
(_dbus_demarshal_int32), (_dbus_demarshal_uint32),
(_dbus_demarshal_string), (_dbus_marshall_get_field_end_pos),
(_dbus_verbose_bytes), (_dbus_marshall_test):
* dbus/dbus-marshal.h:
* dbus/dbus-message-internal.h:
* dbus/dbus-message.c: (_dbus_message_set_client_serial),
(dbus_message_write_header), (_dbus_message_lock),
(dbus_message_new), (dbus_message_ref), (dbus_message_unref),
(dbus_message_get_name), (dbus_message_append_int32),
(dbus_message_append_uint32), (dbus_message_append_double),
(dbus_message_append_string), (dbus_message_append_byte_array),
(dbus_message_get_fields_iter), (dbus_message_iter_ref),
(dbus_message_iter_unref), (dbus_message_iter_has_next),
(dbus_message_iter_next), (dbus_message_iter_get_field_type),
(dbus_message_iter_get_string), (dbus_message_iter_get_int32),
(dbus_message_iter_get_uint32), (dbus_message_iter_get_double),
(decode_header_data), (_dbus_message_loader_return_buffer),
(message_iter_test), (_dbus_message_test):
* dbus/dbus-message.h:
* dbus/dbus-protocol.h:
* dbus/dbus-test.c: (main):
* dbus/dbus-test.h:
```



```
* glib/test-dbus-glib.c: (message_handler), (main):
* test/echo-client.c: (main):
* test/watch.c: (check_messages):
Make messages sendable and receivable for real.
```

```
glib/test-dbus-glib.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
commit 9784f2653287c62f32a637f869ad0029412ebfae
Author: Anders Carlsson <andersca@gnome.org>
Date: 2002-12-27 08:26:26 +0000
```

Ssh

```
glib/.cvsignore | 7 ++++++
1 file changed, 7 insertions(+)
```

```
commit c27a1609f595a75be3910c5185f7312cb93575c7
Author: Havoc Pennington <hp@redhat.com>
Date: 2002-12-24 06:37:32 +0000
```

2002-12-24 Havoc Pennington <hp@pobox.com>

```
* glib/dbus-gthread.c: fix include

* glib/dbus-glib.h: rename DBusMessageHandler for now.
I think glib API needs to change, though, as you don't
want to use DBusMessageFunction, you want to use the
DBusMessageHandler object. Probably
dbus_connection_open_with_g_main_loop()
and dbus_connection_setup_g_main_loop() or something like that
(but think of better names...) that just create a connection
that has watch/timeout functions etc. already set up.

* dbus/dbus-connection.c
(dbus_connection_send_message_with_reply): new function just to
show how the message handler helps us deal with replies.

* dbus/dbus-list.c (_dbus_list_remove_last): new function

* dbus/dbus-string.c (_dbus_string_test): free a string that
wasn't

* dbus/dbus-hash.c: use memory pools for the hash entries
(rebuild_table): be more paranoid about overflow, and
shrink table when we can
(_dbus_hash_test): reduce number of sprintfs and write
valid C89. Add tests for case where we grow and then
shrink the hash table.

* dbus/dbus-mempool.h, dbus/dbus-mempool.c: memory pools
```

```
* dbus/dbus-connection.c (dbus_connection_register_handler)
(dbus_connection_unregister_handler): new functions

* dbus/dbus-message.c (dbus_message_get_name): new

* dbus/dbus-list.c: fix docs typo

* dbus/dbus-message-handler.h, dbus/dbus-message-handler.c:
an object representing a handler for messages.
```

```
glib/dbus-glib.h      |          6 +++---
glib/dbus-gmain.c    |           2 +-
glib/dbus-gthread.c  |           2 +-
3 files changed, 5 insertions(+), 5 deletions(-)
```

```
commit 723dc26c4880ae66688ff707588797e7c83b4fe4
Author: Anders Carlsson <andersca@gnome.org>
Date: 2002-12-16 01:43:52 +0000
```

Doh

```
glib/dbus-glib.h |      2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
commit a2c39755d2c9749d22f547e9b92125509e95aae6
Author: Anders Carlsson <andersca@gnome.org>
Date: 2002-12-16 00:56:23 +0000
```

2002-12-16 Anders Carlsson <andersca@codefactory.se>

```
* glib/dbus-glib.h:
* glib/dbus-gthread.c: (dbus_gthread_init):
Don't use the gdbus prefix for public functions.
```

```
glib/dbus-glib.h      |          4 +---
glib/dbus-gthread.c  |          4 +++-
2 files changed, 3 insertions(+), 5 deletions(-)
```

```
commit 03900e96a260048f23176319003671c17b6a0013
Author: Anders Carlsson <andersca@gnome.org>
Date: 2002-12-16 00:26:05 +0000
```

2002-12-16 Anders Carlsson <andersca@codefactory.se>

```
* Makefile.am:
* configure.in:
Add GLib checks and fixup .pc files

* glib/Makefile.am:
* glib/dbus-glib.h:
* glib/dbus-gmain.c: (gdbus_connection_prepare),
(gdbus_connection_check), (gdbus_connection_dispatch),
```

```
(gdbus_add_connection_watch), (gdbus_remove_connection_watch),
(dbus_connection_gsource_new):
* glib/dbus-gthread.c: (dbus_gmutex_new), (dbus_gmutex_free),
(dbus_gmutex_lock), (dbus_gmutex_unlock), (dbus_gthread_init):
* glib/test-dbus-glib.c: (message_handler), (main):
Add GLib support.
```

```
glib/Makefile.am      | 26 ++++++
glib/dbus-glib.h      | 40 ++++++++
glib/dbus-gmain.c     | 194
+++++
glib/dbus-gthread.c   | 84 ++++++++
glib/test-dbus-glib.c | 60 ++++++++
5 files changed, 404 insertions(+)
```

```
commit 7c0c11bfa019cd4be3f7a62fe936aaf2573fa51e
Author: John (J5) Palmieri <johnp@redhat.com>
Date: 2006-06-28 08:14:04 -0400
```

Initial commit of module dbus-glib

0 files changed

File = ChangeLog.pre-1-0

2006-11-09 John (J5) Palmieri <johnp@redhat.com>

* Released 1.0.0

2006-11-09 John (J5) Palmieri <johnp@redhat.com>

* AUTHORS: added Peter and Tor of windows porting fame

2006-11-08 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: add a note about protocol stability

* doc/TODO: Remove "important for 1.0" section, replace with
"important for 1.2" section

2006-11-08 John (J5) Palmieri <johnp@redhat.com>

* README: remove some of the out of date flags

* doc/TODO: Remove todo items which are done

2006-11-08 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-monitor.c: Handle sigint so we flush stdout

(Patch from Olli Salli <olli dot sall i at collabora dot co dot uk>)

2006-11-08 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-monitor.1: update the manpage for dbus-monitor
(Patch from Olli Salli <olli dot sall i at collabora dot co dot uk>)

2006-11-08 John (J5) Palmieri <johnp@redhat.com>

* AUTHORS: Update with all people listed in the ChangeLog

2006-11-08 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-monitor.c: add profiling tools to dbus-monitor
(Patch from Olli Salli <olli dot sall i at collabora dot co dot uk>)

2006-11-07 Havoc Pennington <hp@redhat.com>

* HACKING: Update release instructions to include stuff about stable releases, branching, etc. May not be totally correct, please fix if needed, but keep instructions up-to-date so we do each stable release consistently in the future.

2006-11-07 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml, doc/dbus-faq.xml, README: various documentation updates. Bump faq/spec versions (not to 1.0; I don't think the spec will be "finished"/1.0 when we ship the 1.0 library).

2006-11-06 John (J5) Palmieri <johnp@redhat.com>

* bus/bus.c: Missed patch - pass in the context to the directory watch

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

* Released D-Bus 1.0 RC 3(0.95)

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

* bus/activation.c (bus_activation_new): don't fail if we can not read the directory as it may not exist

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

* bus/config-parser.c (service_dirs_find_dir): use

loop _dbus_list_get_next_link so we don't get stuck in an infinite
loop
 (start_busconfig_child): move processing of
standard_session_servicedirs
 tags here because they have no content
 (bus_config_parser_content): check we don't have content in
 standard_session_servicedirs tag

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

 * tools/Makefile.am: Make sure the /var/lib/dbus directory is
created
 Packagers need to own this directory

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

 * tools/run-with-tmp-session-bus.sh: fixed script to replace the
<standard_session_servicedirs> tag with a <sevicedir> tag that
points to the test service directory

2006-11-02 John (J5) Palmieri <johnp@redhat.com>

 * configure.in: Add a check for DIR *dirp->dd_fd

 * dbus/dbus-sysdeps-util-unix: if HAVE_DDFD is defined use
DIR *dirp->dd_fd else use DIR *dirp->__dd_fd. This takes care
of both Solaris and Irix

2006-11-01 Havoc Pennington <hp@redhat.com>

 * bus/dbus-daemon.1.in: document standard_session_servicedirs

2006-11-01 John (J5) Palmieri <johnp@redhat.com>

 * configure.in: expose DBUS_DATADIR

 * bus/config-parser.c: add the standard_session_servicedirs
element
 to the parser
 (bus_config_parser_content): process the
standard_session_servicedirs
 element by getting the standard directories from sysdeps and
merging
 them into the service directory list
 (test_default_session_servicedirs): make sure we get what we
expect

 * bus/session.conf.in: replace the servicedir tag with the
standard_session_servicedirs tag

 * dbus/dbus-list.h: remove the typedef of DBusList and place it
in

dbus-sysdeps.h to avoid circular header dependencies

* dbus/dbus-sysdeps.h: add the typedef of DBusList

* dbus/dbus-sysdeps-unix.c (split_paths_and_append): utility function which takes a string of directories delimited by colons, parses them

out, appends a suffix and puts them in a list ignoring empty elements

(_dbus_get_standard_session_servicedirs): returns the standard directories for a session bus to look for service activation files

on Unix which includes the XDG_DATA_HOME, XDG_DATA_DIRS and DBUS_DATADIR directories

* test/data/valid-config-files/many-rules.conf: add the standard_session_servicedirs tag to the valid config file tests

2006-10-30 Havoc Pennington <hp@redhat.com>

* tools/dbus-launch.1, doc/TODO: capture intent to change the command line options of dbus-launch a bit post 1.0

2006-10-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_open_private): document how to choose shared vs. private connections

2006-10-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-test.c: enclose more of the file in the DBUS_BUILD_TESTS check.

* dbus/dbus-sysdeps-pthread.c (PTHREAD_CHECK): fix for DBUS_DISABLE_ASSERT case.

* dbus/dbus-connection.c (dbus_connection_get_unix_user): document that it only works on the server side

* dbus/dbus-bus.c: add a global lock covering the BusData we attach to each connection

(internal_bus_get): lock our access to the BusData

(dbus_bus_register): lock the entire registration process

with _DBUS_LOCK(bus_datas). If we get the lock and registration is already complete, silently return (vs. previous behavior of aborting).

(dbus_bus_set_unique_name): lock the BusData

(dbus_bus_get_unique_name): lock the BusData

2006-10-27 John (J5) Palmieri <johnp@redhat.com>

* bus/config-parser.c (service_dirs_find_dir, service_dirs_append_unique_or_free, service_dirs_append_link_unique_or_free): New static methods for only appending unique service directory names into the service directory list
(merge_included, bus_config_parser_content): Only add unique service directory names into the list

2006-10-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-pthread.c: make the "count" and "holder" variables volatile, suggested by Thiago. Document struct fields. (PTHREAD_CHECK): remove pthread error checking if assertions are disabled, should reduce the no-assertions case to the bare minimum code.

2006-10-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-pthread.c (_dbus_pthread_mutex_lock): change to be recursive
(_dbus_pthread_mutex_unlock): make it recursive
(_dbus_pthread_condvar_wait): save/restore the recursion count
(_dbus_pthread_condvar_wait_timeout): save/restore the recursion count

2006-10-26 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: clarify the UUID text slightly
* dbus/dbus-sysdeps-pthread.c: check for and mostly abort on pthread errors. Add DBusMutexPThread and DBusCondVarPThread in preparation for being able to extend them for e.g. recursive mutexes.

2006-10-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-threads.[hc]: Documentation improvements. Clarify how condition variables relate to recursive mutexes.

* dbus/dbus-sysdeps-pthread.c, dbus/dbus-sysdeps-win-thread.c, dbus/dbus-threads.c: Split the platforms-specific thread implementations into their own files.

* dbus/dbus-sysdeps-pthread.c
(_dbus_pthread_condvar_wait_timeout): invert the return value, it was backward. Not that anything uses it.

2006-10-26 John (J5) Palmieri <johnp@redhat.com>

* dbus-sysdeps-unix.c (_dbus_set_local_creds): Clean up the

LOCAL_CREDS vs CMSCRED stuff a bit. Prefer CMSCRED. This needs to be cleaned up more.

* doc/TODO: Add a todo that *BSD hackers should look at cleaning up the CREDS issue.

2006-10-26 John (J5) Palmieri <johnp@redhat.com>

* configure.in, dbus-1.pc.in: Check to see if thread methods are in glibc or libpthread and add -lpthread to the link stage if it is the latter

2006-10-26 Thiago Macieira <thiago@kde.org>

* dbus/dbus-connection.c (_dbus_connection_open_internal): Fix bug 8780: the connection lock is only required while recording the shared connection, so protect only that code section. Don't require connection_lookup_shared to return a locked connection.

2006-10-26 Thiago Macieira <thiago@kde.org>

* tools/dbus-launch-x11.c (get_session_file, init_x_atoms): check if get_machine_uid() returns NULL before proceeding any further: we can't init the X atoms or create a session file name if there is no machine ID.
This solves a crash reported by some users if --exit-with-session was used without --autolaunch=<machine-id>

2006-10-26 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-launch.c (main): run the dbus-daemon in the builddir if tests are enabled and the DBUS_USE_TEST_BINARY env variable is set

* tools/run-with-tmp-session-bus.sh: set DBUS_USE_TEST_BINARY before we run dbus-launch

* configure.in: define TEST_BUS_BINARY to be the full path to dbus-daemon in the build root

2006-10-25 Sjoerd Simons <sjoerd@luon.net>

* dbus/Makefile.am:
* tools/Makefile.am:
* tools/dbus-uuidgen.1:
Move the machine-id file to /var/lib/dbus

2006-10-24 David Zeuthen <davidz@redhat.com>

* dbus/dbus-threads.c (init_uninitialized_locks): Fix typo so it's


```
    _dbus_assert (thread_init_generation != _dbus_current_generation)
    not _dbus_assert (thread_init_generation ==
_dbus_current_generation)
```

2006-10-24 Thiago Macieira <thiago@kde.org>

- * dbus/dbus-sysdeps.h:
- * dbus/dbus-sysdeps-unix.c: Add function
_dbus_make_file_world_readable that chmods a file to 0644.
- * dbus/dbus-sysdeps-unix.c (_dbus_get_autolaunch_address):
Avoid writing to the static "argv" array, thereby avoiding a
COW on the child process.
- * dbus/dbus-internals.c (_dbus_create_uuid_file_exclusively):
call _dbus_make_file_world_readable on the created file.

2006-10-23 David Zeuthen <davidz@redhat.com>

- * dbus/dbus-memory.c: Use atomic variable to protect
n_blocks_outstanding otherwise OOM will be reported using SMP on
some arches
- * bus/dispatch.c: Add missing end of line characters
- * bus/desktop-file.c (parse_section_start, parse_key_value)
(bus_desktop_file_load): Properly handle OOM
- * dbus/dbus-threads.c (init_uninitialized_locks): Check that
thread_init_generation equals _dbus_current_generation, not 0

2006-10-22 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-connection-internal.h: move prototype of
_dbus_bus_notify_shared_connection_disconnected_unlocked() here
so
it isn't in a public header and doesn't end up in the DbusBus
group in Doxygen

2006-10-21 Havoc Pennington <hp@redhat.com>

- * Makefile.am (EXTRA_DIST): dist cleanup-man-pages.sh so it's in
the tarball if packagers want to run it
- * cleanup-man-pages.sh: Add a script which munges all the
internal
API man pages out of the Doxygen output. This reduces the size of
the installed man pages from 7 to 2 megs, and avoids
namespace-polluting pages. Right now (like Doxygen) this script
isn't in the build, it's something packagers can do manually.

2006-10-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.h: don't include the watch/timeout functions in connection section

2006-10-21 Havoc Pennington <hp@redhat.com>

* Documentation! Whee! Doxygen now 100% silent. If you make it angry again, you will be punished.

2006-10-21 Havoc Pennington <hp@redhat.com>

* More documentation - all public API now documented according to Doxygen

2006-10-21 Havoc Pennington <hp@redhat.com>

* Document a bunch of undocumented stuff

2006-10-21 Havoc Pennington <hp@redhat.com>

* Clean up Doxygen group markers for public API so Doxygen finds everything (not comprehensively fixed for private API). Means all remaining Doxygen warnings are just about missing docs and thus pretty simple to resolve.

2006-10-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-macros.h: add DBUS_GNUC_DEPRECATED macro

* dbus/dbus-message.h: mark dbus_message_iter_get_array_len() as DBUS_GNUC_DEPRECATED

2006-10-21 Havoc Pennington <hp@redhat.com>

* Move all private API into the private Doxygen group sections of the files, so they don't show up in the public API docs

2006-10-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.h: put #ifndef DBUS_DISABLE_DEPRECATED around dbus_message_iter_get_array_len().

* throughout: documentation improvements.

2006-10-20 Havoc Pennington <hp@redhat.com>

* doc/TODO: remove the int64 thing from 1.0 since it doesn't matter, and the message-loader-breaker thing since nobody is going to do it. Add an item to 1.0 about supporting recursive locks in dbus_threads_init_default() though, since it should be easy.

* dbus/dbus-connection.c (`_dbus_connection_read_write_dispatch`):
Fix this in the `!dispatch` case to avoid busy-looping after
disconnection

* More misc docs improvements

2006-10-19 Havoc Pennington <hp@redhat.com>

* more Doxygen-related fixes (lots of moving things from the
public to internal section in `dbus-connection.c`)

2006-10-19 Havoc Pennington <hp@redhat.com>

* Fix a pile of Doxygen warnings and missing docs

2006-10-19 John (J5) Palmieri <johnp@redhat.com>

* `bus/dir-watch-default.c`, `bus/dir-watch-dnotify.c`,
`bus/dir-watch-kqueue.c` (`bus_watch_directory`): Pass in a
`BusContext`

instead of a `void *`. `kqueue` uses this to get the context's loop
while the other modules ignore the parameter. This allows us to
avoid platform conditionals

* `bus/bus.c` (`process_config_postinit`): Pass in the context to the
watch

2006-10-19 John (J5) Palmieri <johnp@redhat.com>

* `bus/messagebus.in`, `bus/rc.messagebus.in`: run `dbus-uuidgen --
ensure`
when starting the system bus

2006-10-18 John (J5) Palmieri <johnp@redhat.com>

* `configure.in`: More fixups for cross compile (Patch from
Marco Pracucci <development at pracucci dot com>)

2006-10-17 Havoc Pennington <hp@redhat.com>

* `tools/dbus-send.c` (`main`): don't close shared connection, fix
from Timo Hoenig

2006-10-17 Havoc Pennington <hp@redhat.com>

* `configure.in` (`CPPFLAGS`): don't do `-DDBUS_API_SUBJECT_TO_CHANGE`
here

* `dbus/dbus.h`: drop the `DBUS_API_SUBJECT_TO_CHANGE` requirement,
since realistically it doesn't look like we'll be changing it
anymore.

2006-10-17 Havoc Pennington <hp@redhat.com>

* dbus/dbus-internals.c (_dbus_warn_check_failed): new function to be used for return_if_fail type warnings; prefixes the pid, and fatal by default.

2006-10-14 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Released 1.0 RC 2(0.94)
Add check for -fPIC and enable it if available

2006-10-14 John (J5) Palmieri <johnp@redhat.com>

* doc/TODO: Remove the check TODO item

2006-10-13 John (J5) Palmieri <johnp@redhat.com>

* dbus-transport-socket.c (exchange_credentials):
Print out more detailed errors if reading or sending credentials fail (Patch from Julio M. Merino Vidal <jmmv at NetBSD dot org>)

2006-10-13 John (J5) Palmieri <johnp@redhat.com>

* configure.in: when checking for posix getpwnam_r assume true for cross compiles

2006-10-13 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Check for gethostbyname first before we check for it in libnsl. On gnu systems it is implemeneted

2006-10-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c
(dbus_connection_send_with_reply_and_block): fix to handle closed connections, from Tabet Ingo bug #8631

2006-10-11 John (J5) Palmieri <johnp@redhat.com>

* configure.in: use AC_TRY_COMPILE for dirfd instead of AC_TRY_RUN

2006-10-11 John (J5) Palmieri <johnp@redhat.com>

* test/name-test/Makefile.am: don't link against both libdbus and libdbus-convenience

2006-10-11 John (J5) Palmieri <johnp@redhat.com>

* configure.in: check for dirfd even if it is a macro
(Patch from Timothy Redaelli <drizzt at gufi dot org>)

2006-10-08 John (J5) Palmieri <johnp@redhat.com>

* configure.in: define _POSIX_PTHREAD_SEMANTICS on solaris
avoid using dirfd on systems that don't have dirfd
(Patch by Brian Cameron <brian dot cameron at sun dot com>)

2006-10-02 John (J5) Palmieir <johnp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_abort): Remove from
#ifndef DBUS_DISABLE_ASSERTS macro to fix distcheck

* dbus/dbus-sysdeps-unix.c (_dbus_print_backtrace): Remove from
#if !defined (DBUS_DISABLE_ASSERT) || defined (DBUS_BUILD_TESTS)
macro because _dbus_abort calls it

* tools/Makefile.am: Add dbus-launch.h to the source list so
distcheck
works

2006-10-02 John (J5) Palmieir <johnp@redhat.com>

* dbus/dbus-sysdeps-util-unix.c (dirent_buf_size): Add check for
MAXNAMELEN for Mac OSX. (Patch from Jonas B <sanoix at gmail dot
com>)

2006-10-02 Thiago Macieira <thiago@kde.org>

* tools/dbus-launch.c (main): make it complain and abort
execution in case an unknown option is given. Also, process
the '--' argument as the end of arguments.

2006-10-01 Havoc Pennington <hp@redhat.com>

* tools/dbus-launch.c (print_variables): if no syntax is given,
don't print something that's sort-of-half-sh-syntax, just print
a plain key-value pairs thing.

* tools/dbus-launch-x11.c: use machine ID rather than hostname
for
the local machine representation (but still have the hostname in
the display). Remove the hostname from the display if it is
localhost. Change session files to be named
~/.dbus/session-bus/machine-display. Change X atoms to be
underscore-prefixed so nobody whines about ICCCM compliance.
Otherwise name them the same as the env variables.
Change session file format to include key-value pairs and an
explanatory comment. Keys are the same as the env variables.
(set_address_in_x11): X property format can't depend on

sizeof(pid_t) on a particular machine, fix to always be 32 bits

* tools/dbus-launch.c: make --autolaunch take a machine id argument. If --autolaunch is used with a program to run, complain for now (but add a FIXME). Also, don't look for existing bus if there's a program to run (but add a FIXME).

* dbus/dbus-sysdeps-unix.c (_dbus_get_autolaunch_address): pass machine uuid to dbus-launch (avoids linking dbus-launch to libdbus just to get this, and avoids duplicating uuid-reading code).

* tools/dbus-launch.1: clarify various things

2006-10-01 Havoc Pennington <hp@redhat.com>

* test/test-service.c (path_message_func): remove broken extra unref that was hidden by the bugs in dbus-connection.c/dbus-bus.c

* test/test-shell-service.c (path_message_func): same fix

* dbus/dbus-connection.c
(_dbus_connection_get_dispatch_status_unlocked): break up the function a little for clarity and fix the notification of dbus-bus.c to not require dispatch to be complete

* dbus/dbus-connection.c (dbus_connection_unref): improve the warning when you try to finalize an open connection.

2006-10-01 Havoc Pennington <hp@redhat.com>

* dbus/dbus-bus.c
(internal_bus_get): only weak ref the connection; this means _dbus_bus_notify_shared_connection_disconnected_unlocked can be called safely in any context
(_dbus_bus_notify_shared_connection_disconnected_unlocked): don't unref

* dbus/dbus-connection.c
(_dbus_connection_get_dispatch_status_unlocked): move _dbus_bus_notify_shared_connection_disconnected_unlocked here when queuing Disconnected instead of when the Disconnected message arrives, so dbus_bus_get() won't return closed connections.

2006-10-01 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c
(_dbus_connection_close_if_only_one_ref):
Add a hack to make DBusNewConnectionFunction work right.

* dbus/dbus-server-socket.c (handle_new_client_fd_and_unlock):
use
the hack here. Also, fix the todo about refcount leak.

* dbus/dbus-server-debug-pipe.c (_dbus_transport_debug_pipe_new):
and use the hack here

* dbus/dbus-connection.c: Kill the "shared" flag vs. the
"shareable" flag; this was completely broken, since it meant
dbus_connection_open() returned a connection of unknown
shared-ness. Now, we always hold a ref on anything opened
as shareable.

Move the call to notify dbus-bus.c into
connection_forget_shared_unlocked, so libdbus consistently
forgets
all its knowledge of a connection at once. This exposed numerous
places where things were totally broken if we dropped a ref
inside
get_dispatch_status_unlocked where
connection_forget_shared_unlocked was previously, so move
connection_forget_shared_unlocked into
_dbus_connection_update_dispatch_status_and_unlock. Also move the
exit_on_disconnect here.

(shared_connections_shutdown): this assumed weak refs to the
shared connections; since we have strong refs now, the assertion
was failing and stuff was left in the hash. Fix it to close
still-open shared connections.

* bus/dispatch.c: fixup to use dbus_connection_open_private on
the
debug pipe connections

* dbus/dbus-connection.c (dbus_connection_dispatch): only notify
dbus-bus.c if the closed connection is in fact shared
(_dbus_connection_close_possibly_shared): rename from
_dbus_connection_close_internal
(dbus_connection_close, dbus_connection_open,
dbus_connection_open_private): Improve docs to explain the deal
with when you should close or unref or both

* dbus/dbus-bus.c
(_dbus_bus_notify_shared_connection_disconnected_unlocked):
rename
from _dbus_bus_check_connection_and_unref_unlocked and modify to
loop over all connections

* test/test-utils.c (test_connection_shutdown): don't try to
close
shared connections.

* test/name-test/test-threads-init.c (main): fix warnings in here

* dbus/dbus-sysdeps.c (_dbus_abort): support DBUS_BLOCK_ON_ABORT env variable to cause blocking waiting for gdb; drop DBUS_PRINT_BACKTRACE and just call _dbus_print_backtrace() unconditionally.

* configure.in: add -export-dynamic to libtool flags if assertions enabled so _dbus_print_backtrace works.

* dbus/dbus-sysdeps-unix.c (_dbus_print_backtrace): use fprintf instead of _dbus_verbose to print the backtrace, and diagnose lack of -rdynamic/-export-dynamic

2006-09-30 Havoc Pennington <hp@redhat.com>

* dbus/dbus-bus.c (dbus_bus_get_private, dbus_bus_get) (internal_bus_get): fix screwy code formatting. whoever committed that was not paying attention!

2006-09-30 Havoc Pennington <hp@redhat.com>

* configure.in (LT_CURRENT, LT_AGE): increment current and age to reflect addition of interfaces.

* doc/dbus-specification.xml: describe a new org.freedesktop.DBus.Peer.GetMachineId method

* dbus/dbus-string.c (_dbus_string_skip_white_reverse): new function (_dbus_string_skip_white, _dbus_string_skip_blank): use new DBUS_IS_ASCII_BLANK, DBUS_IS_ASCII_WHITE macros and fix assertion at end of skip_white (_dbus_string_chop_white): new function

* bus/connection.c (bus_connections_setup_connection): call dbus_connection_set_route_peer_messages.

* dbus/dbus-connection.c (_dbus_connection_peer_filter_unlocked_no_update): modify to support a GetMachineId method.

Also, support a new flag to let the bus pass peer methods through to apps on the bus, which can be set with dbus_connection_set_route_peer_messages.

Finally, handle and return an error for anything unknown on the Peer interface, which will allow us to extend the Peer interface in the future without fear that we're now intercepting something apps were wanting to see.

- * tools/dbus-uuidgen.c: a thin wrapper around the functions in dbus/dbus-uuidgen.c
- * dbus/dbus-uuidgen.c: implement the bulk of the dbus-uuidgen binary here, since most of the code is already in libdbus
- * dbus/dbus-sysdeps.c (_dbus_read_local_machine_uid): read the uid from the system config file
- * dbus/dbus-internals.c (_dbus_generate_uid, _dbus_uuid_encode) (_dbus_read_uid_file_without_creating) (_dbus_create_uid_file_exclusively, _dbus_read_uid_file): new uid-related functions, partly factored out from dbus-server.c
- * dbus/dbus-sysdeps.c (_dbus_error_from_errno): convert EEXIST to DBUS_ERROR_FILE_EXISTS instead of EEXIST
- * dbus/dbus-protocol.h (DBUS_ERROR_FILE_EXISTS): add file exists error
- * tools/dbus-cleanup-sockets.1: explain what the point of this thing is a bit more
- * autogen.sh (run_configure): add --config-cache to default configure args
- * dbus/dbus-internals.h (_DBUS_ASSERT_ERROR_IS_SET): disable the error set/clear assertions when DBUS_DISABLE_CHECKS is defined
- * tools/dbus-launch.c (main): if xdisplay hasn't been opened, don't try to save address, fixes crash in make check

2006-09-30 Thiago Macieira <thiago@kde.org>

- * configure.in: add DBUS_BINDIR as a #define to C source code.
- * tools/dbus-launch.c
- * tools/dbus-launch.h
- * tools/dbus-launch-x11.c:
- * tools/dbus-launch.1: Add the --autolaunch option to dbus-launch, which makes it scan for an existing session started with --autolaunch. With that option, it also creates an X11 window and saves the bus address and PID to it.
- * dbus/dbus-sysdeps.h:
- * dbus/dbus-sysdeps-unix.c (_dbus_get_autolaunch_address): Add a function that runs "dbus-launch --autolaunch" to retrieve the running D-Bus session address (or start one if none was running)
- * dbus/dbus-transport.c: Add the handler for the "autolaunch:"

address protocol, which tries to get the running session from dbus-launch.

- * dbus/dbus-bus.c:
- * dbus/dbus-internals.h: Make "autolaunch:" be the default D-Bus session bus address.

- * dbus/dbus-connection.c: Fix horrible typo in error message.

2006-09-18 John (J5) Palmieri <johnp@redhat.com>

- * tools/Makefile.am: use @EXPANDED_DATADIR@ instead of @DATADIRNAME@

2006-09-17 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-transport.c (_dbus_transport_get_is_authenticated): fix so that if unix_user_function is set on windows, we still do the default check that the auth identity matches the bus identity

2006-09-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-transport.c (_dbus_transport_open): modify to delegate to _dbus_transport_open_platform_specific, _dbus_transport_open_socket, and _dbus_transport_open_debug_pipe

- * dbus/dbus-transport-protected.h: add _dbus_transport_open_platform_specific

2006-09-16 Havoc Pennington <hp@redhat.com>

Attempt auditing public API to remove all cases where a Unix function returns weird emulated goo to Windows. This probably breaks the bus daemon on Windows, to fix it again we may need to stop processing unix-specific config options on Windows, and may need to add Windows-specific public API or config options.

- * configure.in (LT_CURRENT, LT_AGE): increment current and age, to reflect added interfaces; should not break soname.

- * dbus/dbus-transport.c (_dbus_transport_get_is_authenticated): do not invoke unix user function on Windows. Kind of a hacky fix, but we don't want a "unix uid" leaking out on Windows.

- * dbus/dbus-connection.c (dbus_connection_get_socket): add new API

to get the socket fd on Windows or UNIX
(dbus_connection_get_unix_fd): make this always fail on Windows

2006-09-16 Havoc Pennington <hp@redhat.com>

* dbus/dbus-server.c (dbus_server_listen): change how this works to be able to delegate to a set of handlers that can succeed, fail, or choose not to handle. Allows us to have dbus_server_listen_platform_specific.

* dbus/dbus-server-socket.c (_dbus_server_new_for_tcp_socket): factor out the tcp socket stuff to be used on windows, leaving unix domain socket only in dbus-socket-unix.c

* dbus/dbus-transport-socket.c
(_dbus_transport_new_for_tcp_socket): factor out the tcp socket stuff to be used on windows, leaving unix domain socket only in dbus-transport-unix.c

* dbus/dbus-connection.c (dbus_connection_get_unix_user): insert temporary hack to be sure this fails on windows
(dbus_connection_get_unix_process_id): ditto

2006-09-16 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-unix.c (_dbus_open_tcp_socket)
(_dbus_open_unix_socket, _dbus_open_socket): change API for _dbus_open_socket so the domain/type/protocol from system headers are not required. This is kind of pointless though unless we move part of _dbus_connect_tcp_socket into sysdeps.c instead of sysdeps-unix.c, which would mean adding a wrapper around bind/listen/etc.

Also, add DBusError to the socket-opening functions so they don't require use of errno.

2006-09-16 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-unix.h: small change to Peter's patch to make dbus-sysdeps-unix-util.c build, add unix-specific sysdeps header.

* dbus/dbus-sysdeps.h, dbus-sysdeps-unix.c: patch from Peter KÄmmel bug #8249 to make the sysdeps.h read/write/open/close functions specifically for sockets only, and move generic read/write/open/close into unix-specific code.

2006-09-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-transport-unix.c (unix_do_iteration): fix a valgrind complaint about accessing revents when poll returns 0, from Julian Seward.

2006-09-14 John (J5) Palmieri <johnp@redhat.com>

* Released 1.0 RC 1 (0.93)

2006-09-14 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps-util-unix.c (_dbus_write_pid_file):
use _dbus_close instead of close

2006-09-14 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps.c: Add support for LOCAL_CREDS socket
credentials. Fixes "external" authentication under e.g. NetBSD
which does not support any other socket credentials mechanism.
(Patch from Julio M. Merino Vidal <jmmv at NetBSD dot org>)

2006-09-14 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-threads.c: Allow recursive mutex's to be passed into
dbus_threads_init and be used by the dbus mutex functions to
avoid deadlocks.

* doc/TODO: Remove recursive mutex dbus_connection_dispatch TODO
item

2006-09-13 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps-util-unix.c (_dbus_directory_get_next_file):
use threadsafe readdir_r instead of readdir

2006-09-13 John (J5) Palmieri <johnp@redhat.com>

* dbus-threads.c (dbus_threads_init_default): New method for
initializing the internal thread implementation (Patch from
Alexander Larsson <alexl at redhat dot com>)

2006-09-11 John (J5) Palmieri <johnp@redhat.com>

* remove a bunch of todo items from the 1.0 list

2006-09-11 John (J5) Palmieri <johnp@redhat.com>

* bus/activation.c, bus/desktop-file.c: Distinguish between OOM
and
key not found

2006-09-11 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-internal.c: Add dbus_is_verbose so we can have more
complex debugging code

* dbus/dbus-marshal-basic.c (_dbus_marshal_read_fixed_multi):
Move

between the test suite ifdefs
(_dbus_verbose_bytes): return if verbosity is not enabled

2006-09-11 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-marshal-recursive-util.c, dbus/dbus-marshal-recursive.c:
remove DBusMark

2006-09-10 Havoc Pennington <hp@redhat.com>

patch mostly by Peter KÄmmel, bug #8211

* dbus/dbus-sysdeps-unix.c: new file, which splits out
unix-specific stuff in dbus-sysdeps.c

* dbus/dbus-sysdeps.c: don't include any UNIX-only headers,
and move unix-specific stuff to dbus-sysdeps-unix.c

* configure.in: check HAVE_ERRNO_H

2006-09-08 John (J5) Palmieri <johnp@redhat.com>

* bus/test-main.c (main): Initialize threading during tests

* dbus/dbus-connection.c (_dbus_connection_new_for_transport):
Unlock connection on error
(generate_local_error_message): static method for generating
an error message when we don't have a message to reply to
(_dbus_connection_block_pending_call): Send a disconnect error
instead of just a timeout (NULL) when the bus gets disconnected
while blocking for a reply.

2006-09-08 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_dispatch): Properly
remove
filters (Patch from Kimmo HÄmÄlÄinen
<kimmo dot hamalainen at nokia dot com>)

2006-09-07 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection-internal.h: Add prototype for
_dbus_connection_test_get_locks to remove warning

2006-09-07 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-spawn.c (_dbus_spawn_async_with_babysitter): Make
sure we
close child_err_report_pipe[WRITE_END] on exec (Patch from
Kimmo HÄmÄlÄinen <kimmo dot hamalainen at nokia dot com>)

2006-09-07 John (J5) Palmieri <johnp@redhat.com>

* Clean up close calls to use `_dbus_close` (Patch from Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>)

2006-09-06 John (J5) Palmieri <johnp@redhat.com>

* doc/TODO:

- Remove pending call locking todo item
- `_dbus_connection_open` now holds hard ref. Remove todo item
- do proper locking on `_dbus_bus_check_connection_and_unref` and handle `DBUS_BUS_STARTER`. Remove todo item
- Warn on closing of a shared connection. Remove todo item

* `bus/bus.c`, `bus/connection.c`, `bus/dispatch.c`, `dbus/dbus-bus.c`, `dbus/dbus-connection.c`: Use the `_dbus_connection_close_internal` so we don't get the warning when closing shared connections

* `test/test-service.c`, `test/test-shell-service.c`: Applications don't close shared connections themselves so we `unref` instead of `close`

* `test/test-utils.c` (`test_connection_shutdown`): Close the connection

* `dbus/dbus-bus.c` (`_dbus_bus_check_connection_and_unref`): Changed to

this `_dbus_bus_check_connection_and_unref_unlocked` since we only call

method on a locked connection.

Make sure we call `_dbus_connection_unref_unlocked` instead of `dbus_connection_unref` also.

Handle `DBUS_BUS_STARTER` correctly

* `dbus/dbus-connection.c` (`connection_record_shared_unlocked`): Mark as shared and hard ref the connection

(`connection_forget_shared_unlocked`): Remove the hard ref from the connection

(`_dbus_connection_close_internal_and_unlock`): New internal function

which takes a locked connection and unlocks it after closing it

(`_dbus_connection_close_internal`): New internal function which acts

like the original `dbus_connection_close` method by grabbing a connection

lock and calling `_dbus_connection_close_internal_and_unlock`

(`dbus_connection_close`): Public close method, warns when the app tries to close a shared connection

2006-09-06 John (J5) Palmieri <johnp@redhat.com>

* `bus/driver.c`:

(bus_driver_generate_introspect_string): New method for populating a DBusString with the introspect data
(bus_driver_handle_introspect): Move introspect generation code to bus_driver_generate_introspect_string

* bus/main.c:
(introspect): New function which prints out the introspect data and exits
(main): Add a --introspect switch

2006-09-06 John (J5) Palmieri <johnp@redhat.com>

* doc/TODO: Removed dtd publishing item.
It seems the dtd has already been added at
<http://www.freedesktop.org/standards/dbus/1.0/introspect.dtd>

2006-09-05 John (J5) Palmieri <johnp@redhat.com>

* doc/TODO, various source files: Audited todo's and FIXME's and prepended the ones we should be looking at with 1.0. Those prepended with 1.0? need clarification or might not be needed for 1.0

2006-09-05 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-pending-call.c: Add some checks so we don't crash on NULL in the public API (Patch from Kimmo Hamalainen <kimmo dot hamalainen at nokia dot com>)

2006-09-05 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Fix for OS-X compile time endian issues (patch from Benjamin Reed <ranger at befunk dot com>)

2006-09-05 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Check for no-common compiler flag (OS-X fix from Benjamin Reed <ranger at befunk dot com>)

2006-09-01 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-launch.c: Add a sigterm handler (patch from Frederic Crozat <fcrozat at mandriva dot com>)

2006-08-29 Havoc Pennington <hp@redhat.com>

* test/test-service.c (path_message_func): fix lack of return value

* dbus/dbus-sysdeps.c (_dbus_printf_string_upper_bound): fix formatting, remove #ifdef, and fix docs. #ifdef doesn't make any more sense than on anything else in this file.
(_dbus_get_tmpdir): add const to return value, and keep the results of the various getenv around in a static variable.

2006-08-29 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-util.c, dbus/dbus-sysdeps-util-unix.c: change from Ralf Habacker to move UNIX-specific sysdeps into a separate file.

2006-08-25 John (J5) Palmieri <johnp@redhat.com>

* test/Makefile.am: change find to use syntax that works with non gnu versions of find

2006-08-25 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-transport.c: fix invalid deref when checking if a vtable method exists (Patch from Christian Ehrlicher <ch dot ehrlicher at gmx dot de>)

2006-08-25 John (J5) Palmieri <johnp@redhat.com>

* configure.in, dbus/Makefile.am, test/name-test/run-test.sh, test/name-test/Makefile.am: Fixed some issues with getting get dbus to build with builddir != srcdir (Taken from a patch by Bernard Leak <bernard at brenda-arkle dot demon dot co dot uk>)

2006-08-25 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Fix DBUS_DAEMONDIR to use EXPANDED_BINDIR for the default case

2006-08-25 John (J5) Palmieri <johnp@redhat.com>

* configure.ac, bus/Makefile.am: Generalize kqueue support so that it works on any system providing this interface, not only FreeBSD.
For example, NetBSD. (Patch by Julio M. Merino Vidal <jmmv at NetBSD dot org>)

2006-08-20 Havoc Pennington <hp@redhat.com>

* doc/dbus-faq.xml, doc/dbus-tutorial.xml: some improvements to

the docs

2006-08-18 John (J5) Palmieri <johnp@redhat.com>

* Released 0.92

2006-08-18 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-threads.c (dbus_threads_init): change the documentation to reflect the init late change

* bus/bus.c (bus_context_new): Check user before we fork so we can print out an error message a user will be able to see

2006-08-18 John (J5) Palmieri <johnp@redhat.com>

Patch provided by Ralf Habacker (ralf dot habacker at freenet dot de)

* dbus/dbus-sysdeps.c, dbus/dbus-threads.c, dbus/dbus-internals.h:

Add two more global locks for use on windows platforms. These are unused on non-windows platforms but are not ifdefed out to avoid potential bugs (i.e. the few bytes lost does not warrant the extra maintenance and complexity that having separate sets of locks would cause)

2006-08-18 John (J5) Palmieri <johnp@redhat.com>

* bus/services.c (bus_registry_acquire_service): Return an error when an application tries to register the org.freedesktop.DBus Bus Name

* bus/services.c (bus_registry_release_service): Return an error when an application tries to release the org.freedesktop.DBus Bus Name

2006-08-17 Alp Toker <alp@atoker.com>

* doc/dbus-specification.xml: Fix some minor typos.

2006-08-17 John (J5) Palmieri <johnp@redhat.com>

* configure.in: use \$with_init_scripts instead of \$operating_system to determine where to store the pid since the init scripts manipulate

the pid file (patch from Marcelo Ricardo Leitner
<marcelo dot leitner at terra dot com dot br>.

2006-08-16 John (J5) Palmieri <johnp@redhat.com>

```
* dbus/dbus-threads.c: Add static DBusList
*uninitialized_mutex_list and
    static DBusList *uninitialized_condvar_list to support new late
    initialization threading model. In this model threads can be
initialized
    even after the D-Bus API has been used but still needs to be
initialized
    before the second thread has been started. Mutexes and condvar
addresses
    are stored in the two static lists and are replaced with actual
locks
    when threads are initialized.
    (_dbus_mutex_new_at_location): New method for creating a mutex
and placing
    the location into the static list
    (_dbus_mutex_free_at_location): New method for removing a mutex
location
    from the static list and freeing the mutex
    (_dbus_condvar_new_at_location): New method for creating a
conditional
    variable and placing the location into the static list
    (_dbus_condvar_free_at_location): New method for removing a
conditional
    variable location from the static list and freeing the
conditional variable
    (init_uninitialized_locks): Atomic method which goes through the
static
    lists of mutex and condvar location and updates them with actual
locks
    (init_global_locks): changed to init_locks

* dbus/dbus-connection.c:
    (_dbus_connection_test_get_locks): New method for tests to check
connections
    (_dbus_connection_new_for_transport): Use the new at_location
mutex and
    condvar API
    (dbus_connection_allocate_data_slot): Pass in the global lock
address
    to _dbus_data_slot_allocator_alloc

* dbus/dbus-dataslot.c:
    (_dbus_data_slot_allocator_alloc): Use the address of the mutex
instead of the mutex itself

* dbus/dbus-message.c:
```

(dbus_message_allocate_data_slot): Pass in the global lock address
to _dbus_data_slot_allocator_alloc

* dbus/dbus-pending-call.c:
(dbus_pending_call_allocate_data_slot): Pass in the global lock address
to _dbus_data_slot_allocator_alloc

* dbus/dbus-server.c:
(_dbus_server_init_base): Use the new at_location mutex API
(dbus_server_allocate_data_slot): Pass in the global lock address
to _dbus_data_slot_allocator_alloc

* test/name-test/test-threads-init.c: New test case for late thread initialization

2006-08-14 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-dataslot.c (_dbus_data_slot_allocator_alloc):
Change _dbus_abort to _dbus_assert_not_reached because
_dbus_abort
causes compile problems when asserts are turned off
Keeping _dbus_warn for printing out the message so even if
asserts are turned off the user gets the messages that something
is
wrong

2006-08-14 John (J5) Palmieri <johnp@redhat.com>

Patches by Kjartan Maraas <kmaraas at gnome dot org>

* bus/services.c (bus_service_list_queued_owners):
Add a pointer cast to fix compiler warning

* dbus/dbus-dataslot.c (_dbus_data_slot_list_get):
return a NULL instead of FALSE since the return type
is not expecting a boolean

* dbus/dbus-marshal-basic.c (_dbus_marshal_test):
Remove unused variable

* dbus/dbus-marshal-recursive-util.c (node_new):
return a NULL instead of FALSE since the return type
is not expecting a boolean

* dbus/dbus-server-debug-pipe.c (_dbus_transport_debug_pipe_new):
Send a NULL into _dbus_transport_new_for_fd instead of a FALSE
because we are expecting a pointer not a boolean

* dbus/dbus-sysdeps-util.c (_dbus_get_tmpdir):

add void as the parameter so some compilers don't complain

- * dbus/dbus-transport-unix.c
(`_dbus_transport_new_for_domain_socket`,
`_dbus_transport_new_for_tcp_socket`):
Send a NULL into `_dbus_transport_new_for_fd` instead of a FALSE because we are expecting a pointer not a boolean
- * test/shell-test.c (`test_command_line`):
cast the second argument to `_dbus_list_append` to avoid compiler warnings
- * test/test-names.c (`main`): remove unused variable
- * test/test-service.c (`check_hello_from_self_reply`):
Initialize `echo_message` and `echo_reply` to NULL
- * test/test-shell-service.c (`handle_echo`):
Remove unused variable and cast the third parameter passed to `dbus_connection_get_object_path_data` to avoid compiler warnings
- * test/name-test/test-names.c (`clear_message_queue`):
Remove unused function
- * test/name-test/test-pending-call-dispatch.c:
Fix format string in `printf`

2006-08-14 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-bus.c:
- * test/name-test/test-names.c:
Remove test method `dbus_bus_connection_get_unique_name` because we already have public method `dbus_bus_get_unique_name` that does the exact same thing

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-signature.c:
fix typos in Julio's previous patch which cause make check to fail

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-address.c (`_dbus_address_test`): Revert leaking `strcmp`.
In any case it was wrong since this is a test checking to see if address parsing is correct. There was no need to get the true tmp directory.

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-macros.h: Revert the addition of stddef.h
as we should not be adding it to library headers

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-signature.c:
Fix the unit tests so that they work if assertions are not
enabled.
(patch from Julio M. Merino Vidal <jmmv at NetBSD dot org>)

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* tools/run-with-tmp-session-bus.sh:
* test/name-test/run-test.sh:
Remove bashisms (patch from Julio M. Merino Vidal
<jmmv at NetBSD dot org>)

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* configure.in: add a version (>= 2.6.0) check for libxml2

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* configure.in: make sure the compiler supports -Wfloat-equal

* bus/dir-watch-dnotify.c: move functional code after
variable declarations (C99 fix) (patches from Jens Granseuer
<jensgr at gmx dot net>)

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-macros.h:
add #include <stddef.h> so that Sun compilers don't complain
about
the defining NULL

2006-08-10 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps.c:
* dbus/dbus-address.c:
* bus/activation.c:
* test/shell-test.c:
don't hardcode tmp directory (patch from Dave Meikle
<dmk at davemeikle dot co dot uk>)

2006-08-09 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-dataslot.c (_dbus_data_slot_allocator_alloc):
Change an assert to a detailed warning and abort.
When allocator->lock != mutex it indicates that the user has
failed

to initialize threads before using the D-Bus library. This warning helps the user identify the issue and fix their app.

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

These are all patches from Kjartan Maraas <kmaraas at gnome dot org>

with cleanups of bugs found from Coverity reports:

- * dbus/dbus-sysdeps-util.c (_dbus_write_pid_file): close the file on error to avoid a leak

- * bus/expirelist.c (bus_expire_list_test): Check for NULL on dbus_new0

- * bus/activation.c (update_directory): remove dead code

- * bus/config-parser.c (merge_service_context_hash, start_selinux_child): Fix some leaks

- * bus/bus.c (process_config_every_time): Fixed a leak

- * bus/desktop-file.c (parse_key_value): Fixed leak

- * bus/selinux.c (bus_selinux_id_table_insert): Fixed leak

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-object-tree.c (_dbus_object_subtree_new): remove dead code

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

- * tools/run-with-tmp-session-bus.sh: use #!/bin/sh instead of #!/bin/bash

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

- * Doxyfile.in: fix the INPUT line so we can generate docs when sourcedir != builddir (patch from Cygwin Ports maintainer <yseikowitz at users dot sourceforge dot net>

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-sysdeps.h:
- * dbus/dbus-sysdeps.c:

* dbus/dbus-string.c:
s/_dbus_printf_length/_dbus_printf_string_upper_bound to conform
with
GLib's function which does the same thing

* configure.in:
* bus/Makefile.am:
* bus/dir-watch-default.c:
* bus/dir-watch-dnotify.c:
* bus/dir-watch-kqueue.c:
Add kqueue directory watching for freebsd and split the directory
watching code into separate files per method/arch
(patches from Timothy Redaelli <drizzt at gufi dot org>)

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

* configure.in:
* tools/Makefile.am:
* tools/dbus-launch.c:
* bus/Makefile.am:
allow --with-dbus-daemon-dir switch to be used to make the
daemon install to a separate bindir like /usr/libexec
(patch from Brian Cameron <brian dot cameron at sun dot com>)

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

* bus/Makefile.am (install-data-hook): removed the slash after
\$(DESTDIR) so we don't get the double slash which does not work
in windows (patch from Andras Porjesz
<andras dot porjesz at siemens dot com>)

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps.h:
* dbus/dbus-sysdeps.c:
add _dbus_printf_length (patch from Peter Kämmel
<syntheticpp at gmx dot net>)

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-internals.c: unistd.h is not used
(_dbus_verbose_real): only enable verbose printing
if DBUS_VERBOSE environment var is set to '1'
(patch from Peter Kämmel <syntheticpp at gmx dot net>)

2006-08-08 John (J5) Palmieri <johnp@redhat.com>

* configure.in: add a GCC flag check for -Wdeclaration-after-
statement
so we still compile on older gcc (patch from Frederic Peters
<fpeters at 0d dot be>)

2006-08-04 Havoc Pennington <hp@redhat.com>

- * configure.in: add -Wdeclaration-after-statement
- * dbus/dbus-connection.c: change all the pending call stuff to reflect the fact that pending call operations use the connection lock
- * dbus/dbus-pending-call.c: add locking here
- * dbus/dbus-errors.c (struct DBusRealError): don't make the name field const consistent with how message field is done

2006-08-03 John (J5) Palmieri <johnp@redhat.com>

- * s/D-BUS/D-Bus/g

2006-08-03 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-object-tree.c: Patch by Benjamin Otte <in7yl18 at public.uni-hamburg dot de> - fix invalid read/write reported by valgrind

2006-07-24 John (J5) Palmieri <johnp@redhat.com>

- * Released 0.91

2006-07-22 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-connection.c:
(_dbus_connection_attach_pending_call_unlocked):
(connection_timeout_and_complete_all_pending_calls_unlocked):
Make sure we set timeout_added on pending calls to FALSE when we remove the timeout from the connection

2006-07-21 John (J5) Palmieri <johnp@redhat.com>

- * Removed some extra bindings stuff lingering around (thanks timo)
- * dbus-pendingcall.c (_dbus_pending_call_new):
s/dbus_connection_ref/_dbus_connection_ref_unlocked fixes
assertion
when we tried to take a lock on an already locked connection

2006-07-17 John (J5) Palmieri <johnp@redhat.com>

- * Released 0.90

2006-07-17 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-marshall-basic.c (_dbus_type_to_string): support 64-bit

ints, reported by Owen Taylor

2006-07-17 John (J5) Palmieri <johnp@redhat.com>

- * doc/TODO:
- * dbus/dbus-bus.c:
- * dbus-errors.c:
- * dbus/dbus-marshall-validate.c:
Removed TODO items which were fixed or are no longer relevant

2006-07-17 John (J5) Palmieri <johnp@redhat.com>

- * dbus-gt4-1.pc.in, dbus-sharp.pc.in:
Remove stray files from the stripped bindings

2006-07-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-pending-call.c
(_dbus_pending_call_set_timeout_error):
Improve the error message on reply timeout

2006-07-14 John (J5) Palmieri <johnp@redhat.com>

- * Remove all bindings

2006-07-13 John (J5) Palmieri <johnp@redhat.com>

- * dbus-connection.c (dbus_connection_send_with_reply): return TRUE
and set pending_reply out arg to NULL if connection is disconnected
(connection_timeout_and_complete_all_pending_calls_unlocked): New static method for cleaning up pending calls on disconnect
(_dbus_connection_get_dispatch_status_unlocked): If we have pending calls queued timeouts on disconnect
- * dbus/dbus-pending-call.c (_dbus_pending_call_set_connection):
Remove

2006-07-13 Carlos Garcia Campos <carlosgc@gnome.org>

- * bus/activation.[ch] (bus_activation_list_services): new function to
get the list of services that can be activated
- * bus/dispatch.c: test coverage for the new bus method
ListActivatableNames
- * bus/driver.c: new bus method ListActivatableNames to get the list of
services that can be activated

* doc/dbus-specification.xml: ListActivatableNames method documentation

2006-07-12 John (J5) Palmieri <johnp@redhat.com>

* dbus/Makefile.am: add dbus-pending-call-internal.h to the list of source files

2006-07-12 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-message-factory.c:
Fix index into an array (patch by Peter KÄmmel <syntheticpp at gmx.net>)

2006-07-12 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection-internal.h:
* dbus/dbus-connection.c:
* file dbus/dbus-pending-call.c:
* dbus/dbus-pending-call.h:
Make DBusPendingCall an opaque type even to D-Bus internals

2006-07-07 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.h: remove connection_disconnect and replace with connection_close

* dbus/dbus-connection.c: include dbus-bus.h
(_dbus_connection_read_write_dispatch): make static

2006-07-07 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_close): removed deprecated function
(dbus_connection_dispatch): On disconnect unref any shared connections

* dbus/dbus-bus.c (_dbus_bus_check_connection_and_unref): new function for cleaning up shared connections on disconnect
(internal_bus_get): get a hard reference to shared connections when they are created

* doc/TODO: Remove items which are no longer relevant or have been fixed
Split 1.0 todo items with a 0.90 freeze todo list

2006-06-14 Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c:

Free a leaking GArray (surely not!) in dbus_g_method_return.

2006-06-14 Ross Burton <ross@openedhand.com>

- * tools/Makefile.am:
- * tools/dbus-monitor.c:
Don't use the GLib bindings in dbus-monitor (patch from Ralf Habacker).

2006-06-14 Ross Burton <ross@openedhand.com>

- * tools/dbus-print-message.c:
Also print the object path when outputting signals or method calls.

2006-06-13 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/src/Makefile.am: install the qdbus.h header.
This allows people to actually use the installed code.

2006-06-12 Ross Burton <ross@openedhand.com>

- * glib/dbus-gproxy.c:
Don't leak a GArray when firing signals (thank Rob Taylor for review).

2006-06-12 Thiago Macieira <thiago.macieira@trolltech.com>

- * Released 0.62

2006-06-12 Thiago Macieira <thiago.macieira@trolltech.com>

- * dbus/dbus-arch-deps.h.in: Remove spurious semi-colons that break pedantic builds. Closes bug 6043 (patch approved by Havoc back in February).

2006-06-12 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/src/qdbusintegrator.cpp: Fix bug in parsing async methods that took a QDBusMessage parameter.
- * qt/src/qdbusbus.h: Add a default flag for RequestName.
- * qt/tools/dbus.cpp: Don't use automatic call because we might be calling an async method: request a reply.

2006-06-11 Thiago Macieira <thiago.macieira@trolltech.com>

- * test/qt/*: Update the testcases, including testing the new functionality of sending null QByteArray and QString over the bus. Add new headertest test and restore the old qdbusxmlparser test.

2006-06-11 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/tools/dbuscpp2xml.cpp: Compile on Windows.
- * qt/tools/dbusidl2cpp.cpp: Add missing newline.

- * qt/examples/Makefile.am:
- * qt/examples/chat.h: Use UI-generated files with the ui_*.h form.

- * qt/src/qdbusmarshall.cpp: Allow sending of QString() and QByteArray() (nulls) over the bus.
- * qt/src/qdbusabstractinterface.cpp: Use the correct variable, the one that has the signature suffix stripped.
- * qt/src/qdbusreply.h: Make some methods const.

2006-06-09 Thiago Macieira <thiago.macieira@trolltech.com>

Patch from Timo Hoenig <thoenig@suse.de>.

- * qt/dbus/Makefile.am: New file. Fix "make dist", add all headers required during build to EXTRA_DIST.
- * qt/src/Makefile.am: Fix "make dist", add 'qdbus.h' to EXTRA_DIST.
- * qt/Makefile.am: Fix "make dist", add 'dbus' to DIST_SUBDIRS.
- * configure.in: Fix "make dist", take care that the Makefile for qt/dbus is being generated.

2006-06-07 John (J5) Palmieri <johnp@redhat.com>

- * bus/bus.c: Fix eavesdropping on method calls

2006-06-07 John (J5) Palmieri <johnp@redhat.com>

- * configure.in:
- * dbus/dbus-userdb-util.c:

Add Solaris console owner patch from Artem Kachitchkine

2006-06-07 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/Makfile.am:
- * qt/src/Makefile.am: Fix the EXTRA_DIST after the reorganisation. Thanks to Timo Hoenig for pointing this out.

2006-06-06 Robert McQueen <robot101@debian.org>

- * glib/dbus-gtype-specialized.c: Fix obvious leak of GArray in every call to dbus_g_type_get_struct.

2006-06-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Fix the failing test where static string pointers were put into a GPtrArray-based specialised collection, and then freed along with the array. GValues which you add into collections or maps which have the NOCOPY flag set are assumed to not belong to the caller, so rather than the existing pointer-stealing semantics, they are copied instead. Given that the main consumers of this abstraction are the bindings themselves, I don't think this is too bad, but others should watch their choice of take vs set_static.

2006-06-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Spotted a warning about the return value of g_slist_prepend not being used. Fixed copying of slist-based specialised collections, then wrote a test case and found that it was all broken. Went on to fix iterating and appending too. Good thing nobody uses this code yet.

2006-06-06 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Remove duplicated code by having all of the iterators use gvalue_take_ptrarray_value (the GValues themselves are discarded without unsetting, so it makes no difference whether we take or set_static). Remove cases for G_TYPE_POINTER because there really is nothing useful we can do with them in our specialised types -we *need* boxed copy/free functions at the very least.

2006-06-05 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/dbus: Add directory. I had forgotten to add this yesterday after the move...

* qt/examples/Makefile.am:

* qt/examples/dbus.cpp: Moved to qt/tools/dbus.cpp.

* qt/tools/Makefile.am:

* qt/tools/dbus.cpp: Moved from qt/examples/dbus.cpp. Added feature to get and set properties.

Added validation of service, object path and interface names.

* qt/tools/dbusidl2cpp.cpp: Two new features:

1) Allow specifying both the header and the source file names, by separating them with a colon.

2) Don't write an interface output if the -p switch wasn't given, but the -a was.

* qt/src/*: Fix usage of Iterators and ConstIterators.

Fix shadowing of variables by other variables (-Wshadow).

Fix keyword-cleanliness in headers.

Fix ASCII-cast (QLatin1String, QLatin1Char).

Fix validation of member names.

Add extra checking of introspection data during XML parsing.

Various bug fixes.

2006-06-04 Thiago Macieira <thiago.macieira@trolltech.com>

* dbus/Makefile.am:

* dbus/qdbus.h: Remove unnecessary file. This is mirrored into qt/dbus/qdbus.h now.

2006-06-04 Thiago Macieira <thiago.macieira@trolltech.com>

* configure.in: Make --disable-qt actually do something. Patch inspired by Zack Rusin.

2006-06-04 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/: Update to Subversion r548032.

This includes a big reorganisation of the files inside the subdir.

2006-05-30 Sjoerd Simons <sjoerd@luon.net>

* dbus/dbus-sysdeps.c: Make tcp socket connection error somewhat more

clear:

"Failed to connect to socket <host>:<port> <error>" instead of
"Failed to connect to socket <host>: <error>:<port>"

* dbus/dbus-transport-unix.c: Fix crash when no host option is given
for a tcp transport.

2006-05-29 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/*: Update the QtDBus bindings up to revision 546310 in Subversion.

This adds the dbuscpp2xml tool, that parses a C++ header and outputs a D-BUS Introspection XML.

2006-05-21 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c: Put in a pile of assertions that the proxy name is not NULL when it shouldn't be. Also a couple of possible fixes for #4637 though I don't understand why the bug happens, to be honest... also the object constructor has an assert name != NULL and the name is only currently NULL for peer-to-peer proxies that I don't think anyone uses? So it should be asserting. Anyway, for now at least see if we get an earlier assertion failure.

* glib/dbus-gvalue-utils.c: Put in a couple of assertions for apparently broken code to be sure the tests fail and someone will fix them...

2006-05-07 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusmarshall.cpp: Fix a problem of demarshalling lists and arrays when they had a single element: has_next returns false, even before you read the element. So, instead, check the array length.

2006-05-06 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusmessage.cpp:
* qt/qdbustypehelper_p.h:
* qt/qdbusintegrator.cpp: gcc 3.4 doesn't like Q_FOREACH when the list is a const-reference

2006-05-03 John (J5) Palmieri <johnp@redhat.com>

* Adding old doc patch that never got applied
* dbus/bus.c (dbus_bus_add_match): Add documentation
* doc/dbus-specification.xml: Add documentation for the match rules and the AddMatch and RemoveMatch methods

2006-05-02 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/dbusidl2cpp.cpp: There's no callAsync. Use the correct call (r535506)
* qt/dbusidl2cpp.cpp:
* qt/qdbusabstractadaptor.cpp:
* qt/qdbusabstractadaptor.h: Make QDBusAdaptorConnector be a sibling of the QDBusAbstractAdaptor objects instead of the parent. (r535848)
* qt/dbusidl2cpp.cpp:

- * qt/qdbusabstractinterface.cpp:
- * qt/qdbusabstractinterface.h:
- * qt/qdbusabstractinterface_p.h:
- * qt/qdbusinterface.cpp: Make properties in interfaces actually work. The code that was generated would not compile, due to moc calls to functions that did not exist. They now shall. (r536571)

2006-04-30 Thiago Macieira <thiago.macieira@trolltech.com>

- * Makefile.am:
- * configure.in:
- * dbus-qt4-1.pc.in: Add a pkg-config file for libdbus-qt4-1. Thanks to Brad Hards for providing the patch

2006-04-29 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/dbusidl2cpp.cpp: There's no callAsync. Use the correct call. (r535506)

2006-04-29 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/examples/dbus.cpp: Enhance error messages and use QDBusInterfacePtr.

2006-04-29 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/qdbusinterface.h: Rename QDBusRef to QDBusInterfacePtr and disable the copy operators. (r533772, r534746)
- * qt/qdbuserror.h: Remove the automatic cast to bool. (r533929)
- * qt/qdbusabstractinterface.cpp:
- * qt/qdbusabstractinterface.h: Change the default call mode to not use the event loop. Add convenience call() methods that take a CallMode parameter. (r534042)
- * qt/qdbusconnection.h: Change the default call mode to not use the event loop. (r534042)
- * qt/qdbusinterface.cpp:
- * qt/qdbusinterface.h: Add a method to tell us if the interface is valid (since we don't return a null pointer anymore) (r534099)
- * qt/qdbusinterface_p.h: Don't crash if metaObject is 0 (r534101)
- * qt/qdbusinternalfilters.cpp: Decouple the introspection function in two so taht we get the chance to introspect without having a QDBusMessage (r534102)

* qt/qdbusbus.h:
* qt/qdbusconnection.cpp:
* qt/qdbusconnection_p.h:
* qt/qdbusintegrator.cpp: Keep a list of our own names to avoid a round-trip to the server when attempting to introspect one of our own objects. Also make sure the filter functions match the empty interface as well. (r534108)
Don't keep the connection names. Instead, trust the unique connection name (r534111)
Remove event loop usage (r534112)

2006-04-29 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusintegrator.cpp: Fix assertion failure spotted by Brad Hards.

2006-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gproxy.c: Fix properties so that they can be given in any order, making it easier for people who inherit from this object.

2006-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue-utils.c: Patch from Jakub Stachowski to fix leaking of memory from within pointer arrays and lists. Fixes bug #6300.

2006-04-28 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue.c: Patch from Jakub Stachowski to fix a leak in generating struct signatures. Fixes bug #6083.

2006-04-28 Robert McQueen <robot101@debian.org>

* qt/Makefile.am: Tweak CLEANFILES from qdbusconnection.moc to qdbusconnection_p.moc.

2006-04-24 John (J5) Palmieri <johnp@redhat.com>

* README, INSTALL: Doc fixes
Patch from Brad Hards <bradh at frogmouth.net>

2006-04-23 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/examples/dbus.cpp: Use the new merged-interface mode for the dynamic meta object. No need to guess which interface to call.

2006-04-23 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusconnection_p.h:
* qt/qdbusmetaobject.cpp:
* qt/qdbusmetaobject_p.h:
* qt/qdbusintegrator.cpp: Use the new merged-interface mode for the dynamic meta object. No need to guess which interface to call.
* qt/qdbusabstractinterface_p.h:
* qt/qdbusconnection.cpp:
* qt/qdbusintegrator.cpp:
* qt/qdbusinterface.cpp:
* qt/qdbusinterface.h: Make findInterface always return a non-null pointer.
Add a QDBusRef that looks and behaves like DCOPRef.

2006-04-23 Thiago Macieira <thiago.macieira@trolltech.com>

* dbus/dbus-connection.c: Interfaces are optional in method calls, so don't give up if the interface parameter is NULL. Patch reviewed by Havoc Pennington.

2006-04-23 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusreply.h: Add default constructor and operator= (r532625)
* qt/qdbustypehelper_p.h: Use a clean namespace: no foreach() in public headers (r532952)
* qt/qdbusabstractinterface.cpp:
* qt/qdbusabstractinterface_p.h: Add the AutoDetect mode and make it the default (r532951)

2006-04-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c: Fix asserts
Patch from Tim Moloney <t.moloney at verizon.net>

2006-04-19 John (J5) Palmieri <johnp@redhat.com>

* mono/Connection.cs, mono/Message.cs: Check Target type
Patch from Aaron Bockover (abockover at novell.com)

2006-04-13 Thiago Macieira <thiago.macieira@trolltech.com>

* INSTALL: fine-tune the requirements for the Qt4 binding.

2006-04-16 Daniel P. Berrange <dan@berrange.com>

* tools/dbus-print-message.c: Added support for printing of all remaining data types. Fixed logic for indentation of compound data types.

2006-04-15 Daniel P. Berrange <dan@berrange.com>

* INSTALL: fill out content of file providing DBus specific build installations, followed by generic Auto* INSTALL file content

2006-04-13 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusintegrator.cpp: Work around g++ 3.3 bug.
Patch by Stefan Eilers. (r529537)

2006-04-13 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/qdbusinternalfilters.cpp: Don't show the parent's contents (r528208)

2006-04-10 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/Makefile.am: fix the dependency for qdbusconnection_p.moc. It's included in qdbusintegrator.cpp, not in qdbusconnection.cpp.
Thanks to Jakub Stachowski <stachowski@hypair.net> for spotting this.

2006-04-10 Thiago Macieira <thiago.macieira@trolltech.com>

* qt/examples/listnames.cpp:
* qt/examples/Makefile.am: Three ways to list the names on the bus.

2006-04-10 Thiago Macieira <thiago.macieira@trolltech.com>

* test/qt/tst_hal.cpp: Remove the waiting, since it's not needed anymore. Requires Qt 4.1.3 to work properly.
(r528148)

2006-04-10 Thiago Macieira <thiago.macieira@trolltech.com>

Merge from Subversion:

* qt/qt-dbus.qdocconf: Update Trolltech's webpage link to something that exists (r526315)
* qt/qdbusinternalfilters.cpp: Correctly detect non-scriptable slots/signals (r526316)
* qt/qdbusinternalfilters.cpp: Fix the setProperty call and also return an unknown-method error if the parameters don't match for org.freedesktop.DBus.Properties. (r526842)
* qt/examples/dbus.cpp: Allow passing of QVariants (r526843)
* qt/qdbusintegrator.cpp: Restore the proper order of delivery: don't make method returns be delivered on priority (r528150)

2006-03-28 Thiago Macieira <thiago.macieira@trolltech.com>

* configure.in qt/Makefile.am: add qt/examples

- * qt/examples: Add QtDBus example programs:
 - hello: Hello, World
 - ping: Simple method-calling program
 - pong: Simple object-exporting program (not using adaptors)
 - complexping: Interactive method-calling program (also gets and sets properties).
 - complexpong: Sample program exporting methods, signals and properties, using adaptors.
 - dbus: Simple implementation of a generic method-calling program, similar to 'dbus-send', but with semantics similar to 'dcop'.
 - chat: Simplistic chat program, implemented using signals and the system bus. Looks like IRC.

2006-03-28 Thiago Macieira <thiago.macieira@trolltech.com>

- * configure.in: Detect QtGui (necessary for one of the example programs). Note: this increases the minimum required version of Qt to 4.1.3.

2006-03-28 Thiago Macieira <thiago.macieira@trolltech.com>

- * test/qt/*: Sync with KDE Subversion revision 523647. Update the testcases to the new API. Remove testcases for classes that are no longer public or have been removed.

2006-03-28 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/*:
 - * dbus/qdbus.h: Sync with KDE Subversion revision 523647. Hopefully, this will be the last of the source-incompatible changes. Documentation has been improved; support for QList<basic-types> has been added; QDBusObject is gone; QDBus(Abstract)Interface is now a QObject with auto-generated meta-object; QDBusIntrospection is marked private, since QMetaObject can be used now; lots of bugfixes.

2006-03-16 John (J5) Palmieri <johnp@redhat.com>

Patch from Milosz Derezynski <internalerror at gmail.com>

- * configure.in:
 - Output the service directory in the configure summary
- * dbus-1.pc.in:
 - Add session_bus_services_dir

2006-03-10 Ross Burton <ross@openedhand.com>

- * tools/dbus-print-message.c:
 - Add support for object paths and signatures.

2006-03-06 Sjoerd Simons <sjoerd@luon.net>

- * bus/bus.c: (bus_context_reload_config): Flush the user database cache on config reload.
- * bus/dbus-daemon.1.in: Also note that SIGHUP flushes the user/group information caches
- * dbus/dbus-hash.c: (_dbus_hash_table_remove_all):
- * dbus/dbus-hash.h: Add function to remove all entries from a hash table
- * dbus/dbus-userdb.c: (_dbus_user_database_flush):
- * dbus/dbus-userdb.h: Add function to flush all user/group information caches.

2006-03-06 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/dbusidl2cpp.cpp:
- * qt/Makefile.am: add the dbusidl2cpp tool, the replacement for dcopyidl2cpp, found in the KDE installations (or the more modern kalyptus): generate Qt4 C++ code for the input XML introspection. Currently no IDL parsing.

2006-03-06 Thiago Macieira <thiago.macieira@trolltech.com>

- * test/qt/*: Update the self-tests.

2006-03-06 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/*:
- * dbus/qdbus.h: Sync with KDE Subversion revision 516237. This represents the first feature-complete version of the Qt4 bindings since I took over maintainership.

2006-03-06 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/Doxyfile: Adding a Doxyfile for the Qt4 bindings dir. This is C++, so we can't use the Dbus ones.

2006-03-02 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx: Remove reference to sys/cdefs.h (Patch from Artem Kachitchkine <Artem.Kachitchkin at Sun.COM>)

2006-03-02 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-connection.c: (_dbus_connection_block_pending_call): Check to see if our data has already been read off the connection by another blocking pending call before we block in poll. (check_for_reply_and_update_dispatch_unlocked):

Code taken from `_dbus_connection_block_pending_call` - checks for an already read `reply` and updates the `dispatch` if there is one.

* `test/name-test/test-pending-call-dispatch.c`:
New test for making sure we don't get stuck polling a `dbus` connection which has no data on the socket when blocking out of order on two or more pending calls.

2006-02-28 Thiago Macieira <thiago.macieira@trolltech.com>

* `qt/Makefile.am`: Patch by Sjoerd Simons. More `.moc` issues: `make/automake` don't detect that we're talking about the same `.lo` file if I specify the full path to the source files.

2006-02-26 Havoc Pennington <hp@redhat.com>

* `bus/dbus-daemon.1.in`: improve the language in a couple spots I noticed

* `dbus/dbus-bus.c` (`internal_bus_get`): in the error message if the session bus variable is unset, suggest "man `dbus-launch`" and "man `dbus-daemon`" to figure out how to fix the problem

2006-02-25 Havoc Pennington <hp@redhat.com>

* `glib/dbus-glib-tool.c` (`usage`): fix up the usage message, someone should make this thing use the new `glib` options parser

2006-02-25 Thiago Macieira <thiago.macieira@trolltech.com>

* `qt/Makefile.am`: Patch by Sjoerd Simons. Fix the path to the `.lo` files taking `moc` sources.

2006-02-25 Havoc Pennington <hp@redhat.com>

* `dbus/dbus.h`, `dbus/Makefile.am`: add `dbus-signature.h` to `dbus.h` and install it as a public header

2006-02-24 John (J5) Palmieri <johnp@redhat.com>

* Released 0.61

2006-02-24 John (J5) Palmieri <johnp@redhat.com>

* `proxies.py`: Fix the callchain

2006-02-24 John (J5) Palmieri <johnp@redhat.com>

* patch from Sjoerd Simons <sjoerd at debian.org>:

* `dbus/dbus-sysdeps-util.c` (`_dbus_group_info_free`): Moved to

dbus/dbus-sysdeps.c

- * dbus/dbus-userdb.c (`_dbus_group_info_free_allocated`): Don't call `_dbus_group_info_free_allocated` which causes infinite loop, correctly call `_dbus_group_info_free`

2006-02-20 Thiago Macieira <thiago.macieira@trolltech.com>

- * qt/qdbusinterface_p.h:
- * qt/qdbusinterface.cpp: Use the standard `org.freedesktop.DBus.Method.NoReply` annotation for the "async" calls instead of creating one for us.

- * qt/qdbusconnection_p.h:
- * qt/qdbusintegrator.cpp: Remove debugging code.

- * qt/qdbusintegrator.cpp:
- * qt/qdbusmessage.cpp:
- * qt/qdbusmessage_p.h:
- * qt/qdbusmessage.h: Change the behaviour of automatic reply-sending: now a reply is always sent, unless the caller didn't request one or if the user slot has already sent one.

2006-02-16 Robert McQueen <robot101@debian.org>

- * `configure.in`: Patch from Debian packages by Sjoerd Simons <sjoerd@debian.org> to add `--with-qt-moc` and `--with-qt3-moc` arguments so it's possible to build both bindings in the same tree.

- * `qt/Makefile.am`: Fix truncated value so that `make dist` works.

2006-02-16 Robert McQueen <robot101@debian.org>

- * `acinclude.m4`, `configure.in`: Patch from Brad Hards <bradh@frogmouth.net> to avoid warnings from `autoconf 1.9` by improving quoting, re-ordering a few checks, and a few other aesthetic tidy-ups.

2006-02-16 Robert McQueen <robot101@debian.org>

- * `dbus/dbus-message.c` (`dbus_message_iter_get_fixed_array`): Patch from Rob Taylor <rob.taylor@collabora.co.uk> to correct a bogus assertion that the next element to read from the iter is fixed in size. This is not the case when you are at the end of the iter, because the next element type is `INVALID`.

- * `dbus/dbus-string.c` (`_dbus_string_init_const_len`): Correct a bogus assert which means that you may not initialise a 0-length string unless you provide a non-NULL pointer. This prevented you from marshalling messages containing zero-length arrays in

some cases.

* glib/dbus-gvalue.c (demarshal_collection_array): Another patch from Rob to correct bogus asserts when trying to demarshal an array and get_fixed_array got you 0 elements. Append nothing to the GArray in this case.

* test/glib/test-dbus-glib.c: Add a test case for round-tripping an empty array via the glib bindings. Without all of the above patches, this new test fails.

2006-02-16 Robert McQueen <robot101@debian.org>

* glib/dbus-gmain.c: Make the previous commit compile.

* python/_dbus.py, python/matchrules.py: Patch from Ole Andre Ravnaas <ole.andre.ravnaas@collabora.co.uk> to allow you to specify sender_keyword="foo", path_keyword="bar" when adding a signal listener, so that you can bind to signals generically but still do something useful in your callback.

* python/dbus_bindings.pyx: Demarshal the byte type as unsigned chars so that they're not cast to chars and made negative. Thanks to Jakub Stachowski for reporting this and testing the fix.

2006-02-15 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-glib.h:

* glib/dbus-gmain.h:

(dbus_g_connection_open): new method for opening a connection to an arbitrary address in the glib bindings

* ChangeLog: checkin last entry which doesn't seem to be committed

2006-02-13 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-launch.c: Fixed sh syntax output

2006-02-13 Robert McQueen <robot101@debian.org>

* glib/dbus-binding-tool-glib.c, glib/dbus-gmain.c, glib/dbus-gsignature.c, glib/dbus-gtype-specialized.c, glib/dbus-gtype-specialized.h, glib/dbus-gvalue-utils.c, glib/dbus-gvalue-utils.h, glib/dbus-gvalue.c:

Patch from Rob Taylor <rob.taylor@collabora.co.uk> to add a big missing piece of the glib bindings jigsaw puzzle. This modifies the existing specialised types to have N type parameters (rather than the current 1 or 2 for arrays and dictionaries respectively).

You can then use this to get a glib type to represent any arbitrary

D-Bus struct type using dbus_g_type_get_struct. The only

implementation of these types is with GValueArrays as before, but it's now possible to store these in arrays, emit them in signals, etc.

2006-02-10 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-signature.c (dbus_signature_iter_recurse): Correctly deal with nested arrays (Bug #5823) Patch by Thiago Macieira <thiago.macieira at trolltech.com>

2006-02-10 John (J5) Palmieri <johnp@redhat.com>

* mono/doc/Makefile.am: Fix parallel make problem with mono-doc (Bug #4213) Patch from Doug Goldstein <cardoe at gentoo.org>

2006-02-10 John (J5) Palmieri <johnp@redhat.com>

* bus/connection.c (bus_connections_expect_reply): Make pending reply limit not common to all connections (Bug #5416) Patch from Kimmo Hämmäläinen <kimmo.hamalainen at nokia.com>

2006-02-10 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-launch.c: Fixed csh syntax output (Bug #5720)

2006-02-10 John (J5) Palmieri <johnp@redhat.com>

* gcj/Makefile.am: point to correct jar command (Bug #4529) patch from Doug Goldstein <cardoe at gentoo.org>

2006-02-09 Joe Shaw <joeshaw@novell.com>

* mono/Arguments.cs: Fix a magic number in the mono bindings that doesn't work on 64 bit arches. Patch from Peter Johanson.

2006-01-27 Robert McQueen <robot101@debian.org>

* glib/dbus-binding-tool-glib.[ch]: Patch based on Ricardo Kekki's patch to use an annotation org.freedesktop.DBus.GLib.ClientCSymbol when generating the client-side methods, instead of overloading CSymbol which broke everything horribly. My apologies.

2006-01-27 Robert McQueen <robot101@debian.org>

* glib/dbus-gtype-specialized.[ch], glib/dbus-gvalue-utils.c: Patch by me and Rob Taylor to add a simple_free function to D-Bus map and collection types, which allows those types which can be freed with a GDestroyNotify (such as GHashTables and GArrays, but not GPtrArrays) to be stored as the values in hashtables.

* test/glib/test-dbus-glib.c, test/glib/test-service-glib.{c,xml}:
Patch by Rob Taylor to add nested dicts to the glib tests to check
the above code works, and appears not to leak when called repeatedly.

2006-01-27 Robert McQueen <robot101@debian.org>

* glib/dbus-gvalue.c (demarshal_valuearray): Patch from Rob Taylor
to free a D-Bus allocated string with dbus_free () instead of g_free ().

2006-01-27 Iain Holmes <iain@openedhand.com>

* glib/dbus-gproxy.c (dbus_g_proxy_dispose): Protect the dispose method from being called multiple times.

2006-01-19 Robert McQueen <robot101@debian.org>

* glib/dbus-binding-tool-glib.c: Patch from Rob Taylor <rob.taylor@collabora.co.uk> to add support for generating bindings
to arrays that are represented as GPtrArrays rather than GArrays (ie
size-variable things, such as strings, objects, structs, etc).

2006-01-05 Robert McQueen <robot101@debian.org>

* dbus/dbus-glib.h, glib/dbus-gproxy.c: Patch from Ricardo Kekki <ricardo.kekki@movial.fi> to make it possible to inherit from DBusGProxy, by splitting the DBusGProxy struct into a public part
and a private part, and moving the setting of the DBusGProxyManager into a
connection property, allowing proper GObject construction.

2006-01-05 Robert McQueen <robot101@debian.org>

* glib/dbus-binding-tool-glib.c: Patch from Ricardo Kekki <ricardo.kekki@movial.fi> to make dbus-binding-tool heed C symbol
name annotations when generating glib client bindings.

2005-12-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-shared.h: Call it shared constants instead of shared macros

* dbus/dbus-protocol.h: add DOxygen markup to quiet warnings

2005-12-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-shared.h: add DOxygen markup to quiet warnings

2005-12-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-macros.h: correct DOxygen end of section (s/}@/{@})

2005-12-19 Ross Burton <ross@openedhand.com>

* doc/dbus-tutorial.xml:

Document the Glib client-side bindings, and list all possible annotations.

2005-12-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/bus.c (dbus_bus_release_name): Add documentation

2005-12-06 Robert McQueen <robot101@debian.org>

* python/service.py: s/sucessful/successful/ so we're allocating to and reading from the same variable. Oops.

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

* Released 0.60

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

* test/qt/Makefile.am: build from srcdir

* qt/qtconnection.cpp (requestName): Changed PROHIBIT_REPLACE to ALLOW_REPLACE

Note - this code is wrong and needs to be fixed by the Qt binding developers. The flags should be treated as bitfields and not enums.

* qt/qtconnection.h: Change ProhibitReplace to AllowReplace

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-list.c (_dbus_list_insert_after_link, _dbus_list_insert_after, link_after): remove #ifdef DBUS_BUILD_TESTS since we use these methods in production code

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_read_write): Add new

method for getting messages off the bus in the absence of a mainloop. This method is much like `dbus_connection_read_write_dispatch` except it does not dispatch the messages to a registered filter function. Instead it allows a developer to process messages by directly popping them off the bus.

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

- * `bus/desktop-file.c` (`parse_key_value`): Ignore locales allowing the parser to continue instead of returning error
- (`bus_desktop_file_load`): Do not free parser data when `parse_section_start` or `parse_key_value` fails because it was already freed by `parser_free` (patch from Carlos Garcia Campos <carlosgc at gnome.org>)

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

- * `dbus/dbus-auth.c`, `dbus/dbus-connection.c`, `dbus/dbus-keyring.c`, `dbus/dbus-server-debug-pipe.c`, `glib/dbus-binding-tool-glib.c`, `glib/dbus-glib-tool.c`, `glib/dbus-gparser.c`, `glib/dbus-gproxy.c`, `test/test-segfault.c`, `test/test-utils.c`, `test/glib/test-dbus-glib.c`, `tools/dbus-cleanup-sockets.c`, `tools/dbus-launch.c`, `tools/dbus-tree-view.c`, `tools/dbus-viewer.c`: Various cleanup of dead code and compiler warnings (patch from Kjartan Maraas <kmaraas at gnome.org>)

2005-11-30 John (J5) Palmieri <johnp@redhat.com>

- * `glib/dbus-gmain.c` (`connection_setup_add_watch`): plugged a leak (patch from Carlos Garnacho Parro <carlosg at gnome.org>)

2005-11-27 Robert McQueen <robot101@debian.org>

- * `python/dbus_bindings.pyx`: Repair my previous commit which reverted part of the preceding one. Oops. Merge patch by Johan Hedberg <johan.hedberg@nokia.com> to fix marshalling of 16-bit integer values on big-endian platforms.

- * `test/python/test-client.py`: Add some 16-bit integers to the test values.

2005-11-27 Carlos Garcia Campos <carlosgc@gnome.org>

- * `glib/dbus-gobject.c`: Append a `GValue` instead of a basic type in method return message for property getters

2005-11-27 Robert McQueen <robot101@debian.org>

* python/dbus_bindings.pyx: Fix a bug where doing a strict append with type v of an instance of dbus.Variant(foo, type='x') caused it to be boxed twice before sending over the bus.

* python/dbus_bindings.pyx, python/service.py, test/python/test-client.py: Update the constants for the new request_name flags, and update comments/test cases now that queueing is the default action.

2005-11-22 John (J5) Palmieri <johnp@redhat.com>

* configure.in:
- Change version to 0.60 for impending release
- upped the sonames because of ABI and API breakage

2005-11-22 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Add test/name-test/Makefile to the generated Makefile list

* dbus/dbus-shared.h (#define DBUS_NAME_FLAG_ALLOW_REPLACEMENT): New flag which replaces DBUS_NAME_FLAG_PROHIBIT_REPLACEMENT (#define DBUS_NAME_FLAG_DO_NOT_QUEUE): New flag for specifying not to queue an owner if it can't be the primary owner

* bus/bus.h: Add new internal BusOwner struct

* bus/driver.c (bus_driver_handle_hello): Send flags (0 for default) to bus_registry_ensure and don't set the prohibit_replacement flag since they are now set per BusOwner and not per name. (bus_driver_handle_list_queued_owners): bus method (ListQueuedOwners) that returns the list of connections in a name's connection queue

* bus/services.c (struct BusService): remove prohibit_replacement field (struct BusOwner): new struct for keeping track of queued connections and their associated flags for the queue (struct BusRegistry): add a BusOwner memory pool (bus_registry_new): initialize the BusOwner memory pool (bus_registry_unref): free the BusOwner memory pool (_bus_service_find_owner_link): new internal method for searching the queue for a specific connection (bus_owner_set_flags): new method for adding setting the flags on a bus owner (bus_owner_new): new method that creates a BusOwner object from the

pool and sets its flags
(bus_owner_ref, bus_owner_unref): ref counting for BusOwner
objects
(bus_registry_ensure): Add the flags parameter
(bus_registry_acquire_service): Switch from using raw connections
to
using the BusOwner struct
Add new state machine for dealing with the new set of flags
(bus_registry_set_service_context_table, struct
OwnershipCancelData,
cancel_ownership, free_ownership_cancel_data,
add_cancel_ownership_to_transaction, struct OwnershipRestoreData,
restore_ownership, free_ownership_restore_data,
add_restore_ownership_to_transaction): Switch to using BusOwner
instead of raw connections
(bus_service_add_owner): Add flags parameter
Switch to using BusOwner instead of raw connections
Add state machine for dealing with the new set of flags
(bus_service_swap_owner): Swaps the first and second owners in
the
queue. Used to make sure proper signals are sent when a service
looses
or gains primary ownership. We never insert an owner at the top
of the
queue. Instead we insert it in the second position and then
swap.
(bus_service_remove_owner): Remove the owner from the queue
sending
out the NameLost and NameOwnerChanged signals if the we were the
primary owner
(bus_service_get_primary_owners_connection): New method that
extracts
the connection from the primary owner
(bus_service_get_primary_owner): Returns the BusOwner instead of
the
connection
(bus_service_get_allow_replacement): Changed from the old
bus_service_get_prohibit_replacement method. Checks the flags of
the
primary owner and returns if it can be replaced or not
(bus_service_set_prohibit_replacement): removed
(bus_service_has_owner): returns TRUE if and owner with
the specified connection exists in the queue

* dbus/dbus-bus.c (dbus_bus_connection_get_unique_name): New
helper
method that only compiles if tests are enabled. Allows us to get
the
unique name of a connection so we can check it against the queue
when
doing regression tests

* bus/activation.c (bus_activation_send_pending_auto_activate),
bus/dispatch.c (bus_dispatch),
bus/driver.c (bus_driver_handle_get_service_owner,
bus_driver_handle_get_connection_unix_user,
bus_driver_handle_get_connection_unix_process_id,
bus_driver_handle_get_connection_selinux_security_context),
bus/signals.c (connection_is_primary_owner):
use bus_service_get_primary_owners_connection instead of
bus_service_get_primary_owner

* dbus/dbus-sysdeps.c (_dbus_connect_unix_socket,
_dbus_listen_unix_socket): Calculate the length of the socket
path and use that instead of using a fixed length which was
causing socket names to contain many trailing Nul bytes.

* dbus/dbus-glib-lowlevel.h, glib/dbus-gobject.c
(dbus_g_method_get_sender): New method for extracting the sender
from a DBusGMethodInvocation
(dbus_g_method_return_get_reply): changed name to
dbus_g_method_get_reply
(dbus_g_method_return_send_reply): changed name to
dbus_g_method_send_reply

* doc/dbus-specification.xml: New docs that describe how the new
queueing system works and talks about the changes to the how
we specify socket names

* glib/examples/example-service.c,
glib/examples/example-signal-emitter.c,
glib/examples/statemachine/statemachine-server.c:
Changed the RequestName flags to the new system

* test/name-test/ (test-names.c, run-test.sh, Makefile.am): New
regression test suite for testing various states of the new
queueing system

2005-11-15 Robert McQueen <robot101@debian.org>

* dbus/dbus-glib-lowlevel.h, glib/dbus-gobject.c: Patch from Rob
Taylor to add two methods, dbus_g_method_return_get_reply and
dbus_g_method_return_send_reply, to allow you to get the reply
message from a DBusGMethodInvocation, append arbitrary stuff to
it,

and send it. The GLib bindings can't marshal a return value of
something like a(s) if the array is empty - ultimately they
should be
made to heed the signature of the out arguments as the Python
bindings
now can, but this is a workable interim solution which might have
other applications.

2005-11-15 Robert McQueen <robot101@debian.org>

* bus/driver.c, bus/services.c, bus/services.h: Add a ReleaseName method to org.freedesktop.DBus to release a bus name or give up waiting in the queue for it.

* dbus/dbus-bus.c, dbus/dbus-bus.h, dbus/dbus-shared.h: Add a dbus_bus_release_name method to send the ReleaseName method calls.

Add constants for the return values to dbus/dbus-shared.h.

* doc/dbus-specification.xml: Document the new ReleaseName method in the specification.

* python/dbus_bindings.pyx: Add a low-level python binding for the release name method.

* python/exceptions.py, python/service.py: Make freeing BusName objects release the name. Add a NameExistsException, and fix a bug with creating UnknownMethodException.

* test/python/test-client.py: Add tests for freeing BusName objects causing names to be released.

2005-11-14 Robert McQueen <robot101@debian.org>

* python/service.py: Include the traceback in the error reply when we send an exception over the bus. BEST_PATCH_EVER

2005-11-14 David Zeuthen <davidz@redhat.com>

Patch from Timo Hoenig <thoenig@suse.de>.

* bus/bus.c: I've recently investigated why the automatic reload of configuration files does not work as expected.

Currently, reloading configuration files does only work when running dbus-daemon with --nodaemon. If we are running as daemon we're hitting a dnotify bug once we fork the process.

We're initializing the dnotify fds before calling fork(). Once the child process forked it does still have the fds (and they still show up in /proc/`pidof dbus-daemon`/fd/) but we're not getting SIGIO as changes are made to the configuration files.

The attached patch moves the initialization of the dnotify fds to process_config_postinit(). This is safe for all current code paths and solves the dnotify disfunction. If we're running dbus-daemon as daemon the fds for dnotify are now being initialized after fork() for the child process.

* configure.in: The current configure.in check for dnotify probes 'x\$target_os' for being 'xlinux-gnu'. I've changed the check to match for 'xlinux', too. Additionally I have adapted the configure option's style to match with the others.

2005-11-14 Robert McQueen <robot101@debian.org>

* python/decorators.py, python/service.py: Add a new argument to the dbus.service.method decorator called sender_keyword, which if set, specifies the name of an argument which will be provided the bus name of the method caller.

* test/python/test-client.py, test/python/test-service.py: Add a method and test to check the sender_keyword functionality.

2005-11-07 John (J5) Palmieri <johnp@redhat.com>

* bus/driver.c (bus_driver_handle_reload_config): Make sure we send an empty reply so blocking calls don't block forever (Patch from Sjoerd Simons <sjoerd at luon.net>)

* AUTHORS: Add Robert McQueen for his work on the Python Bindings and other parts of D-Bus

2005-11-07 Robert McQueen <robot101@debian.org>

* python/decorators.py: Change emit_signal function to use the signature annotation of the signal when marshalling the arguments from the service. Fix a bug where the code checking signature length against argument length referenced the wrong variable.

* python/introspect_parser.py: Avoid adding the type signature of signal arguments to any methods which occur after them in the introspection data (!) by making the parser a little more careful about its current state.

* python/service.py: Remove debug prints from last commit (again :D).

* test/python/test-client.py, test/python/test-service.py: Add test signals with signature decorators to test the strict marshalling code gives errors at the right time. Could do with checking the signals actually get emitted too, given that the test does nothing with

signals at the moment...

2005-11-07 Robert McQueen <robot101@debian.org>

* python/_dbus.py: Add WeakReferenceDictionary cache of dbus.Bus instances to stop madness of creating new instances representing the same bus connection all the time, rendering any tracking of match rules and bus names quite meaningless. Caught a bug where the private argument to SessionBus() and friends was being passed in as use_default_mainloop by mistake. Still some problems with multiple dbus_binding.Connection instances representing the same low-level connection (eg when you use both SessionBus() and StarterBus() in same process), but it's a lot better now than it was.

* python/dbus_bindings.pyx: Add constants with the return values for bus_request_name().

* python/service.py: Store bus name instances in a per-dbus.Bus cache and retrieve the same instances for the same name, so deletion can be done with refcounting. Also now throws some kind of error if you don't actually get the name you requested, unlike previously...

* test/python/test-client.py: Add tests for instance caching of buses and bus name objects.

2005-11-04 Robert McQueen <robot101@debian.org>

* python/dbus_bindings.pyx, test/python/test-client.py: Fix marshalling of boolean values. Add some booleans to the values in the test client.

* python/decorators.py, python/service.py: Add an 'async_callbacks' argument to the dbus.service.method decorator, which allows you to name arguments to take two callback functions for replying with return values or an exception.

* test/python/test-client.py, test/python/test-service.py: Add test case using asynchronous method reply functions, both return values and errors, and from within both the function itself and from a mainloop callback.

* python/decorators.py, python/service.py: Perform checking that the

number of method/signal arguments matches the number of types in the signature at class loading time, not when you first introspect the class.

* python/service.py: Remove debug print left by the last commit.

2005-11-03 Robert McQueen <robot101@debian.org>

* python/service.py: Heavy refactoring of method invocation, with hopefully no effect on functionality. Nuked `_dispatch_dbus_method_call` in favour of a new `_message_cb` that uses separate functions for looking up the method to call, marshalling the return values, and sending exceptions as errors, and is easier to follow as a consequence. Fixes some corner cases about returning things that don't match your declared `out_signature`, allows exceptions to define `_dbus_error_name` and have it be sent over the bus as the error name, and paves the way for cool stuff like heeding the message no reply flag, asynchronous method implementations, informing the method of the sender, and including backtraces in the error messages.

* test/python/test-client.py: Catch and print exceptions thrown in the async callback tests, rather than passing them to the low-level bindings to be ignored in a noisy and frustrating manner.

2005-11-03 Robert McQueen <robot101@debian.org>

* python/_dbus.py, python/proxies.py, python/service.py: Add `__repr__` functions to `dbus.Bus`, `dbus.service.BusName` and `dbus.service.Object`, tweak others to be consistent.

* test/python/test-client.py: Tweak output of `testInheritance`.

2005-10-29 Robert McQueen <robot101@debian.org>

* python/service.py: Major changes to allow multiple inheritance from classes that define D-Bus interfaces:

1. Create a new Interface class which is the parent class of `Object`, and make the `ObjectType` metaclass into `InterfaceType`.
2. Patch written with Rob Taylor to replace use of `method_vtable`

with code that walks the class's `__MRO__` (method resolution order) to behave like Python does when invoking methods and allow overriding as you'd expect. Code is quite tricky because we have to find two methods, the one to invoke which has the right name and isn't decorated with the /wrong/ interface, and the one to pick up the signatures from which is decorated with the right interface.

The same caveats apply as to normal multiple inheritance - this has undefined behaviour if you try and inherit from two classes that define a method with the same name but are decorated with different interfaces. You should decorate your overriding method with the interface you want.

3. Replace grungy introspection XML generation code in the metaclass with dictionaries that cope correctly with multiple inheritance and the overriding of methods. This also uses the signature decorations to provide correct introspection data, including the debut appearance of the types of your return values. :D

* test/python/test-client.py, test/python/test-service.py: Add a test case to try invoking an method that overrides one inherited from a D-Bus interface class.

2005-10-29 Robert McQueen <robot101@debian.org>

* python/dbus_bindings.pyx: Tweak 'raise AssertionError' to `assert()`.
Add checking for the end of struct character when marshalling a struct in `MessageIter.append_strict`.

* python/examples/example-service.py,
python/examples/gconf-proxy-service.py,
python/examples/gconf-proxy-service2.py: Update to use gobject mainloop directly rather than appearing to depend on gtk.

* python/test/test-client.py, python/test/test-server.py: Remove obsolete and broken test scripts for old bindings. We have up to date and working tests in test/python/.

2005-10-29 Robert McQueen <robot101@debian.org>

* python/decorators.py: Add optional arguments to the method and signal decorators to allow you to specify the signature of arguments

and return values. Preserve the doc strings of signal functions in the decorated version, for pydoc and friends.

* python/dbus_bindings.pyx, python/proxies.py: Replace the parse_signature_block function with an iterable dbus.Signature() type. Fix a bug in MessageIter.append_strict where you could append anything by claiming it was a string.

* python/service.py: Use the out_signature decoration on methods to marshal return values, meaning you no longer require dbus.Array() or dbus.Dictionary() to indicate the type when returning empty arrays or dictionaries. Fix a bug where exceptions which are defined in __main__ are not turned into error replies.

* test/python/test-client.py, test/python/test-service.py: Add test for correct marshalling of return values according to out_signature. Fix a bug in the async call test where the error_handler is missing a self argument.

2005-10-29 Robert McQueen <robot101@debian.org>

* glib/Makefile.am, glib/examples/Makefile.am, glib/examples/statemachine/Makefile.am: Merge patch from Ubuntu by Daniel Stone to replace explicit calls to libtool with \$(LIBTOOL).

* test/python/.cvsignore: Add run-with-tmp-session-bus.conf.

* tools/dbus-monitor.1, tools/dbus-monitor.c: Merge dbus-monitor patch from Ubuntu by Daniel Silverstone to allow specifying match rules on the command line.

2005-10-27 Ross Burton <ross@openedhand.com>

* dbus/dbus-marshal-header.c:
Remove dead code.

* glib/dbus-gobject.c:
Stop compiler warning.

2005-10-25 Ross Burton <ross@openedhand.com>

- * dbus/dbus-auth.c:
- * dbus/dbus-server-unix.c:
- * dbus/dbus-transport-unix.c:
- * glib/dbus-gmain.c:
- * glib/dbus-gobject.c:

Add some const keywords.

2005-10-25 Ross Burton <ross@openedhand.com>

- * doc/dbus-specification.xml:
Document the NoReply annotation.

- * glib/dbus-binding-tool-glib.h:
- * glib/dbus-binding-tool-glib.c:
Respect the NoReply annotation.

2005-10-24 Robert McQueen <robot101@debian.org>

- * python/dbus_bindings.pyx (String, MessageIter): make D-Bus strings derive from unicode instead of str, and encode/decode UTF-8 when marshalling/unmarshalling bus messages

- * python/introspect_parser.py: encode introspection data as UTF-8 before passing the buffer into libxml2

- * test/python/test-client.py: add unicode test strings

- * test/data/valid-service-files/.cvsignore,
test/python/.cvsignore:
ignore generated python test files

2005-10-17 John (J5) Palmieri <johnp@redhat.com>

- * glib/dbus-gvalue-utils.c (hash_free_from_gtype): handle gdouble and G_TYPE_VALUE_ARRAY (DBUS_TYPE_STRUCT) (gvalue_from_hash_value, hash_value_from_gvalue): handle gdouble

- * glib/dbus-gvalue.c (dbus_gvalue_to_signature): add missing DBUS_STRUCT_BEGIN_CHAR and DBUS_STRUCT_END_CHAR characters when constructing struct signatures

- * python/_dbus.py (Bus): handle private connections using the private keyword in the constructor. defaults to private=False (Bus::close): new method to close a connection to the bus

- * python/dbus_bindings.pyx (Connection::close): renamed method was previously called disconnect (bus_get): now supports getting a private connection

- * python/proxies.py (ProxyMethod::__call__): check if ignore_reply

keyword is set to True. if it is, execute the method without waiting
for a reply
(ProxyObject::_introspect_execute_queue): new method for executing
all the pending methods that were waiting for the introspect to finish. this is called when introspect either succeeds or fails
(ProxyObject::_introspect_error_handler): call queued methods

2005-10-14 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx (MessageIter::append_strict): check for
STRUCT_BEGIN not TYPE_STRUCT in indicate we are marshalling a struct

* python/service.py (Object::_message_cb): handle exceptions correctly
by sending them over the wire to the calling app. This makes sure
the client returns immediately instead of waiting the 15 seconds to
timeout.

* test/python/test-client.py
(TestDBusBindings::testBenchmarkIntrospect):
Add a test to benchmark how long it takes to introspect a service and
call a method which returns a large element (pretty fast)

* test/python/test-service.py (TestObject::GetComplexArray): new test
method which pushes a lot of data

2005-10-13 John (J5) Palmieri <johnp@redhat.com>

* python/service.py(ObjectType::_reflect_on_signal, _reflect_on_method):
reclaim memory outside of the loop and use del instead of just setting
the key to None

2005-10-13 John (J5) Palmieri <johnp@redhat.com>

* python/service.py (ObjectType::_reflect_on_signal): Always close
signal tag even when there are no arguments

2005-10-13 John (J5) Palmieri <johnp@redhat.com>

* configure.in: Set mono, mono-docs and Qt3 to default to no instead of auto when building. These bindings do not

have full time maintainers and will not be supported for the 1.0 release.

2005-10-12 John (J5) Palmieri <johnp@redhat.com>

patches from Michael Krivoruchko <misha at sun.com>:

- * dbus/dbus-connection.c
(_dbus_connection_queue_received_message_link,
_dbus_connection_message_sent,
_dbus_connection_send_preallocated_unlocked_no_update,
_dbus_connection_pop_message_link_unlocked): handle the case when
path
is NULL when calling _dbus_verbose
- * configure.in: check for functions getpeerucred and getpeereid
- * dbus/dbus-sysdeps.c (_dbus_read_credentials_unix_socket):
provides
support of auth EXTERNAL on Solaris 10+ (getpeerucred), FreeBSD
4.6+,
OpenBSD 3.0+ and FreeBSD 5.0+ as well as MacOSX 10.2+
(getpeereid).
Patch was only tested on Solaris 10 x86 so it might be issues
with other platforms (i.e. BSDs and MacOSX)

2005-10-05 John (J5) Palmieri <johnp@redhat.com>

- * glib/dbus-gvalue.c (marshal_variant): call _dbus_gvalue_marshal
instead of marshal basic so we can handle recursive types in a
variant
- * test/glib/test-dbus-glib.c: Add test for marshaling recursive
types
in variants
- * test/glib/test-service-glib.c, test-service-glib.xml
(my_object_echo_variant [EchoVariant],
my_object_process_variant_of_array_of_ints123
[ProcessVariantOfArrayOfInts123]):
Add two test methods
- * python/introspect_parser.py: New module for parsing introspect
data.
- * python/dbus_bindings.pyx:
(various places): when throwing errors fix to use errmsg
instead
of message local variable because Pyrex can get confused with
other
message variables (initial patch by Robert McQueen)

<robert.mcqueen at collabora.co.uk>
(MessageIter::parse_signature_block): new method for getting the
next
block in a signature.
(MessageIter::append_strict): new method for appending values
strictly
using the passed in signature instead of guessing at the type
(MessageIter::append_dict, append_struct, append_array): use
signatures to marshal children if the signature is available

* python/exceptions.py (IntrospectionParserException): new
exception

* python/proxies.py (ProxyMethod::__call__): Marshal args with
introspected signatures if available, else we fall back to the
old way of doing things.
(ProxyObject::__introspect_reply_handler): parse introspection
data

* python/service.py (ObjectType::__reflect_on_method): Properly
terminate <method> if there are no args in the reflection data

* test/python/test-client.py: add tests for talking with the GLib
test server. This gives us better coverage for introspection
since
python to python will always generate arguments as variants. It
also
allows us to test the robustness of the GLib bindings and
interlanguage
communications.

2005-10-03 John (J5) Palmieri <johnp@redhat.com>

* bus/driver.c (bus_driver_handle_introspect): Add signals
to the introspect data. (patch from Daniel P. Berrange
<dan at berrange.com>)

* bus/dispatch.c (check_existent_ping): Add testcase for Ping

* dbus/dbus-connection.c (_dbus_connection_peer_filter,
_dbus_connection_run_builtin_filters): Changed these to
be unlock_no_update functions and call
_dbus_connection_send_unlocked_no_update instead of
dbus_connection_send to avoid locking errors.

* doc/TODO: Removed the make Ping test TODO

2005-09-26 John (J5) Palmieri <johnp@redhat.com>

* dbus/Python.pyx: Fixed memory leaks when throwing errors.
We now copy the message from a DBusError and then free

the error object before throwing the error

* glib/dbus-glib-tool.c: removed extra comma at the end of the DBusBindingOutputMode enum which was causing a warning.
#include <time.h> so using time_t is explicitly defined

2005-09-26 John (J5) Palmieri <johnp@redhat.com>

* Integrate patches from Lennart Poettering <mzsqb@openpioneer.de>:
- dbus/dbus-bus.c
(internal_bus_get): new method that takes over the heavy lifting of dbus_bus_get and adds the ability to get a private connection to the bus
(dbus_bus_get): wrapper to internal_bus_get that provides the same interface as in previous versions
(dbus_bus_get_private): new method that is a wrapper to internal_bus_get to get a private connection to the bus
- dbus/dbus-bus.h
(dbus_bus_get_private): add as a public libdbus interface
- dbus-1.pc.in: output system_bus_default_address and sysconfdir variables so apps can use them when compiling

2005-09-23 Harald Fernengel <harry@kde.org>

* dbus/qt: New Qt bindings

2005-09-12 Waldo Bastian <bastian@kde.org>

* dbus/dbus-marshall-validate.c,
doc/dbus-specification.xml, test/Makefile.am,
test/test-names.c: allow hyphens in bus names.

2005-09-11 Mark McLoughlin <mark@skynet.ie>

* test/data/auth/fallback.auth-script: we don't retry the EXTERNAL method when we know it's going to fail anymore.

2005-09-11 Mark McLoughlin <mark@skynet.ie>

* dbus/dbus-connection-internal.h: rename (add|remove|toggle)_(watch|timeout) to unlocked()

* dbus/dbus-connection.c: ditto.

* dbus/dbus-timeout.c, dbus/dbus-transport-unix.c: Update some callers for the renaming.

2005-09-10 Mark McLoughlin <mark@skynet.ie>

- * dbus/dbus-auth.c: (record_mechanisms): don't retry the first auth mechanism because we know we're just going to get rejected again.
- * dbus/dbus-keyring.c: (_dbus_keyring_reload): Fix thinko ... and what a nasty little bugger to track down you were ...
- * dbus/dbus-connection.c:
 - (_dbus_connection_add_watch),
 - (_dbus_connection_remove_watch): add note about these needing the connection to be locked.
 - (_dbus_connection_get_dispatch_status_unlocked): set status to DATA_REMAINS when we queue the disconnected message.
- * bus/dispatch.c:
 - (bus_dispatch): fix warning.
 - (check_existent_service_no_auto_start): Expect ChildSignaled error too.
 - (check_existent_hello_from_self): fix another couple of warnings.

2005-09-08 Joe Shaw <joeshaw@novell.com>

Patches from James Willcox <snorp@snorp.net>

- * mono/Makefile.am: Add Int16.cs and UInt16.cs
- * mono/DBusType/Array.cs: Handle multidimensional arrays, and support array "out" parameters.
- * mono/DBusType/Int16.cs, mono/DBusType/UInt16.cs: New files, for 16-bit int support.

2005-09-06 John (J5) Palmieri <johnp@redhat.com>

- * Released 0.50
- * Patch from Steve Grubb:
 - bus/activation.c (bus_activation_service_reload_test): clean up some indentation
 - dbus/dbus-keyring.c (_dbus_keyring_reload): fix conditional
 - dbus/dbus-message-factory.c (generate_special): fix a couple of buffer overflows in the test suite. This is non critical because it can not be exploited and this code is only run when doing a make check.
- * Patch from Yaakov Selkowitz: Build fixes for Cygwin
 - configure.in: Don't check and link against kdec core, only qt headers

- dbus/Makefile.am: Add -no-undefined to libdbus_1_la_LDFLAGS
- gcj/org/freedesktop/dbus/Makefile.am:
add libdbus_gcj_1_la_LDFLAGS = -no-undefined
- glib/Makefile.am: Add -no-undefined to
libdbus_glib_1_la_LDFLAGS
and \$(DBUS_GLIB_LIBS) to dbus_binding_tool_LDADD
- qt/Makefile.am: Add -no-undefined to libdbus_qt_1_la_LDFLAGS
- tools/Makefile.am: Add platform extensions to binaries
(i.e. .exe on windows)

* configure.in:

- Make it so if no suitable version of python is found we only
disable building python instead of exiting the configure script
- Require version 2.4 of glib for glib bindings
- Up version to 0.50

* python/__init__.py: Sync version with libdbus to (0,50,0)

2005-09-05 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-object-tree.c (find_subtree_recurse):
a couple of optimizations (bug #710):

- do a binary search in the tree
- insert a new child at the right place directly, no need for
qsort anymore
- do the "double alloc" thing when allocating children

2005-08-31 John (J5) Palmieri <johnp@redhat.com>

* python/Makefile.am: Break on pyrex errors instead of ignoring
them

* python/dbus_bindings.pyx: Memory management foo
(global): remove hacky_user_data_references global list
(GIL_safe_cunregister_function_handler): userdata now stuffed
into
tuples. Unref user_data
(GIL_safe_cmessage_function_handler): userdata now stuffed into
tuples
(Connection::__del__): Remove and replace with __dealloc__ method
(Connection::add_filter): Stuff user_data into a tuple. Use
Py_INCREF
to keep tuple from being deallocated instead of the global var
hack
(Connection::register_object_path): Stuff user_data into a tuple.
Use Py_INCREF to keep tuple from being deallocated instead of the
global var hack
(Connection::register_fallback): Stuff user_data into a tuple.
Use Py_INCREF to keep tuple from being deallocated instead of the
global var hack
(GIL_safe_pending_call_notification): Don't unref the message
because it gets unreffed when going out of scope. Py_XDECREF

the user_data
(PendingCall::__del__): Remove and replace with __dealloc__
method
(PendingCall::set_notify): ref the pending call because we will
need it to stick around for when the notify callback gets called
(Message::__del__): Remove and replace with __dealloc__ method

- * python/dbus_glib_bindings.pyx (init_gthreads): Changed to
gthreads_init to match up with the dbus call
- * python/glib.py (init_threads): Changed to threads_init to match
up with gobject.threads_init(). init_threads is kept for
backwards
compat but will most likely be deprecated in the future
- * test/python/test-client.py:
 - revamp to use Python's unittest functionality
 - add async call tests
 - setup threads in glib and dbus so we make sure locks are
working

2005-08-30 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx
(_pending_call_notification, cunregister_function_handler,
cmessage_function_handler): All callback functions have been
rearranged
to workaroud a bug in Pyrex when working with the GIL which is
Python's
global lock when dealing with threads. They have been split into
a wrapper function (which assumes the name of the old function)
and
a _GIL_safe_<function name> function which contains the
functionality
of the old function. This ensures that Pyrex does not write code
the lock is released.

2005-08-30 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx (_pending_call_notification): Obtain
the
GIL global lock when calling back into Python

2005-08-29 John (J5) Palmieri <johnp@redhat.com>

- * Release 0.36.2
- * Add Havoc's patch that never got applied to HEAD (Bug #2436):
- * bus/policy.c (bus_policy_allow_user): change default "user is
allowed" to be "user has same uid as the bus itself"; any

allow/deny rules will override.

* bus/session.conf.in: don't allow all users, since now by default the user that ran the bus can connect.

2005-08-26 Colin Walters <walters@verbum.org>

* tools/dbus-print-message.c (print_message): Flush stdout after printing a message, so that redirecting to a file, then hitting Ctrl-C works.

2005-08-25 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx: Tracked down a major memleak and fixed it
(EmptyMessage): new class that subclasses Message. This is a workaround to a Pyrex bug that fails to call `__del__` when the Message object goes out of scope. For some reason subclassing Message fixes this bug
(Bus::send_with_reply_and_block): use EmptyMessage instead of Message
- s/Message(_create=0)/EmptyMessage everywhere else

* test/python/test-{server|client}.py: add the python/.libs directory to the lookup path so dbus_bindings and dbus_glib_bindings don't get picked up from the system

2005-08-25 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (dbus_g_proxy_call): Doc update, thanks to Ryan Lortie for the suggestion.

2005-08-24 John (J5) Palmieri <johnp@redhat.com>

* test/python: Add python regression test
* configure.in: Add test/python/Makefile
* test/Makefile.am: Add the python directory to SUBDIRS

2005-08-24 John (J5) Palmieri <johnp@redhat.com>

* Release 0.36.1
* python/_dbus.py:
(Interface::connect_to_signal): propagate keywords for match on args
(Bus::add_signal_receiver): Fix typos
s/dbus_inteface/dbus_interface

* python/proxies.py (ProxyObject::connect_to_signal):
propagate keywords for match on args

* Makefile.am: point everything to pyexecdir since python borks
on multilib

2005-08-23 John (J5) Palmieri <johnp@redhat.com>

* Release 0.36

2005-08-23 Colin Walters <walters@verbum.org>

* test/glib/Makefile.am: Don't multiply-define EXTRA_DIST.

2005-08-23 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_glib_bindings.pyx: reorder imports and c
definitions
to fix some warnings. We now use dbus_bindings.DBusConnection
instead
of defining DBusConnection ourselves.

2005-08-18 John (J5) Palmieri <johnp@redhat.com>

* python/dbus.pth: New path file to fix up problems when
installing
c libraries to lib64 and python files to lib.

* python/Makefile.am: install dbus.pth in the correct spot

2005-08-17 John (J5) Palmieri <johnp@redhat.com>

* ChangeLog: clean up my last entry a bit

* doc/introspect.xsl: New stylesheet for converting introspection
data
into browser renderable xhtml. Contributed by Lennart Poettering.

* doc/introspect.dtd: Fixups in the introspect format from
Lennart
Poettering.

* doc/dbus-tutorial.xml:

- Add Colin Walter to the Authors section for authoring the GLib
section
- Add descriptions of the new signature and type functionality
in the Python complex type mapping section
- Add a sidenote on the new args matching functionality in
the Python bindings
- Fixed up some of the examples to use the GObject.MainLoop
instead of gtk.main

```
* python/_dbus.py:
(Bus::_create_args_dict): New. Converts a hash of arg matches
to a more useable format
(Bus::_add_signal_receiver): add a **keywords parameter for
catching
arg match parameters
(Bus::_remove_signal_receiver): add a **keywords parameter for
catching
arg match parameters

* python/matchrules.py:
(MatchTree::_exec_matches): Check for arg matches
(SignalMatchRule::_add_args_match): New method
(SignalMatchRule::_execute): Added args_list parameter as an
optimization
so we don't have to marshal the args more than once
(SignalMatchRule::_match_args_from_list): New method that checks
to see
if the rule's arg matches match an argument list. Only arguments
set in the rule are checked.
(SignalMatchRule::_match_args_from_rule): New method that checks
to see
if the rule's arg matches match another rule's. All args have to
match
in order for this method to return true. If either rule has more
args
then it is not a match.
(SignalMatchRule::_is_match): Add args match
(SignalMatchRule::_repr): Add args to the final output if they
exist
```

2005-08-17 Ross Burton <ross@burtonini.com>

```
* glib/dbus-gproxy.c:
(dbus_g_proxy_call_no_reply): unref the message once sent.
(dbus_g_proxy_call): protect against NULL proxy.
```

2005-08-16 John (J5) Palmieri <johnp@redhat.com>

```
* python/__init__.py: Version updated (0, 43, 0)

* python/dbus_bindings.pyx:
- Fixed type objects to have self passed into __init__
- Added the Variant type
- Add the ability to specify types or signatures for Array,
Variant
and Dictionary
(Connection::_send_with_reply_handlers): return a PendingCall
object
(_pending_call_notification): handle the case when an error is
returned
without an error message in the body
```



```

    (MessageIter::get_boolean): return True or False instead of an
integer
    (MessageIter::python_value_to_dbus_sig): add direct checking of
types
    and add checks for objects with embeded signatures or types
(Array,
    Variant and Dictionary)
    (MessageIter::append_byte): handle case when the value is a
dbus.Byte
    (MessageIter::append_dict): handle embeded types or signatures
    (MessageIter::append_array): handle embeded types or signatures
    (MessageIter::append_variant): new method

* python/proxies.py:
    (DeferredMethod): New. Dummy executable object used when queuing
calls
    blocking on introspection data
    (ProxyMethod::__call__): add the timeout keyword for specifying
longer
    or shorter timeouts for method calls
    (ProxyObject): Add first pass at an introspection state machine
    (ProxyObject::__init__): Add introspect keyword for turing off an
on
    introspection.
    (ProxyObject::_Introspect): Internal Introspect call that
bypasses
    the usual mechanisms for sending messages. This is to avoid a
deadlock
    where the Intospect call would be queued waiting for the
Introspect
    call to finish ;-)
    (ProxyObject::_introspect_reply_handler): New. This method is
called
    when introspection returns with no error
    (ProxyObject::_introspect_error_handler): New. This method is
called
    when introspection encounters an error
    (ProxyObject::__getattr__): Code to handle different
introspection
    states. Queue async calls or block blocking calls if we are
introspecting. Pass through as normal if we are not or are done
with
    introspecting.

* python/service.py: Import signal and method from decorators.py

* python/types.py: Add Variant type

```

2005-08-16 Colin Walters <walters@verbum.org>

```

* glib/dbus-gobject.c (dbus_set_g_error): Don't lose if the
DBusError message is NULL.

```

2005-08-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-errors.c: apply patch from Timo Teras to make a malloc'd copy of the name parameter

2005-08-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_set_reply_serial): print warning if the reply serial is set to 0

2005-08-04 Colin Walters <walters@verbum.org>

```
* glib/dbus-gvalue-utils.h
(_dbus_g_type_specialized_builtins_init)
  (dbus_g_type_is_fixed, dbus_g_type_fixed_get_size)
  (dbus_gvalue_set_from_pointer, dbus_g_hash_table_value_foreach)
  (dbus_g_hash_table_insert_values,
dbus_g_hash_table_insert_steal_values)
  (dbus_gtype_is_valid_hash_key, dbus_gtype_is_valid_hash_value)
  (dbus_g_hash_func_from_gtype, dbus_g_hash_free_from_gtype)
  (dbus_g_hash_equal_from_gtype, dbus_gvalue_stor,
dbus_gvalue_take):
* glib/dbus-gvalue.h (dbus_g_value_types_init)
  (dbus_gvalue_demarshal, dbus_gvalue_demarshal_variant)
  (dbus_gvalue_demarshal_message, dbus_gvalue_marshal):
```

Prefix name with _ to ensure they're not exported. All callers updated.

```
* glib/dbus-gvalue.c (typecode_to_gtype)
  (dbus_typecode_maps_to_basic, basic_typecode_to_gtype)
  (signature_iter_to_g_type_dict)
  (signature_iter_to_g_type_array)
  (dbus_gtype_from_signature_iter, dbus_gtype_from_signature)
  (dbus_gtypes_from_arg_signature):
Move to dbus-gsignature.c.
```

```
* glib/dbus-binding-tool-glib.c
(dbus_binding_tool_output_glib_server): Call
  dbus_g_type_specialized_builtins_init instead of
dbus_g_value_types_init.
  (dbus_binding_tool_output_glib_client): Ditto.
```

* glib/Makefile.am (DBUS_GLIB_INTERNALS): Add dbus-gsignature.c and dbus-gsignature.h

* test/glib/test-service-glib.c (my_object_rec_arrays): Delete unused variable.

2005-08-03 Colin Walters <walters@verbum.org>

* glib/dbus-gobject.c: Add tests on hardcoded object info; this should catch any incompatible changes accidentally made.

2005-08-03 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_read_credentials_unix_socket): fix typo, from Julien Puydt

* bus/connection.c (bus_connection_disconnected): we were always doing a wait_for_memory due to a buggy loop, found by Timo Hoenig

2005-08-01 Colin Walters <walters@verbum.org>

Patch from Joe Markus Clarke:

* glib/dbus-gidl.c (property_info_unref, arg_info_unref): Fix use-after-free.

2005-08-01 Colin Walters <walters@verbum.org>

Patch from Joe Markus Clarke:

* tools/dbus-send.c (main):

Don't use C99 style initializers (bug #3933).

2005-08-01 Colin Walters <walters@verbum.org>

Patch from Joe Markus Clarke:

* glib/dbus-gvalue.c (dbus_g_value_types_init):
* glib/dbus-gvalue-utils.c
(dbus_g_type_specialized_builtins_init)
* glib/dbus-gobject.c (write_interface):

Don't use C99 style initializers (bug #3933).

2005-07-31 Havoc Pennington <hp@redhat.com>

* tools/dbus-viewer.c (load_child_nodes): fix invocation of dbus_g_proxy_call, fix from Piotr Zielinski bug #3920

2005-07-30 Havoc Pennington <hp@redhat.com>

* fix a bunch of Doxygen warnings and mistakes

2005-07-30 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_string_parse_uint): remove #ifdef DBUS_BUILD_TESTS since it's now used in production code

2005-07-29 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c (write_junk): initialize the junk buffer so valgrind doesn't have a breakdown

2005-07-29 Havoc Pennington <hp@redhat.com>

* bus/signals.c (bus_signals_test): add match_rule_equal() tests (match_rule_matches): remove unused arg (test_matching): add tests for match_rule_matches()

* bus/signals.c (bus_match_rule_parse_arg_match): add ability to do arg0='foo' arg5='bar' in the match rules (match_rule_matches): don't match if the arg0='foo' doesn't match.

* dbus/dbus-protocol.h (DBUS_MAXIMUM_MATCH_RULE_ARG_NUMBER): add this

2005-07-29 Ross Burton <ross@openedhand.com>

* dbus/dbus-connection.c:
Don't create a DBusCondVar which is never used.

2005-07-27 Ross Burton <ross@openedhand.com>

* dbus/dbus-message.c:
Reduce the size of the maximum cached message to 10K.

2005-07-25 Ross Burton <ross@openedhand.com>

* glib/dbus-gproxy.c:
Remove matches when all proxies are unregistered.

2005-07-24 Colin Walters <walters@verbum.org>

* glib/dbus-gvalue.c (signature_iter_to_g_type_array): Don't require typedata; recursive arrays won't have it.

* test/glib/test-dbus-glib.c:
* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml: Add recursive arrays tests.

2005-07-20 John (J5) Palmieir <johnp@redhat.com>

* python/_dbus.py, _util.py, decorators.py, extract.py, matchrules.py, proxies.py, service.py: Cleanup of code after running it through the pyflakes code checker mostly dealing with undefined names. (Bug #3828, Patch from Anthony Baxter <anthony@interlink.com.au>)

2005-07-17 John (J5) Palmieri <johnp@redhat.com>

* NEWS: Update to 0.35.2

2005-07-17 John (J5) Palmieri <johnp@redhat.com>

* python/_dbus.py: Remove import of the dbus.services module as it no longer exists (patch from Dimitur Kirov)

* python/service.py (Object::__init__): Fixed typo s/name/bus_name (patch from Dimitur Kirov)

* python/examples/example-signal-emitter.py: import dbus.glib to get the main loop and use glib mainloop instead of gtk so X doesn't have to be running.

* python/examples/example-signal-recipient.py: import dbus.glib to get the main loop and use glib mainloop instead of gtk so X doesn't have to be running. Import the decorators module directly.

* test/glib/Makefile.am: Added DIST_EXTRA files that distcheck didn't pick up on but are needed to build

* configure.in: upped version to 0.35.2

* bus/driver.c, bus/selinux.c, bus/selinux.h, dbus/dbus-protocol.h:

added Colin Walters' SELinux API rename patch from head s/unix security context/selinux security context/

2005-07-16 John (J5) Palmieri <johnp@redhat.com>

* python/Makefile.am: dbus_binding.pxd.in should be included in EXTRA_DIST not dbus_binding.pxd
fix up \$(srcdir) hopefully for the last time

* NEWS: Update to 0.35.1

2005-07-16 Colin Walters <walters@verbum.org>

* bus/driver.c
(bus_driver_handle_get_connection_selinux_security_context): Renamed from bus_driver_handle_get_connection_unix_security_context.

Update for

error usage.

(message_handlers): Update for renames.

* bus/selinux.c (bus_selinux_allows_send): Handle OOM on _dbus_string_init failure correctly.

(bus_selinux_append_context): Convert SID to context. Append it

as a byte array.
(bus_selinux_shutdown): Handle the case where
bus_selinux_full_init
hasn't been called.

* bus/selinux.h: Update prototype.

* dbus/dbus-protocol.h
(DBUS_ERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN): Renamed
from DBUS_ERROR_UNIX_SECURITY_CONTEXT_UNKNOWN.

2005-07-15 Colin Walters <walters@verbum.org>

* doc/TODO: Add note about convenience wrappers.

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

* NEWS: Update to 0.35

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

* glib/Makefile.am: Add make-dbus-glib-error-switch.sh to
EXTRA_DIST
so distcheck doesn't fail

* glib/examples/Makefile.am: Add example-service.xml and
example-signal-emitter.xml to EXTRA_DIST so distcheck doesn't
fail

* glib/examples/statemachine/Makefile.am: Add statemachine.xml
and
statemachine-server.xml to EXTRA_DIST so distcheck doesn't fail

* python/Makefile.am: Prepend \$(srcdir)/ to source files so the
compiler looks in the right places during distcheck

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

* glib/example/Makefile.am: Fix a typo which cause make distcheck
to fail

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

* python/examples/example-service.py,
python/examples/example-signal-emitter.py: Fixed up examples
for API changes

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

* python/__init__.py: Upped to version (0,42,0) because of
the API change

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

- * ChangeLog: fix date in last entry
- * configure.in, bus/system.conf.in: add the ability to configure the system bus user at compiletime with the --with-dbus-user flag (patch from Kristof Vansant)

2005-07-15 John (J5) Palmieri <johnp@redhat.com>

- * bus/dispatch.c, test/test-service.c: Add testcase for sending messages to oneself (TODO item).
- * python/service.py (class Object): Swap ordering of bus_name and object_path parameters to better support inheritance.
- * doc/dbus-tutorial.xml: change Python docs to reflect change in parameter ordering and fix the inheritance section.
- * doc/TODO: remove sending message to oneself TODO item

2005-07-15 Ross Burton <ross@openedhand.com>

- * glib/dbus-gproxy.c:
Fix a leak when calling methods via the proxy.

2005-07-15 Colin Walters <walters@verbum.org>

- * bus/selinux.c (bus_selinux_append_context): Wrap in HAVE_SELINUX.

2005-07-14 John (J5) Palmieri <johnp@redhat.com>

- * python/_dbus.py (Bus::remove_signal_receiver):
don't add a callback to the match if none has been passed in
- * python/matchrules.py (SignalMatchTree::remove): if the rule being matched does not have a callback treat it as a wildcard fix matching logic
- * doc/dbus-tutorial.xml: Add Python tutorial

2005-07-14 Colin Walters <walters@verbum.org>

- * bus/driver.c
(bus_driver_handle_get_connection_unix_security_context): New function.
(message_handlers): Add.
- * bus/selinux.c (bus_selinux_append_context): New function; appends security context to message.

* bus/selinux.h: Prototype.

* dbus/dbus-protocol.h
(DBUS_ERROR_UNIX_SECURITY_CONTEXT_UNKNOWN): New.

2005-07-14 John (J5) Palmieri <johnp@redhat.com>

* bus/activation.c: clean up all tabs to be 8 spaces
(bus_activation_activate_service): make sure we clean up
if activation fails

* bus/dispatch.c: clean up all tabs to be 8 spaces
(check_shell_fail_service_auto_start): New function
tests to make sure we get fail properly when trying to auto start
a service
with a faulty command line
(check_shell_service_success_auto_start): New function tests to
make sure
auto started services get the arguments on the command line

* test/test-shell-service.c: Added service for testing auto-
starting with
command line arguments

* test/data/valid-service-files/debug-shell-echo-fail.service.in,
test/data/valid-service-files/debug-shell-echo-
success.service.in:
Added service files for testing auto-starting with command line
arguments

* */.cvsignore: added a bunch of generated files to various
.cvsignore files

2005-07-14 Rodrigo Moya <rodrigo@novell.com>

* dbus/dbus-shell.[ch]: copy/pasted code from GLib.

* dbus/Makefile.am: added new files to build.

* bus/activation.c (bus_activation_activate_service): support
activation commands with parameters.

* test/shell-test.c: added test program for the shell parsing
code.

2005-07-13 David Zeuthen <davidz@redhat.com>

* tools/dbus-send.c (append_arg, type_from_name): Also support 16
and
64 bit signed and unsigned parameters

2005-07-13 John (J5) Palmieri <johnp@redhat.com>

- * python/.cvsignore: remove dbus_bindings.pyx, add dbus_bindings.pxd

- * python/service.py (class Name): renamed BusName to make it clearer what the object is for (a name on the bus)

- * python/examples/example-service.py, python/examples/example-signal-emitter.py: change the Name object to BusName

2005-07-12 Colin Walters <walters@verbum.org>

Patch from Jim Gettys <Jim.Gettys@hp.com>.

- * tools/dbus-launch.c: Include sys/select.h.

2005-07-12 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx.in: removed

- * python/dbus_bindings.pyx: Added.
 - Fixed some memleaks (patch from Sean Meiners <sean.meiners@linspireinc.com>)
 - Broke out the #include "dbus_h_wrapper.h" and put it in its own pxd file (Pyrex definition)
 - Broke out glib dependancies into its own pyx module

- * python/dbus_bindings.pdx: Added.
 - Defines C class Connection for exporting to other modules

- * python/dbus_glib_bindings.pyx: Added.
 - New module to handle lowlevel dbus-glib mainloop integration

- * python/glib.py: Added.
 - Registers the glib mainloop when you import this module

- * python/services.py: Removed (renamed to service.py)

- * python/service.py: Added.
 - (class Server): renamed Name

- * python/__init__.py: Bump ro version (0,41,0)
 - don't import the decorators or service module by default. These now reside in the dbus.service namespace

- * python/_dbus.py (Bus::__init__): Add code run the main loop setup function on creation

- * python/examples/example-service.py,

python/examples/example-signal-emitter.py: update examples

* python/examples/gconf-proxy-service.py,
python/examples/gconf-proxy-service2.py: TODO fix these up

* doc/TODO: Addition

- Added a Python Bindings 1.0 section

- added "Add match on args or match on details to match rules"

2005-07-12 Colin Walters <walters@verbum.org>

* glib/examples/statemachine/Makefile.am (statemachine-server-glue.h)

(statemachine-glue.h):

* glib/examples/Makefile.am (example-service-glue.h)

(example-signal-emitter-glue.h):

* glib/Makefile.am (dbus-glib-error-switch.h):

Add libtool --mode=execute so we use the built library instead of any installed one.

2005-07-11 Colin Walters <walters@verbum.org>

* glib/dbus-gvalue.c (struct _DBusGValue): Delete.

(dbus_g_value_types_init): Remove assertion.

(dbus_g_value_get_g_type, dbus_g_value_open)

(dbus_g_value_iterator_get_values, dbus_g_value_get_signature)

(dbus_g_value_copy, dbus_g_value_free): Delete unimplemented functions related to DBusGValue. Now we marshal/demarshal structures as GValueArray.

(dbus_gtype_from_signature_iter): Return G_TYPE_VALUE_ARRAY for structures.

(signature_iter_to_g_type_array): Don't call signature_iter_to_g_type_struct.

(signature_iter_to_g_type_struct): Delete.

(dbus_gvalue_to_signature): Delete.

(dbus_gvalue_to_signature): New function with same name as other one; we can convert structures to signatures.

(demarshal_valuearray): New function.

(get_type_demarshaller): Use it.

(demarshal_recurse): Delete.

(marshal_proxy): New function.

(marshal_map): Warn if we can't determine signature from type.

(marshal_collection_ptrarray): Ditto.

(marshal_collection_array): Ditto.

(get_type_marshaller): Use marshal_valuearray.

(marshal_recurse): Delete.

(_dbus_gvalue_test): Add some tests.

* dbus/dbus-glib.h (struct _DBusGValueIterator):

(dbus_g_value_get_g_type, DBUS_TYPE_G_VALUE)

(dbus_g_value_open, dbus_g_value_iterator_get_value)

(dbus_g_value_iterator_get_values, dbus_g_value_iterator_recurse)
(dbus_g_value_free): Remove prototypes.

* glib/dbus-binding-tool-glib.c
(dbus_g_type_get_lookup_function): Handle
G_TYPE_VALUE_ARRAY.

* glib/examples/example-service.c:
* glib/examples/example-client.c: Implement GetTuple.

* test/glib/test-dbus-glib.c:
* test/glib/test-service-glib.c:
* test/glib/test-service-glib.xml: Add structure tests.

2005-07-10 Colin Walters <walters@verbum.org>

* doc/TODO: Knock off some GLib items with this patch.

* glib/dbus-gvalue-utils.c (_dbus_gtype_can_signal_error)
(_dbus_gvalue_signals_error): New functions.

* glib/dbus-gvalue-utils.h: Prototype them.

* glib/dbus-gobject.c (arg_iterate): Update to handle return vals
and change to not output const/retval flags for input args. All
callers updated.
(invoke_object_method): Refactor to handle return values. Add
some more comments in various places. Remove debug g_print.

* glib/dbus-binding-tool-glib.h (DBUS_GLIB_ANNOTATION_RETURNVAL):
New.

* glib/dbus-binding-tool-glib.c (dbus_g_type_get_marshall_name):
Handle G_TYPE_NONE.
(compute_gsignature): New function; refactored from code from
compute_marshall and compute_marshall_name. Enhance to
handle return values and async ops more cleanly. Update for
async ops returning NONE instead of BOOLEAN.
(compute_marshall, compute_marshall_name): Call
compute_gsignature
and output appropriate string.
(generate_glue): Handle return value annotation. Also don't dump
constness flag for input arguments.

* glib/Makefile.am (DBUS_GLIB_INTERNALS): New variable; contains
files shared between installed library and utilities.
(libdbus_glib_1_la_SOURCES): Move some stuff into
DBUS_GLIB_INTERNALS.
(libdbus_gtool_la_SOURCES): Suck in DBUS_GLIB_INTERNALS so the
binding tool can access gtype utility functions.

* test/glib/test-service-glib.c:

- * test/glib/test-service-glib.xml:
- * test/glib/test-dbus-glib.c: Add some tests for return values.

2005-07-09 Colin Walters <walters@verbum.org>

- * glib/dbus-gparser.c (parse_annotation): Add annotations to argument if available, not method.
- * glib/dbus-gobject.c (arg_iterate): More verbose warnings. (invoke_object_method): First, remove some redundant GValues (object_value, error_value) in favor of working on array directly. Second, rework constness to be less buggy. Now we iterate directly over the argument metadata instead of parallel iterating over output signature and metadata.
- * glib/dbus-glib-tool.h: Add INVALID_ANNOTATION error.
- * glib/dbus-binding-tool-glib.c (generate_glue): Barf on const annotation on input args.

2005-07-09 Colin Walters <walters@verbum.org>

- * glib/dbus-binding-tool-glib.h (DBUS_GLIB_ANNOTATION_CONST): Define.
- * glib/dbus-binding-tool-glib.c (generate_glue): Handle Const annotation.
- * glib/dbus-gobject.c (arg_iterate): Update to parse constval too. (method_dir_signature_from_object_info): Handle arg_iterate change. (write_interface): Ditto. (lookup_object_info): Don't barf if format_version is > 0. (invoke_object_method): Handle arg constness.
- * glib/dbus-gidl.c (struct ArgInfo): Add annotations. (arg_info_new): Create. (arg_info_unref): Destroy. (arg_info_get_annotations, arg_info_get_annotation) (arg_info_add_annotation): New functions.
- * glib/dbus-gidl.h: Prototype them.
- * glib/dbus-gparser.c (parse_annotation): Allow annotations in args, disallow them in properties. (parse_annotation): Handle arg annotations.
- * test/glib/test-service-glib.xml:
- * test/glib/test-service-glib.c: Update to make some methods const.

2005-07-08 Colin Walters <walters@verbum.org>

- * test/glib/test-service-glib.xml:
- * test/glib/test-service-glib.c:
- * test/glib/test-dbus-glib.c: Test a{sv}.

- * glib/examples/statemachine/statemachine.c:
- * glib/examples/statemachine/statemachine-server.c:
- * glib/examples/statemachine/statemachine-client.c: Fix some bugs,
add progress bar, etc.

- * glib/dbus-gvalue.c (register_array, register_dict): Delete; not needed anymore due to generic array/map marshalling.
(dbus_g_value_types_init): Don't register basic arrays or the string/string hash.
(dbus_gtype_from_signature_iter): Don't try to recurse into variants.
(dbus_gtype_to_signature): Check collection/map before type metadata.
(demarshal_garray_basic): Renamed to demarshal_collection_array.
(demarshal_ghashtable): Renamed to demarshal_map; fix to use new generic map creation/append functions instead of hash table specifically.
(get_type_demarshaller): Handle maps.
(demarshal_collection): Dispatch on collection type to either demarshal_collection_ptrarray or demarshal_collection_array.
(get_type_marshaller): Handle maps.
(marshal_collection): Dispatch collection type to either marshal_collection_ptrarray or marshal_collection_array.
(_dbus_gvalue_test): New test.

- * glib/dbus-gvalue-utils.c (unset_and_free_g_value): New function.
(hash_free_from_gtype): Use it to free GValues.
(hashtable_append): New function.
(ptrarray_append): Fix prototype.
(slist_append): Ditto.
(_dbus_gvalue_utils_test): Extend tests.

- * glib/dbus-gtype-specialized.c
(dbus_g_type_specialized_init_append): Renamed from dbus_g_type_specialized_collection_init_append. Remove const from
value, since we steal it.
(dbus_g_type_specialized_map_append): New function.

- * glib/dbus-gtype-specialized.h: Update prototypes.
Add DBusGTypeSpecializedMapAppendFunc.

- * glib/dbus-gtest.c (dbus_glib_internal_do_not_use_run_tests):
Run

`_dbus_gvalue_test.`

* `glib/dbus-gtest.h`: Prototype it.

2005-07-08 Ross Burton <ross@openedhand.com>

* `dbus/dbus-glib.h`:
Add `DBysGAsyncData` for the async bindings.

* `glib/dbus-binding-tool-glib.c`:
Re-enable the async bindings.

* `test/glib/test-dbus-glib.c`:
Add a test for the generated async bindings.

2005-07-08 Colin Walters <walters@verbum.org>

* `doc/TODO`: Update GLib todo bits, also add a post-1.0 TODO for a connection concept.

2005-07-08 Colin Walters <walters@verbum.org>

* `tools/Makefile.am`: Kill of `print-introspect` in favor of using `dbus-send --print-reply=literal`.

* `test/glib/test-service-glib.xml`:
* `test/glib/test-service-glib.c` (`my_object_get_objs`): New test for "ao".

* `test/glib/test-dbus-glib.c` (`echo_received_cb`): Free echo data. (main): Test `GetObjs`.

* `glib/examples/statemachine/Makefile.am`:
* `glib/examples/statemachine/sm-marshal.list`:
* `glib/examples/statemachine/statemachine-client.c`:
* `glib/examples/statemachine/statemachine-server.c`:
* `glib/examples/statemachine/statemachine-server.xml`:
* `glib/examples/statemachine/statemachine.c`:
* `glib/examples/statemachine/statemachine.h`:
* `glib/examples/statemachine/statemachine.xml`:

New example.

* `glib/examples/example-service.c` (main): Move invocation of `dbus_g_object_type_install_info` earlier, to emphasize it should only be done once.

* `glib/examples/example-signal-emitter.c` (main): Ditto.

* `glib/examples/Makefile.am` (SUBDIRS): Include `statemachine`.

* `glib/dbus-gvalue.h` (`dbus_gtype_to_signature`)

```

(dbus_gvalue_marshal): Update prototypes.

* glib/dbus-gvalue.c: Update all marshalling functions to take
const GValue instead of GValue.
(signature_iter_to_g_type_array): Return a GPtrArray for nonfixed
types.
(dbus_gvalue_to_signature): Update for dbus_gtype_to_signature
change.
(dbus_gtype_to_signature): Handle generic collecitons and maps.
Return a newly-allocated string.
(demarshal_proxy, demarshal_object_path, demarshal_object)
(demarshal_strv, demarshal_ghashtable): Set error, don't assert
if
we get the wrong types from message.
(get_type_demarshaller): New function, extracted from
dbus_gvalue_demarshal.
(demarshal_collection): New function, demarshals generic
collection.
(dbus_gvalue_demarshal): Just invoke result of
get_type_demarshaller. Throw error if we don't have one.
(marshal_garray_basic): Abort on OOM.
(get_type_marshaller): New function, extracted from
dbus_gvalue_marshal.
(collection_marshal_iterator, marshal_collection): New functions;
implements generic marshalling for an iterable specialized
collection.
(dbus_gvalue_marshal): Just invoke result of get_type_marshaller.

* glib/dbus-gvalue-utils.c (gvalue_from_ptrarray_value): Handle
G_TYPE_STRING.
(ptrarray_value_from_gvalue): Ditto.
(ptrarray_append, ptrarray_free): New functions.
(slist_constructor, slist_iterator, slist_copy_elt, slist_copy)
(slist_append, slist_end_append, slist_free): New functions.
(dbus_g_type_specialized_builtins_init): Add append fuctions
for GPtrArray and GSList. Register GSList.
(test_specialized_hash, _dbus_gvalue_utils_test): New functions.

* glib/dbus-gtype-specialized.h
(DBusGTypeSpecializedAppendContext):
New.
(dbus_g_type_specialized_collection_init_append)
(dbus_g_type_specialized_collection_append)
(dbus_g_type_specialized_collection_end_append): Prototype.
(DBusGTypeSpecializedCollectionVtable): Add append_func and
end_append_func.

* glib/dbus-gtype-specialized.c
(dbus_g_type_specialized_collection_init_append)
(dbus_g_type_specialized_collection_append)
(dbus_g_type_specialized_collection_end_append): New functions.
(dbus_g_type_map_value_iterate): Take const GValue.

```

(dbus_g_type_collection_value_iterate): Ditto.
Run * glib/dbus-gtest.c (dbus_glib_internal_do_not_use_run_tests):
_dbus_gvalue_utils_test.

* glib/dbus-gtest.h: Prototype it.

* glib/dbus-gproxy.c (dbus_g_proxy_manager_filter): Avoid using uninitialized owner_list.
(dbus_g_proxy_begin_call_internal): Move return_if_fail to public API.

(dbus_g_proxy_end_call_internal): Update to use error set from dbus_gvalue_demarshal instead of setting it here.
(dbus_g_proxy_begin_call): Move return_if_fail here.

* glib/dbus-gobject.c (write_interface): Update for dbus_gtype_to_signature returning new string.

* configure.in: Add glib/examples/statemachine.

2005-07-08 Joe Shaw <joeshaw@novell.com>

* configure.in: Add a configure option, --with-console-auth-dir

* dbus/dbus-sysdeps-util.c (_dbus_user_at_console): Use the new setting. Patch from Kay Sievers.

2005-07-06 Colin Walters <walters@verbum.org>

* dbus/dbus-glib.h (DBusGPendingCall, DBusGPendingCallNotify) (DBUS_TYPE_G_PENDING_CALL, dbus_g_pending_call_get_g_type) (dbus_g_pending_call_ref, dbus_g_pending_call_unref): Delete. (dbus_g_pending_call_set_notify, dbus_g_pending_call_cancel): Delete in favor of dbus_g_proxy_begin_call and dbus_g_proxy_cancel_call.
user (DBusGProxyCall, DBusGProxyCallNotify): New.
(dbus_g_proxy_begin_call): Change prototype to take callback,

data, and destroy function. This replaces dbus_g_pending_call_set_notify.

(dbus_g_proxy_cancel_call): Prototype.

(DBusGAsyncData): Delete, shouldn't be needed anymore.

* glib/dbus-gproxy.c (struct _DBusGProxy): Add call_id_counter and pending_calls map.

(struct _DBusGProxyManager): Add bus_proxy member, which is an internal proxy for calls to the bus. Remove pending_nameowner_calls, now the internal proxy keeps track.

(dbus_g_proxy_manager_unref): Unref bus proxy, remove reference to

pending_nameowner_calls.
 (got_name_owner_cb): Update prototype, and use
 dbus_g_proxy_end_call.
 (got_name_owner_cb): Remove reference to pending_nameowner_calls.
 (dbus_g_proxy_manager_register): Delete directly libdbus code in
 favor of using internal proxy.
 (dbus_g_proxy_manager_unregister): Update to use
 dbus_g_proxy_cancel_call for any pending GetNameOwner call.
 (dbus_g_proxy_init): Initialize pending calls map.
 (dbus_g_proxy_constructor): New.
 (dbus_g_proxy_class_init): Add get/set property functions,
 constructor, and add NAME, PATH, and INTERFACE properties.
 (cancel_pending_call): New function.
 (dbus_g_proxy_dispose): Iterate over any outstanding calls and
 cancel them.
 (dbus_g_proxy_set_property, dbus_g_proxy_get_property): New.
 (GPendingNotifyClosure): New structure.
 (d_pending_call_notify, d_pending_call_free): Moved here from
 dbus-glib.c.
 (DBUS_G_VALUE_ARRAY_COLLECT_ALL): Moved around to satisfy
 function
 ordering.
 (manager_begin_bus_call): New internal function for talking to
 internal bus proxy.
 (dbus_g_proxy_new): Construct object using GObject properties.
 (dbus_g_proxy_begin_call_internal): Update to take user data,
 etc.
 Create closure of same, and insert call into map of pending
 calls.
 (dbus_g_proxy_end_call_internal): Take call id instead of pending
 call. Look up pending call in current set. Remove it when we've
 completed.
 (dbus_g_pending_call_end, dbus_g_proxy_end_call_internal):
 Delete.
 (dbus_g_proxy_begin_call): Change API to take callback, user
 data,
 and destroy function directly.
 (dbus_g_proxy_end_call): Update to take DBusGProxyCall.
 (dbus_g_proxy_call): Invoke with NULL callback.
 (dbus_g_proxy_cancel_call): New function, replaces
 dbus_g_pending_call_cancel.

 * glib/dbus-gparser.c (validate_signature): Fix call to
 dbus_set_g_error.

 * glib/dbus-gobject.c (dbus_g_object_type_dbus_metadata_quark):
 New quark for attaching metadata to GType.
 (info_hash): Delete.
 (lookup_object_info): Look up using quark.
 (dbus_g_object_type_install_info): Check that a type is classed,
 not that it's an object. Also just install type data using quark
 instead of using global hash.

- * glib/dbus-glib.c (dbus_g_pending_call_ref)
 (dbus_g_pending_call_unref, dbus_pending_call_get_g_type)
 (GPendingNotifyClosure): Delete.
 (d_pending_call_notify, d_pending_call_free): Move to dbus-
 gproxy.c.
 (dbus_g_pending_call_set_notify, dbus_g_pending_call_cancel):
 Delete.

- * glib/dbus-binding-tool-glib.c (generate_client_glue): Disable
 async
 client method generation until we can fix it...

- * tools/dbus-viewer.c (load_child_nodes): Use dbus_g_proxy_call.
 (load_from_service_thread_func): Ditto.

- * tools/dbus-names-model.c (struct NamesModel): Hold
 DBusGProxyCall.
 (have_names_notify): Update prototype, use
 dbus_g_proxy_cancel_call.
 (names_model_reload): Update for new dbus_g_proxy_begin_call API.

- * tools/dbus-monitor.c (filter_func): Update for print_message
 API change.

- * test/glib/test-dbus-glib.c: Add more tests for async
 invocations. Update many begin_call/end_call pairs to just use
 dbus_g_proxy_call.

- * tools/dbus-send.c (main): Add --print-reply=literal mode. This
 allows us to dump print-introspect.c.

- * tools/dbus-print-message.h (print_message): Add literal
 argument
 to print_message which is intended to allow printing arguments
 without
 metadata like "string=".

- * tools/dbus-print-message.c (print_iter): Add literal argument.
 (print_message): Allow printing string messages literally.

2005-07-05 Colin Walters <walters@verbum.org>

- * glib/dbus-gproxy.c (marshal_dbus_message_to_g_marshaller):
 Remove value refcount leak, original patch from Jorn Baayen
 <jorn@openedhand.com>. Also remove useless extra value in favor
 of prepending to value array directly.

2005-07-02 Colin Walters <walters@verbum.org>

- * glib/dbus-gmain.c (_dbus_gmain_test): Fix test.

2005-07-01 Colin Walters <walters@verbum.org>

Patch from Jonathan Matthew <jonathan@kaolin.hn.org>

- * glib/dbus-gvalue.c (basic_typecode_to_gtype): Fix return type.
- (dbus_g_value_types_init): Marshal G_TYPE_CHAR as DBUS_TYPE_BYTE, G_TYPE_LONG as DBUS_TYPE_INT32, G_TYPE_ULONG as DBUS_TYPE_UINT32, and G_TYPE_FLOAT as DBUS_TYPE_DOUBLE.

2005-06-30 Colin Walters <walters@verbum.org>

- * test/glib/test-dbus-glib.c:
- * test/glib/test-service-glib.c:
- * test/glib/test-service-glib.xml: Update tests for new error setting bits, also add async tests (patch from Ross Burton).

- * test/glib/Makefile.am (test_service_glib_LDADD): Add DBUS_GLIB_THREADS_LIBS.

- * glib/dbus-gproxy.c (get_name_owner) (dbus_g_pending_call_end_valist): Ditto.

- * glib/dbus-gobject.c (error_metadata): New mapping from GError domain (GQuark) to DBusGErrorInfo.
- (gerror_domaincode_to_dbus_error_name): Attempt to look up error quark in error_metadata. Take message interface as default error message interface.
- (gerror_to_dbus_error_message): Pass message interface.
- (dbus_set_g_error): Resurrected.
- (dbus_g_error_info_free): New function.
- (dbus_g_object_type_install_info): Use g_type_class_ref instead of _peek to actually create the object class if it hasn't been created yet.
- (dbus_g_error_domain_register): New function.

- * glib/dbus-gmain.c (dbus_g_bus_get): Switch to dbus_set_g_error.

- * glib/dbus-gparser.c (validate_signature): Ditto.

- * dbus/dbus-glib.h (dbus_g_error_set): Delete.
- (dbus_g_error_domain_register): Prototype.

- * glib/dbus-glib.c (dbus_g_error_set): Delete.
- Update tests.

2005-06-29 Colin Walters <walters@verbum.org>

- * dbus/dbus-glib.h: Delete DBUS_TYPE_G_PROXY_ARRAY. Add DBUS_TYPE_G_OBJECT_PATH.

- * glib/dbus-gvalue.c (dbus_g_value_types_init): Removemarshallers

for G_TYPE_OBJECT and DBUS_TYPE_G_PROXY_ARRAY (the latter should be handled more generically). Add DBUS_TYPE_G_OBJECT_PATH.
(dbus_g_object_path_get_g_type): New function.
(dbus_gtype_from_signature_iter): Map DBUS_TYPE_OBJECT_PATH to DBUS_TYPE_G_OBJECT_PATH by default.
(demarshal_proxy): Remove unused name variable.
(demarshal_object_path, marshal_object_path): New functions.
(demarshal_proxy_array, marshal_proxy_array): Delete.

* glib/dbus-binding-tool-glib.c (dbus_g_type_get_c_name): Map DBUS_TYPE_G_OBJECT_PATH to char *.
(dbus_g_type_get_lookup_function): Map builtin DBUS_TYPE_G_OBJECT_PATH.

* test/glib/test-dbus-glib.c
* test/glib/test-service-glib.c (my_object_objpath):
Adapt tests to new object path marshalling.

2005-06-29 John (J5) Palmieri <johnp@redhat.com>

* configure.in: force check for Python >= 2.4

2005-06-29 Colin Walters <walters@verbum.org>

Patch from Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c (invoke_object_method): Unset object value in all cases, not only in async case.

2005-06-29 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (struct DBusGProxy): Add new member name_call for keeping track of any outgoing GetNameOwner call. Also add for_owner and associated.
(struct DBusGProxyManager): Add owner_names, which is hash table that maps a base name to a list of names it owns (that we're interested in). Add pending_nameowner_calls which is a list of all outstanding GetNameOwner; avoids us having to iterate over every proxy. Add unassociated_proxies which keeps track of name proxies with no associated name owner.
(dbus_g_proxy_manager_unref): Destroy owner_names.
(struct DBusGProxyNameOwnerInfo): New struct for keeping track of name refcounts.
(find_name_in_info, name_owner_foreach)
(dbus_g_proxy_manager_lookup_name_owner, insert_nameinfo)
(dbus_g_proxy_manager_monitor_name_owner)
(dbus_g_proxy_manager_unmonitor_name_owner)
(unassociate_proxies, dbus_g_proxy_manager_replace_name_owner): New functions; they manipulate the owner_names mapping.
(got_name_owner_cb): New function.
(get_name_owner): New function, extracted from dbus_g_proxy_new_for_name_owner.

(dbus_g_proxy_manager_register): For now we need to keep track of all NameOwnerChanged. Also if the proxy is for a name, if we don't already know the name owner, queue a new GetNameOwner request and add it to our list of unassociated proxies.

Otherwise

inc the refcount.

(dbus_g_proxy_manager_unregister): If this proxy is for a name, cancel any pending GetNameOwner call, etc.

(dbus_g_proxy_manager_filter): Handle NameOwnerChanged. Also use the owner_names mapping to look up the current names for the signal source, and dispatch to any proxies for that name.

(dbus_g_proxy_new): Initialize new members.

(dbus_g_proxy_new_for_name): Delete unused proxy variable.

(dbus_g_proxy_new_for_name_owner): Use get_name_owner.

(dbus_g_pending_call_end_valist): New function, extracted from dbus_g_proxy_end_call_internal. Useful when we don't have a

proxy

but want to use the GLib infrastructure. Also note how many arguments in reply were over.

(dbus_g_pending_call_end): New function, just call dbus_g_pending_call_end_valist.

(dbus_g_proxy_end_call_internal): Just call dbus_g_pending_call_end_valist.

* glib/dbus-gobject.c (_dbus_gobject_lookup_marshall): Fix lookup of builtin marshaller for STRING_STRING_STRING.

* test/glib/test-dbus-glib.c:

* test/glib/test-service-glib.c:

* test/glib/test-service-glib.xml:

Extend tests to cover name proxies, destruction of owner proxies, etc.

* glib/examples/example-signal-recipient.c

(dbus_g_proxy_new_for_name_owner): Create a name proxy.

* tools/dbus-send.c (main): Print D-BUS error name in addition to message.

2005-06-28 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in (cunregister_function_handler, cmessage_function_handler): Patch from Anthony Baxter <anthony@interlink.com.au> fixes threading

problems

by using the Py_GILState_Ensure/Release to synchronize with the python runtime.

2005-06-28 Ray Strode <rstrode@redhat.com>

* dbus/dbus-spawn.c (_dbus_babysitter_unref): kill

babysitter helper process on last unref, bug #2813.

2005-06-27 Colin Walters <walters@verbum.org>

- * test/glib/test-dbus-glib.c:
- * test/glib/test-service-glib.c:
- * test/glib/test-service-glib.xml:

Test hash table signal emitting.

- * glib/dbus-gobject.c (`_dbus_gobject_lookup_marshallers`): Convert types to their fundamental basis types, since this is whatmarshallers operate on. Also add an entry for `VOID_BOXED`.
- (`dbus_g_object_register_marshallers_array`): Convert to fundamental.

2005-06-26 Havoc Pennington <hp@redhat.com>

- * doc/dbus-tutorial.xml: fix names of interface/service/path, fix from Don Park

2005-06-26 Colin Walters <walters@verbum.org>

- * glib/dbus-glib.c (`dbus_set_g_error`): Delete.
- (`dbus_g_error_set`): New public function from its ashes; used by both service-side method implementation and GLib bindingsinternals.
- (`dbus_g_error_has_name`, `dbus_g_error_get_name`): New function.
- (`_dbus_glib_test`): Add some tests.

- * test/glib/test-dbus-glib.c (main): Test `dbus_g_error_has_name`.

- * test/glib/test-service-glib.c (`my_object_throw_error`): Use `dbus_g_error_set`.

- * glib/dbus-gobject.c (`gerror_to_dbus_error_message`): Handle errors thrown by `dbus_g_error_set`.

- * glib/dbus-gmain.c (`dbus_g_bus_get`): Change to `dbus_g_error_set`.

- * glib/dbus-gparser.c (`validate_signature`): Ditto.

- * glib/dbus-gproxy.c (`dbus_g_proxy_new_for_name_owner`)
- (`dbus_g_proxy_end_call_internal`): Ditto.

- * glib/Makefile.am: Generate `dbus-glib-error-switch.h`, which converts `DBUS_ERROR_x` to `DBUS_GERROR_x`.
- (`libdbus_glib_1_la_SOURCES`, `BUILT_SOURCES`, `CLEANFILES`): Add it.

- * doc/TODO: Remove error TODO.

- * doc/dbus-tutorial.xml: Update with documentation about error handling.

* dbus/make-dbus-glib-error-enum.sh: Tighten up regexp to make sure we only change DBUS_ERROR to DBUS_GERROR, not all ERROR to GERROR. Also add DBUS_GERROR_REMOTE_EXCEPTION.

2005-06-22 Colin Walters <walters@verbum.org>

Patch from Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c (dbus_g_method_return): Free out_sig.

2005-06-20 Colin Walters <walters@verbum.org>

* configure.in: Add glib/examples.

* glib/Makefile.am: Add examples/

* glib/examples/.cvsignore

* glib/examples/Makefile.am

* glib/examples/example-client.c

* glib/examples/example-service.c

* glib/examples/example-service.xml

* glib/examples/example-signal-emitter.c

* glib/examples/example-signal-emitter.xml

* glib/examples/example-signal-recipient.c:

New files; GLib binding examples, ported from python/examples.

2005-06-20 Colin Walters <walters@verbum.org>

* dbus/dbus-glib.h:

* glib/dbus-gproxy.c: Rename dbus_g_proxy_invoke to dbus_g_proxy_call.

* glib/dbus-binding-tool-glib.c:

* doc/dbus-tutorial.xml:

* test/glib/test-dbus-glib.c: Update for rename.

2005-06-20 Colin Walters <walters@verbum.org>

Patch suggested by Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c (export_signals): Free signal name.

(g_value_init): Use G_VALUE_NOCOPY_CONTENTS to plug memory leak. Add a bit of documentation.

(dbus_g_method_return_error): Free context, and note we do so.

2005-06-18 Murray Cumming <murrayc@murrayc.com>

* dbus/dbus-glib.h:

* glib/dbus-gobject.c:

* glib/dbus-gproxy.c:
* glib/dbus-gvalue.c: Predeclare structs as
typedef struct _Something Something instead of
typedef struct Something Something, so we can
redeclare the prototypes. Other GNOME libraries
do this already.

2005-06-17 Colin Walters <walters@verbum.org>

* tools/dbus-names-model.c (have_names_notify): Fix call
to dbus_g_proxy_end_call.

2005-06-17 Colin Walters <walters@verbum.org>

* glib/dbus-gproxy.c (dbus_g_proxy_emit_remote_signal): Don't
spew warnings if we get malformed remote signals.

* glib/dbus-gobject.c (propsig_iterate): New function.
(lookup_object_info): New function, extracted from
lookup_object_and_method.
(introspect_properties, introspect_signals): Delete; these
are merged into write_interface.
(write_interface): Write out signals and properties here;
dump the org.gtk.object stuff and use the interface given
in the introspection data blob. Also fix up property XML.
(lookup_values): New function.
(introspect_interfaces): Gather a mapping from interface to a
list of its methods, signals, and properties, then write out
each interface.
(lookup_object_and_method): Use lookup_object_info.
(struct DBusGSignalClosure): Add interface.
(dbus_g_signal_closure_new): Add interface. Don't dup signame;
we can just use the constant data.
(dbus_g_signal_closure_finalize): Don't free signal name.
(signal_emitter_marshall): Use interface from signal closure.
(export_signals): Only export signals mentioned in introspection
blob.
(dbus_g_connection_register_g_object): Warn if we have no
introspection data for an object.
(funcsig_equal): Remove unused variable.
(dbus_g_object_register_marshall): Take varargs instead of
list.
(dbus_g_object_register_marshall_array): New function,
extracted from old dbus_g_object_register_marshall.

* glib/dbus-binding-tool-glib.c (struct DBusBindingToolCData):
Add
signals and property data.
(write_quoted_string): New function, extracted from
generate_glue.
(generate_glue): Write signals and properties to introspection
blob.

- * dbus/dbus-glib.h (struct DBusGObjectInfo): Include exported_signals and exported_properties.
- (dbus_g_object_register_marshall): Update prototype.
- (dbus_g_object_register_marshall_array): Prototype.

- * test/glib/test-dbus-glib.c: Extend testing to cover new signals.

- * test/glib/test-service-glib.c: Add new test signals and method to emit them.

- * test/glib/test-service-glib.xml: Add some test signals.

- * test/glib/Makefile.am (BUILT_SOURCES): Add my-object-marshal.c and my-object-marshal.h
- (test_service_glib_SOURCES, test_dbus_glib_SOURCES): Add my-object-marshal.c.
- (my-object-marshal.c, my-object-marshal.h): Implement.

- * test/glib/.cvsignore: Update.

- * doc/TODO: Remove two GLib TODO items fixed by this patch.

2005-06-16 Colin Walters <walters@verbum.org>

- * doc/TODO: Update for GLib bindings.

2005-06-16 Colin Walters <walters@verbum.org>

- * glib/dbus-binding-tool-glib.c:
- * glib/dbus-gobject.c:
- * glib/dbus-gproxy.c: Add Nokia copyright; Patch from Ross Burton, for his GLib bindings work.

2005-06-16 Colin Walters <walters@verbum.org>

- * glib/dbus-gobject.c (funcsig_hash, funcsig_equal): Use n_params to iterate instead of walking to G_TYPE_INVALID.

Patch based on a patch from Ryan Gammon.

2005-06-16 Colin Walters <walters@verbum.org>

- * bus/bus.c (bus_context_new): Set parser to NULL after we unref it (Patch from Chris Boscolo, #2174).

2005-06-16 Colin Walters <walters@verbum.org>

- * python/dbus_bindings.pyx.in: Import size_t, __int64_t, __uint64_t, and __signed.

* dbus/dbus-sysdeps.c <HAVE_CMSGCRED> (write_credentials_byte):
Define cmsg struct, output it.
(_dbus_read_credentials_unix_socket):
Use cmsg struct.

Patch from Joe Markus Clarke for FreeBSD support.

2005-06-16 Colin Walters <walters@verbum.org>

* tools/dbus-send.c (append_array): Use strtok.
(append_dict): New function.
(type_from_name): New function, extracted from main.
(main): Handle sending dicts.

* tools/dbus-print-message.c (print_iter): Print dict
entries.

2005-06-16 Colin Walters <walters@verbum.org>

* glib/dbus-gvalue.c (marshal_basic): Marshal NULL string
values as the empty string (#2948).

2005-06-16 Colin Walters <walters@verbum.org>

* dbus/Makefile.am:
* mono/doc/Makefile.am:
* test/glib/Makefile.am:

Fix srcdir != builddir issues (Patch from Chris Wilson, #3477)

2005-06-16 Colin Walters <walters@verbum.org>

* dbus/dbus-marshal-header.c (_dbus_header_load): Set
header byte order from expected byte order (Patch from Chris
Wilson, #3475).

* dbus/dbus-marshal-byteswap.c (byteswap_body_helper):
Increment pointer after swapping fixed array. Add assertion
for array length.

2005-06-15 Colin Walters <walters@verbum.org>

* dbus/dbus-sysdeps.c <HAVE_CMSGCRED>
(_dbus_read_credentials_unix_socket):
Fix call to dbus_set_error. (Patch from Michael Banck, #3461)

2005-06-15 John (J5) Palmieri <johnp@redhat.com>

* NEWS: Update to 0.34

2005-06-15 David Zeuthen <davidz@redhat.com>

* configure.in (LT_CURRENT): Revert back to 1 as the library hasn't changed and we've certainly not committed to protocol stability yet. All this does is to break ABI. See commit note from hp@redhat.com 2005-05-05 for details.

2005-06-15 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c (_dbus_connection_peer_filter): New method
(_dbus_connection_run_builtin_filters): New method
(dbus_connection_dispatch): Run the builtin filters which in turn runs the peer filter which handles Ping messages.

* doc/TODO:
- Ping isn't handled: This patch fixes it

- Add a test case for the Ping message: added TODO item

2005-06-15 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-message.c:
(dbus_message_has_path): New method
(dbus_message_has_interface): New method
(dbus_message_has_member): New method

* dbus/dbus/dbus-sysdeps.c (_dbus_check_dir_is_private_to_user):
New method

* dbus/dbus-keyring.c (_dbus_keyring_reload): Check to see that the keyring directory is private to the user

* doc/TODO:
- The convenience functions in dbus-bus.h should perhaps have the signatures that they would have if they were autogenerated stubs. e.g. the acquire service function. We should also evaluate
which of these functions to include, in light of the fact that GLib/Qt native stubs will probably also exist.: Punted

- add dbus_message_has_path(), maybe has_member/interface:
fixed in this patch

- in dbus-keyring.c, enforce that the keyring dir is not world readable/writable: Fixed in this patch

2005-06-15 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-marshall-validate.h: Added a new validation error code DBUS_VALIDITY_UNKNOWN_OOM_ERROR = -4 for out of memory errors when validating signatures

* dbus/dbus-marshall-header.c: use DBUS_VALIDITY_UNKNOWN_OOM_ERROR in places where we previously used DBUS_VALID and a FALSE return value to indicate OOM

* dbus/dbus-marshall-validate.c
(_dbus_validate_signature_with_reason):
Use a stack to track the number of elements inside containers.
The stack values are then used to validate that dict entries have only two elements within them.
(validate_body_helper): check the reason for failure when validating variants

* dbus/dbus-message.c (load_message): use DBUS_VALIDITY_UNKNOWN_OOM_ERROR in places where we previously used DBUS_VALID and a FALSE return value to indicate OOM

* doc/TODO: remove "- validate dict entry number of fields" as this patch fixes it

2005-06-14 David Zeuthen <davidz@redhat.com>

* bus/bus.c (process_config_every_time): Drop existing conf-dir watches (if applicable) and add new watches

* bus/main.c (signal_handler): Handle SIGIO if using D_NOTIFY
(main): Setup SIGIO signal handler if using D_NOTIFY

* bus/config-parser.h: Add prototype
bus_config_parser_get_conf_dirs

* bus/config-parser.c (struct BusConfigParser): Add conf_dirs list
(merge_included): Also merge conf_dirs list
(bus_config_parser_unref): Clear conf_dirs list
(include_dir): Add directory to conf_dirs list
(bus_config_parser_get_conf_dirs): New function

* bus/dir-watch.[ch]: New files

* bus/Makefile.am (BUS_SOURCES): Add dir-watch.[ch]

* configure.in: Add checks for D_NOTIFY on Linux

2005-06-14 Colin Walters <walters@verbum.org>

* glib/dbus-binding-tool-glib.c:
* glib/dbus-gobject.c:

* glib/dbus-gvalue.c: Fix indentation and brace style.

2005-06-14 Ross Burton <ross@openedhand.com>.

* glib/dbus-glib.h: Make DBusGMethodInvocation a private structure. Rearrange prototypes a bit.

* glib/dbus-gproxy.c (dbus_g_proxy_invoke): Add documentation for first_arg_type.

* glib/dbus-gobject.c: Move DBusGMethodInvocation here, add documentation. Move dbus_g_method_return and dbus_g_method_return_error into public API section.

2005-06-14 Colin Walters <walters@verbum.org>

* glib/dbus-gobject.c (_dbus_gobject_lookup_marshall): Add missing return statements, noticed by Ross Burton.

2005-06-13 Ross Burton <ross@openedhand.com>.

* glib/dbus-gobject.c: Handle errors on message demarshalling by sending error message back.

* glib/dbus-gvalue.c: Initialize return variables.

2005-06-13 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Fix thinko in last patch.

2005-06-13 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Move dbus-gtype-specialized.c and dbus-gtype-specialized.h into a _HEADERS variable, install them.

2005-06-12 Colin Walters <walters@verbum.org>

Async signals and various bugfixes and testing by Ross Burton <ross@openedhand.com>.

* glib/dbus-gvalue.h: (struct DBusBasicGValue): Delete.
(dbus_gvalue_genmarshal_name_from_type)
(dbus_gvalue_ctype_from_type): Moved to dbus-binding-tool-glib.c.
(dbus_gtype_to_dbus_type): Renamed to dbus_gtype_from_signature.
(dbus_g_value_types_init, dbus_gtype_from_signature)
(dbus_gtype_from_signature_iter, dbus_gtype_to_signature)
(dbus_gtypes_from_arg_signature): New function prototypes.
(dbus_gvalue_demarshal): Take context and error arguments.
(dbus_gvalue_demarshal_variant): New function.
(dbus_gvalue_demarshal_message): New function.
(dbus_gvalue_store): Delete.

* glib/dbus-gvalue.c:

File has been almost entirely rewritten; now we special-case more types such as DBUS_TYPE_SIGNATURE, handle arrays and hash tables correctly, etc. Full support for recursive values is not yet complete.

* glib/dbus-gproxy.c (dbus_g_proxy_class_init): Change last argument of signal to G_TYPE_POINTER since we now pass a structure.

(lookup_g_marshall): Delete in favor of
_dbus_gobject_lookup_marshall.

(marshal_dbus_message_to_g_marshall): Use
_dbus_gobject_lookup_marshall and dbus_gvalue_demarshal_message
to handle remote signal callbacks.

(dbus_g_proxy_new_from_proxy): New function; creates a new
DBusGProxy by copying an existing one.

(dbus_g_proxy_get_interface, dbus_g_proxy_set_interface)

(dbus_g_proxy_get_path): New functions.

(dbus_g_proxy_marshal_args_to_message): New function;
factored out of existing code.

(DBUS_G_VALUE_ARRAY_COLLECT_ALL): Collect all arguments
from a varargs array.

(dbus_g_proxy_begin_call_internal): New function.

(dbus_g_proxy_end_call_internal): New function.

(dbus_g_proxy_begin_call): Take GTypes instead of DBus types
as arguments; simply invoke dbus_g_proxy_begin_call_internal
after collecting args into value array.

(dbus_g_proxy_end_call): Take GTypes instead of DBus types;
invoke dbus_g_proxy_end_call_internal.

(dbus_g_proxy_invoke): Simply invoke begin_call_interanl and
end_call_internal.

(dbus_g_proxy_call_no_reply): Take GTypes instead of DBus
types.

(array_free_all): New function.

(dbus_g_proxy_add_signal): Take GTypes.

* glib/dbus-gobject.h:

(_dbus_glib_marshal_dbus_message_to_gvalue_array): Delete.

(_dbus_gobject_get_path, _dbus_gobject_lookup_marshall):
Prototype.

* glib/dbus-gobject.c: Add a global marshal_table hash which
stores mappings from type signatures tomarshallers. Change lots
of invocations of dbus_gtype_to_dbus_type to
dbus_gtype_to_signature.

(_dbus_glib_marshal_dbus_message_to_gvalue_array): Delete.

(introspect_signals): Fix test for query.return_type.

(set_object_property): Update invocation of
dbus_gvalue_demarshal.

(invoke_object_method): Many changes. Handle asynchronous

invocations. Convert arguments with `dbus_gvalue_demarshal_message`. Handle errors. Use `DBusSignatureIter` instead of `strlen` on `args`. Handle all arguments generically. Special-case variants.

`(dbus_g_method_return, dbus_g_method_return_error)`: New function.

`(DBusGSignalClosure)`: New structure, closes over signal information.

`(dbus_g_signal_closure_new)`: New function.

`(dbus_g_signal_closure_finalize)`: New function.

`(signal_emitter_marshall)`: New function; is special marshaller which emits signals on bus.

`(export_signals)`: New function; introspects object signals and connects to them.

`(dbus_g_object_type_install_info)`: Take `GType` instead of `GObjectClass`.

`(dbus_g_connection_register_g_object)`: Invoke `export_signals`.

`(dbus_g_connection_lookup_g_object)`: New function.

`(DBusGFuncSignature)` New structure; used for mapping type signatures tomarshallers.

`(funcsig_hash)`: New function; hashes `DBusGFuncSignature`.

`(funcsig_equal)`: New function; compares `DBusGFuncSignature`.

`(_dbus_gobject_lookup_marshall)`: New function.

`(dbus_g_object_register_marshall)`: New function; used to register a marshaller at runtime for a particular signature.

* `glib/dbus-gmain.c` (`_dbus_gmain_test`): Add various tests.

* `glib/dbus-binding-tool-glib.h`: Add `DBUS_GLIB_ANNOTATION_ASYNC` which notes a server method implementation should be asynchronous.

* `glib/dbus-binding-tool-glib.c`

`(dbus_binding_tool_output_glib_server)`: Call `dbus_g_value_types_init`.

`(write_formal_parameters)`: Use `dbus_gtype_from_signature`. Handle variants specially.

`(dbus_g_type_get_lookup_function)`: Turn `GType` into an invocation of a lookup function.

`(write_args_for_direction)`: Use `dbus_g_type_get_lookup_function`.

`(write_untyped_out_args)`: New method; write output arguments.

`(write_formal_declarations_for_direction)`: Function for writing prototypes.

`(write_formal_parameters_for_direction)`: Function for writing implementations.

`(write_typed_args_for_direction)`: Function for writing arguments prefixed with `GTypes`.

`(write_async_method_client)`: Write out async version of method.

* `glib/dbus-binding-tool-glib.c`: Include `dbus-gvalue-utils.h`.

`(dbus_g_type_get_marshal_name)`: Move mapping from `GType` to marshal name into here.

```

(dbus_g_type_get_c_name): Move into here.
(compute_marshall): Convert signature to type with
dbus_gtype_from_signature, use dbus_g_type_get_marshall_name.
(compute_marshall_name): Ditto.
(compute_marshall): Handle async signal annotations.
(gather_marshallers): Return if we don't have a known
prefix.
(generate_glue): Collect introspection blob here, and
write all of the blob at the end. This allows an object
with multiple interfaces to work.
Mark async methods in introspection blob.

* glib/Makefile.am (libdbus_glib_1_la_SOURCES): Add
dbus-gtype-specialized.c, dbus-gtype-specialized.h,
dbus-gvalue-utils.h, dbus-gvalue-utils.c.

* dbus/dbus-glib.h: Don't include dbus-protocol.h; this
avoids people accidentally using DBUS_TYPE_* which should
not be necessary anymore.
Do include dbus-gtype-specialized.h, which are utilities
for GLib container types.
Add various #defines for types such as
DBUS_TYPE_G_BOOLEAN_ARRAY.
(DBusGValueIterator, DBusGValue): Define, not fully used
yet.
(dbus_g_value_get_g_type): Type for recursive value.
(dbus_g_value_open, dbus_g_value_iterator_get_value)
(dbus_g_value_iterator_get_values, dbus_g_value_iterator_recurse)
(dbus_g_value_free): Prototypes.
(dbus_g_object_register_marshall, dbus_g_proxy_new_from_proxy):
Prototype.
(dbus_g_proxy_set_interface): Prototype.
(dbus_g_proxy_begin_call, dbus_g_proxy_end_call)
(dbus_g_proxy_call_no_reply): Take GLib types instead of Dbus
types.
(dbus_g_proxy_get_path, dbus_g_proxy_get_interface):
Accessors.
(DBusGAsyncData, DBusGMethodInvocation): Structures for
doing async invocations.
(dbus_g_method_return, dbus_g_method_return_error):
Prototypes.
* doc/dbus-tutorial.xml: Update GLib section.

* tools/dbus-viewer.c (load_child_nodes): Update
for new invocation type of dbus_g_proxy_end_call.
(load_from_service_thread_func): Ditto.

* tools/print-introspect.c (main): Ditto.

* tools/dbus-names-model.c (have_names_notify)
(names_model_reload, names_model_set_connection)
Use GTypes.

```


- * python/Makefile.am (INCLUDES): Define DBUS_COMPILATION, needed since Python bindings use GLib bindings.
- * test/glib/Makefile.am (INCLUDES): Define DBUS_COMPILATION. Add --prefix argument.
- * tools/Makefile.am: Define DBUS_COMPILATION. Remove unneeded --ignore-unsupported arg.
- * test/glib/test-service-glib.c:
- * test/glib/test-service-glib.xml:
- * test/glib/test-dbus-glib.c: Add many more tests.

2005-06-06 David Zeuthen <davidz@redhat.com>

- * doc/TODO: Add item about need to remove deprecated functions.
- * dbus/dbus-connection.h: Add prototype for dbus_connection_disconnect
- * dbus/dbus-connection.c (dbus_connection_disconnect): New function to repair the ABI which was broken with the last commit.

2005-06-02 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-connection.c, dbus/dbus-connection.h (dbus_connection_disconnect): renamed to dbus_connection_close for API symmetry with dbus_connection_open (_dbus_connection_open_internal): s/dbus_connection_disconnect/dbus_connection_close
- * dbus/dbus-bus.c (dbus_bus_get): s/dbus_connection_disconnect/dbus_connection_close
- * bus/connection.c (bus_connections_unref, bus_connections_setup_connection, bus_connections_expire_incomplete): s/dbus_connection_disconnect/dbus_connection_close
- * bus/dispatch.c (bus_dispatch, kill_client_connection, kill_client_connection_unchecked, check_hello_connection): s/dbus_connection_disconnect/dbus_connection_close
- * bus/bus.c (new_connection_callback): s/dbus_connection_disconnect/dbus_connection_close
- * tools/dbus-send.c (main): s/dbus_connection_disconnect/dbus_connection_close

```
* test/glib/test-profile.c (no_bus_thread_func,
with_bus_thread_func):
  s/dbus_connection_disconnect/dbus_connection_close

* test/test-service.c (path_message_func, filter_func):
  s/dbus_connection_disconnect/dbus_connection_close

* doc/TODO: remove connection_open/connection_disconnect lacks
symmetry  item that was just fixed
```

2005-05-25 Colin Walters <walters@verbum.org>

```
* dbus/dbus-protocol.h: Move various bus service #defines such
as DBUS_SERVICE_DBUS and DBUS_NAME_FLAG_PROHIBIT_REPLACEMENT to
dbus/dbus-shared.h.
* dbus/dbus-shared.h: Various defines moved here.
* dbus/dbus-marshall-header.c: Include dbus-shared.h.
```

2005-05-25 John (J5) Palmieri <johnp@redhat.com>

```
* python/__init__.py: Python bindings deserve a minor version
update. Upped to (0, 40, 2)
```

2005-05-24 John (J5) Palmieri <johnp@redhat.com>

```
* python/decorators.py: add explicitly_pass_message decorator
for passing in the dbus message as keyword for edge case signal
handling

* python/matchrules.py (SignalMatchRule.__repr__): fix output
to conform with what dbus expects for match rules
(SignalMatchRule.execute): add the dbus message as a keyword
if the signal handler has requested it

* python/examples/example/signal-recipient.py: added some more
examples on how to hook up to signals
```

2005-05-23 John (J5) Palmieri <johnp@redhat.com>

```
* python/decorators.py: import dbus_bindings

* python/matchrules.py (SignalMatchRule, SignalMatchTree,
SignalMatchNode): new classes that implement wildcard signal
callback matching using a tree lookup. Heavily modified from a
patch sent by Celso Pinto (fd.o bug #3241)

* _dbus.py (add_signal_receiver, remove_signal_receiver,
_signal_func):
  use new match classes to handle signals.
```

2005-05-19 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in: s/TYPE_PATH/TYPE_OBJECT_PATH

2005-05-18 Havoc Pennington <hp@redhat.com>

* configure.in: use GLIB_GNU_GETTEXT to get INTLLIBS and require gettext. Not really worth requiring yet perhaps, but any production quality 1.0 would require it so we should go ahead and get things set up. We do have a couple token calls to bindtextdomain in the code already.

2005-05-16 John (J5) Palmieri <johnp@redhat.com>

* glib/dbus-gmain.c (io_handler_dispatch): fix deadlock when using recursive g_main_loops

* python/_dbus.py (class Bus): add the ProxyObjectClass alias for ProxyObject to make it easier for the Twisted networking framework to integrate dbus.

* python/proxies.py (class ProxyObject): add the ProxyMethodClass alias for ProxyMethod to make it easier for the Twisted networking framework to integrate dbus.

2005-05-11 Ross Burton <ross@openedhand.com>

* glib/dbus-glib-tool.c: Add --prefix argument.
* glib/dbus-binding-tool-glib.h: Add prefix argument.
* glib/dbus-binding-tool-glib.c (compute_marshall_name): Add prefix argument.
(generate_glue): Pass prefix argument down.
(dbus_binding_tool_output_glib_server): Pass prefix to glib-genmarshal.

2005-05-11 Colin Walters <walters@verbum.org>

* tools/dbus-send.c (append_array): New function.
(append_arg): Broken out from main.
(main): Add cheesy hack to send arrays and variants.
(usage): Update.
* tools/dbus-print-message.c (print_iter): Broken out from main.

2005-05-11 Colin Walters <walters@verbum.org>

* dbus/dbus-signature.c (dbus_signature_iter_get_signature): New function, returns signature string for signature iter.
* dbus/dbus-signature.h: Prototype it.
* dbus/dbus-message.c (dbus_message_iter_get_signature): New function, returns signature string for message iter.
(dbus_message_iter_get_array_len): New function, returns length of array.
(dbus_message_iter_get_fixed_array): Fix assertion; this

function should be used when the iter is pointing to the contents of an array

- * dbus/dbus-message.h: Prototypes.
- * dbus/dbus-marshal-recursive.c

(`_dbus_type_reader_get_array_length`):
New function; returns length of an array.

- * dbus/dbus-marshal-recursive.h: Prototype it.

2005-05-11 Colin Walters <walters@verbum.org>

- * dbus/dbus-sysdeps-util.c <!HAVE_POSIX_GETPWNAM_R>: Fix compilation error.

2005-05-08 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps-util.c (`_dbus_become_daemon`): write the daemon's pid, not the parent's pid, to the file descriptor. Reported by Taj Morton.

2005-05-05 Havoc Pennington <hp@redhat.com>

- * configure.in (LT_*): add notes on how the libtool versioning works to save thinking. Increment soname to indicate protocol breakage (though really the library interface hasn't changed I guess)
- * dbus/dbus-transport.c (`_dbus_transport_get_is_authenticated`): verify the GUID received from server matches what we were expecting, if we had an expectation
- * dbus/dbus-auth.c (`send_ok`): send GUID along with the OK command
(`_dbus_auth_get_guid_from_server`): new function
(`send_begin`): parse the OK args
- * doc/dbus-specification.xml: add GUID to the auth protocol

2005-05-05 John (J5) Palmieri <johnp@redhat.com>

- * Fix my name in previous changelog ;)
- * python/proxies.py (`ProxyObject.__getattr__`): add further patch from Anthony Baxter to throw an `AttributeError` when python `__special__` functions are called instead of marshling them over the bus (Bug#1685 comment 3).

2005-05-04 John (J5) Palmieri <johnp@redhat.com>

- * python/Makefile.am: changed to use `pyexecdir` for the binding shared libraries (Bug#2494)
- * python/exceptions.py: bring exceptions over from the bindings so they can be used in applications (Bug#2036)

Make all exceptions derive from DBusException

* python/_dbus.py, python/proxies.py: implement __repr__ in a couple of classes so that print obj doesn't throw an exception (Bug #1685)

2005-05-03 Ross Burton <ross@openedhand.com>

* glib/dbus-gobject.c (dbus_g_connection_register_g_object): Return if we get an error during registration. Set up a weak reference on object to unregister if object is destroyed. (unregister_gobject): New function.

2005-05-01 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in:
- added new type classes for hinting to the marshaller what type to send over the wire
- added int16 and uint16 marshalers
- Fixed a bug in the type constants that caused int32 to go out as uint16 over the wire
* python/dbus.py: split up into different files and renamed _dbus.py
* python/__init__.py, python/_util.py, python/decorators.py, python/exceptions.py, python/proxies.py, python/services.py, python/types.py: new files split off from dbus.py
* python/Makefile.am: Add new files, remove dbus.py and install all python files to <python module dir>/dbus
* python/examples/*: Added #!/usr/bin/env python to the top of every example. Patch provided by Tatavarty Kalyan

2005-04-25 John (J5) Palmieri <johnp@redhat.com>

* NEWS: Update to 0.33

2005-04-25 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in (send_with_reply_handlers): New send method for doing async calls
(_pending_call_notification): New C function for handling pending call callbacks
(set_notify): New method for setting pending call notification

* python/dbus.py: new version tuple "version" is set at (0, 40, 0)
Async capabilities added to remote method calls
(Sender): class removed
(RemoteService): class removed
(ObjectTree): class removed for now

(RemoteObject): Renamed to ProxyObject
(RemoteMethod): Renamed to ProxyMethod
(method): Decorator added for decorating python methods as dbus methods
(signal): Decorator added for decorating python methods as signal emitters
(ObjectType): Metaclass added for generating introspection data and the method callback vtable
(Interface): Wrapper class added to wrap objects in a dbus interface
(Object): Uses ObjectType as its metaclass and exports Introspect of the org.freedesktop.DBus.Introspectable interface
(ValidationException, UnknownMethodException): new exceptions

* python/examples/*: Modified to fit with the new bindings

2005-04-23 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_append_args): fix doc comment, reported by Tony Houghton

* test/test-service.c (main): test dbus_connection_get_object_path_data()

* dbus/dbus-object-tree.c (find_handler): be sure we always init the exact_match
(_dbus_object_tree_get_user_data_unlocked): new function used by dbus_connection_get_object_path_data()
(do_register): add assertion test for get_user_data_unlocked
(object_tree_test_iteration): more tests

* dbus/dbus-connection.c (dbus_connection_get_object_path_data): new function from Dan Reed to let you get the user data from dbus_connection_register_object_path()

2005-04-23 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-marshall-recursive-util.c: Fixed buffer overflow in numerous places that did not account for the NULL terminator (signature_from_seed): changed the manual string copy loop to just use strcpy instead
make check should now pass

2005-04-19 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-marshall-header.c (_dbus_header_create): Fix assert so that it allows messages that are not signals to pass in NULL as the interface.

2005-04-18 David Zeuthen <davidz@redhat.com>

* glib/dbus-gmain.c (io_handler_destroy_source):
(timeout_handler_destroy_source, connection_setup_free):
Also unref the source to avoid memory leaks.

2005-04-13 David Zeuthen <davidz@redhat.com>

* bus/config-parser.c (bus_config_parser_new): Bump this to a
more reasonable, yet still totally arbitrary, value :-).

2005-04-13 David Zeuthen <davidz@redhat.com>

* doc/TODO: Added an "important for 1.0" item about selinux
allow/deny messages

2005-04-13 David Zeuthen <davidz@redhat.com>

* bus/selinux.c: Add c-file-style to top of file
(log_audit_callback): Don't free the data here anymore
(bus_selinux_check): Don't take spid and tpid since appending
that to auxdata may OOM.
(bus_selinux_allows_acquire_service): Handle OOM and signal back
to the caller if we are OOM by taking an error object.
(bus_selinux_allows_send): -do-

* bus/selinux.h: Fix prototypes for
bus_selinux_allows_acquire_service
and bus_selinux_allows_send

* bus/bus.c (bus_context_check_security_policy): Pass error and
pass on OOM thrown by bus_selinux_allows_send()

* bus/services.c (bus_registry_acquire_service): Pass error and
pass on OOM thrown by bus_selinux_allows_acquire_service()

2005-04-13 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c (message_queue_dispatch): only dispatch one
message at a time to avoid monopolizing the main loop, bug
#2953 from Benjamin Otte

2005-04-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (copy): change a memcpy to memmove due to
possible overlap, fix from Daniel Reed
(fixup_alignment): fix signedness warnings
(_dbus_string_append_unichar): ditto

2005-04-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message-util.c (_dbus_message_test): fix signedness
warning

- * glib/dbus-glib-tool.c (main): fix warning
- * glib/dbus-binding-tool-glib.c (generate_glue): fix warning
- * dbus/dbus-connection.c (dbus_connection_read_write_dispatch): add a new function that can be used in simple applications that don't have a main loop and are willing to block

2005-04-05 David Zeuthen <davidz@redhat.com>

Fix https://bugs.freedesktop.org/show_bug.cgi?id=2889

- * glib/dbus-gmain.c:
 - (io_handler_destroy_source): Remove from list of IO handlers of the ConnectionSetup object
 - (timeout_handler_destroy_source): -do- for timeout handlers
 - (io_handler_source_finalized): Don't remove from list since we now do that in io_handler_destroy_source(). Renamed from io_handler_source_destroyed
 - (timeout_handler_source_destroyed): -do- for timeout handlers
 - (connection_setup_free): It is now safe to iterate over all IO and timeout handlers as the _destroy_source removes them from the list synchronously

2005-03-30 Havoc Pennington <hp@redhat.com>

- * configure.in: change check to gtk 2.4
- * tools/dbus-viewer.c (name_combo_changed_callback): remove gtk_combo_box_get_active_text() usage to decrement GTK requirement to 2.4

2005-03-29 John (J5) Palmieri <johnp@redhat.com>

- * News: Update 0.32
- * HACKING: Fixed realease instructions. configure.in should be updated to the next release by the person who made the last release.

2005-03-29 John (J5) Palmieri <johnp@redhat.com>

- * python/lvalue_cast_post_process.py - removed. Patch has been submitted to Pyrex maintainers that fixes gcc4.0 errors
- * python/Makefile.am: removed references to lvalue_cast_post_process.py

2005-03-24 Daniel Reed <n@ml.org>

* tools/Makefile.am: Make print-introspect and dbus-bus-introspect.xml building conditional on HAVE_GLIB.

2005-03-22 John (J5) Palmieri <johnp@redhat.com>

* tools/Makefile.am: Patch by Colin Walters that fixes distcheck

* dbus/dbus-userdb.c, dbus/dbus-userdb-util.c: Add patch we have had in Red Hat packages for a while but for some reason never got merged upstream

 (_dbus_is_a_number): New checks if a string can be converted to a number and does the conversion if it can

 (_dbus_user_database_lookup): Add check to see if the given username is a udi. This allows udi's to be used instead of usernames in the config file.

 (_dbus_user_database_lookup_group): Add check to see if the given groupname is a gdi. This allows gdi's to be used instead of groupnames in the config file.

2005-03-21 John (J5) Palmieri <johnp@redhat.com>

* python/lvalue_cast_post_process.py - added post processor to fix Pyrex code so that it compiles with gcc4.0

* python/Makefile.am: Added lvalue_cast_post_process.py to EXTRA_DIST

 run dbus_bindings.c through lvalue_cast_post_process.py and copy the results back to dbus_binding.c

2005-03-20 Colin Walters <walters@verbum.org>

Patch suggested by Inguva Rajasekhar <ringuva@novell.com>.

* configure.in: Require GTK+ 2.6.

2005-03-20 Colin Walters <walters@verbum.org>

* Makefile.am (SUBDIRS, DIST_SUBDIRS): Build tools before test.

2005-03-17 Tom Parker <palfrey@tevp.net>

* dbus/dbus-userdb.c (_dbus_user_database_lookup): Don't print DBUS_UID_UNSET; instead print passed username. Also be sure to actually use gid looked up in cache.

* dbus/dbus-userdb-util.c (_dbus_user_database_lookup_group):
Ditto
for DBUS_GID_UNSET and groupname.

2005-03-17 Colin Walters <walters@verbum.org>

- * bus/print-introspect.c: Move to tools/.
- * bus/run-with-tmp-session-bus.sh: Ditto.

- * glib/Makefile.am (dbus-glib-bindings.h): Move generation to tools/Makefile.am.

- * test/glib/run-test.sh: Update to handle move of run-with-tmp-session-bus.sh.

- * test/glib/test-service-glib.c: Update to handle move of dbus-glib-bindings.h.

- * tools/print-introspect.c: Moved here from bus/, and ported to GLib bindings.

- * tools/run-with-tmp-session-bus.sh: Moved here from bus/.

- * tools/Makefile.am: Generate dbus-glib-bindings.h and dbus-bus-introspect.xml here.

- * tools/.cvsignore, glib/.cvsignore, bus/.cvsignore: Update.

2005-03-17 Colin Walters <walters@verbum.org>

- * bus/driver.c (write_args_for_direction): Use _dbus_string_get_const_data to retrieve string; _dbus_string_get_const_data_len doesn't actually return a NULL-terminated substring.

- * test/glib/test-service-glib.c: Include dbus-glib-bindings.h. (main): Change to use org_freedesktop_DBus_request_name instead of using g_proxy_begin_call/end_call.

2005-03-15 Joe Shaw <joeshaw@novell.com>

- * mono/ProxyBuilder.cs (BuildFinalizer): Fix some invalid IL when generating the finalizer. Fixes from Ben Maurer.

2005-03-12 Joe Shaw <joeshaw@novell.com>

- * mono/BusDriver.cs: Update method names: ListServices becomes ListNames; GetOwner becomes GetNameOwner.

- * mono/ProxyBuilder.cs (BuildFinalizer): Need to load arg 0

onto the eval stack when removing the delegate.

2005-03-12 Joe Shaw <joeshaw@novell.com>

* mono/dbus-sharp.dll.config.in: Don't hardcode 0 for LT_CURRENT. Set it to the autoconf variable.

* mono/ProxyBuilder.cs: Add a finalizer to the generated proxy classes that disconnects the signal handler delegate from the service object. Fixes a big leak of proxy objects on the client side of things. Patch from Ben Maurer <bmaurer@ximian.com>

2005-03-12 Colin Walters <walters@verbum.org>

* bus/driver.c (write_args_for_direction): New function, parses a type signature into arguments and outputs to XML.

(bus_driver_handle_introspect): Use it instead of hardcoding XML for certain signatures.

* bus/Makefile.am (dbus-bus-introspect.xml): Add dependency on dbus-daemon.

* glib/dbus-glib-tool.c (main): Parse ignore_unsupported argument, pass it to dbus_binding_tool_output_glib_client.

* glib/dbus-binding-tool-glib.c
(generate_client_glue): Protect against multiple inclusion.
(dbus_binding_tool_output_glib_client): Add G_BEGIN_DECLS/G_END_DECLS.

* glib/dbus-binding-tool-glib.c (compute_client_method_name): Change to just take iface prefix directly.
(write_formal_parameters): Clarify error message.
(check_supported_parameters): New function; checks to see type signatures of method parameters are supported.
(generate_client_glue): Handle ignore_unsupported flag.
(dbus_binding_tool_output_glib_client): Handle ignore_unsupported parameter.

* glib/Makefile.am (dbus-glib-bindings.h): Pass --ignore-unsupported by default until glib bindings support arrays.

2005-03-11 Colin Walters <walters@verbum.org>

* glib/Makefile.am: Generate dbus-glib-bindings.h and install it.

* bus/print-introspect.c: New file; prints introspection data for a given name and object path.

- * bus/run-with-tmp-session-bus.sh: New file, refactored from test/glib/run-test.sh. Creates a temporary session bus and runs another program.
- * test/glib/run-test.sh: Refactor to invoke run-with-tmp-session-bus.sh.
- * bus/driver.c (bus_driver_handle_introspect): Fix to print new introspection format. Also change to use DBUS_TYPE_x_AS_STRING macros instead of hardcoding.
- * glib/.cvsignore, bus/.cvsignore, test/glib/.cvsignore: Update.

2005-03-11 Joe Shaw <joeshaw@novell.com>

- * dbus/dbus-connection.c (dbus_connection_send_with_reply):
Remove
this unref; it doesn't match up evenly in some codepaths.
(_dbus_connection_block_pending_call): Unref at every exitpoint;
this evenly matches with the ref near the top of this function.

2005-03-09 Joe Shaw <joeshaw@novell.com>

- * dbus/dbus-object-tree.c
(_dbus_object_tree_unregister_and_unlock): If checks are enabled and we try to unregister a path that's not registered, still go through the process of unlocking and don't just return.

2005-03-09 Colin Walters <walters@verbum.org>

- * glib/dbus-gproxy.c (dbus_g_proxy_invoke): New method; calls to this are generated for client-side wrappers. Invokes a D-BUS method and returns reply values.
- * glib/dbus-binding-tool-glib.c (write_args_sig_for_direction):
New
function; writes signature string for argument direction.
(write_args_for_direction): Change to pass input values directly instead of via address, and fix indentation.
(generate_client_glue): Change to invoke dbus_g_proxy_invoke.
- Also
make generated wrappers inlineable.
- * dbus/dbus-message.c (dbus_message_iter_get_fixed_array): Add note about using dbus_type_is_fixed.
- * dbus/dbus-marshal-basic.c (_dbus_type_is_fixed): Moved to dbus/dbus-signature.c as dbus_type_is_fixed.

All callers updated.

* dbus/dbus-signature.c (dbus_type_is_fixed): Moved here from dbus/dbus-marshal-basic.c:_dbus_type_is_fixed.

* dbus/dbus-signature.h: Prototype.

* glib/dbus-binding-tool-glib.c (compute_marshall_name): Fix error printf code.

* test/glib/test-dbus-glib.c (main): Be sure to clear error as appropriate instead of just freeing it.
(main): Free returned strings using g_free.

* test/glib/Makefile.am (test-service-glib-glue.h)
(test-service-glib-bindings.h): Add dependency on dbus-binding-tool.

* glib/dbus-gvalue.c (MAP_BASIC): Refactored from MAP_BASIC_INIT; simply maps a simple D-BUS type to GType.
(dbus_dbus_type_to_gtype): Function which maps D-BUS type to GType.
(dbus_gvalue_init): Just invoke dbus_dbus_type_to_gtype and initialize the value with it.
(dbus_gvalue_binding_type_from_type): Unused, delete.
(dbus_gvalue_demarshal): Switch to hardcoding demarshalling for various types instead of unmarshalling to value data directly. Remove can_convert boolean.
(dbus_gvalue_marshal): Remove duplicate initialization; switch to returning directly instead of using can_convert boolean.
(dbus_gvalue_store): New function; not related to D-BUS per-se. Stores a GValue in a pointer to a value of its corresponding C type.

* glib/dbus-gvalue.h: Remove dbus_gvalue_binding_type_from_type, add dbus_gvalue_store.

2005-03-08 Joe Shaw <joeshaw@novell.com>

Fix a bunch of lifecycle and memory management problems in the mono bindings.

* mono/Arguments.cs (Arguments): Implement IDisposable

* mono/Bus.cs (Bus): Don't allow public instantiation. This is strictly a static class.

* mono/Connection.cs: Move the DBusObjectPathVTable and associated delegates into this file.
(Connection): Implement IDisposable.
(Dispose): Disconnect the connection and set the raw connection pointer to IntPtr.Zero.
(~Connection): Call Dispose().

(RegisterObjectPath): Added. Manages the registration of object paths so we can cleanly disconnect them at dispose/finalize time.
(UnregisterObjectPath): Ditto.
(set_RawConnection): Unregister all of the object paths when changing the underlying DBusConnection. Add them back onto the new connection, if any.

* mono/Handler.cs: Don't implement IDisposable; it doesn't use any more unmanaged resources anymore, so it's not necessary. Move all the DBusObjectPathVTable stuff out of here.
(Handler): Save references to our delegates so that they don't get finalized. Call Connection.RegisterObjectPath() instead of dbus_connection_register_object_path() directly.
(Message_Called): Dispose the message after we're finished with it.

* mono/Message.cs (Message): Implement IDisposable.
(Dispose): Dispose the Arguments, and set the RawMessage to IntPtr.Zero.
(SendWithReplyAndBlock): We own the ref to the reply that comes back from dbus_connection_send_with_reply_and_block() so add a comment about that and unref it after we've constructed a managed MethodReturn class around it. Fixes a big, big leak.

* mono/ProxyBuilder.cs: Reflect into Message to get the Dispose method.
(BuildSignalHandler): After we've sent the Signal message, dispose of it.
(BuildMethod): Dispose of the method call and reply messages after we've sent the message and extracted the data we want from the reply.

* mono/Service.cs (UnregisterObject): Don't call handler.Dispose() anymore.
(Service_FilterCalled): Dispose of the message after we're finished with it.

2005-03-08 Joe Shaw <joeshaw@novell.com>

* dbus/dbus-connection.c (dbus_connection_send_with_reply): After we attach our pending call to the connection, unref it. Fixes a leak.

* mono/Connection.cs (set_RawConnection): Disconnect our filter and match callbacks from the old connection and reconnect them to the new connection, if any.

* mono/DBusType/Array.cs: "Code" is a static member, so don't use "this" to refer to it. Fix for stricter checking in Mono 1.1.4.

* mono/DBusType/ObjectPath.cs (Append): Don't leak the object path that we pass into unmanaged code.

* mono/DBusType/String.cs (Append): Don't leak the string that we pass into unmanaged code.

2005-03-07 John (J5) Palmieri <johnp@redhat.com>

* NEWS: Update for 0.31

* configure.in: Release 0.31
add LT_CURRENT, LT_REVISION, LT_AGE for easy soname bumping

* qt/Makefile.am: fixed build

* dbus/Makefile.am: soname bump for libdbus

* glib/Makefile.am: soname bump for libdbus-glib

2005-03-05 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c:
(pseudorandom_generate_random_bytes_buffer): fix to have no
return
value

(_dbus_generate_random_bytes_buffer): fix return value

* dbus/dbus-sysdeps-util.c: s/GETPWNNAME/GETPWNAM/ so configure
checks actually work, from Tom Parker <fdo@tevp.net>

2005-03-01 Colin Walters <walters@verbum.org>

* test/glib/test-dbus-glib.c (lose, lose_gerror): Utility
functions copied from dbus-glib-tool.c.
(main): Convert lots of error code to use them.
Also add some testing for introspection bits.

2005-03-01 Colin Walters <walters@verbum.org>

* doc/TODO: Remove introspection signature TODO.

2005-02-27 Colin Walters <walters@verbum.org>

* glib/dbus-gidl.c (property_info_get_type, arg_info_get_type):
Change return value to const char * instead of int so we can do
full signatures.

(struct PropertyInfo, struct ArgInfo): Store char *.

(property_info_new, arg_info_new): Update parameters, strdup.

(property_info_unref, arg_info_unref): Free.

- * glib/dbus-gidl.h: Update prototypes.
- * glib/dbus-gparser.c (basic_type_from_string): Delete.
(validate_signature): New function, just validates signature and sets GError.
(parse_property, parse_arg): Invoke validate_signature. Store signature instead of just type code.
- * glib/dbus-gvalue.c (base_type_from_signature): New utility function to return a primary type for a signature, dropping information about types in container types.
(dbus_gvalue_genmarshal_name_from_type)
(dbus_gvalue_binding_type_from_type)
(dbus_gvalue_ctype_from_type): Update to take full signature instead of type code.
(dbus_gtype_to_dbus_type): Moved here from glib/dbus-gobject.c.
- * glib/dbus-gvalue.h: Update prototypes for above.
- * glib/dbus-gobject.c (gtype_to_dbus_type): Moved to glib/dbus-gvalue.c as dbus_gtype_to_dbus_type.
(introspect_properties, introspect_signals, write_interface): Update to handle signatures, and remove usage of _dbus_gutils_type_to_string.
(handle_introspect): Print out type codes instead of e.g. "string"
in hardcoded introspection XML; also use x_AS_STRING constants instead of hardcoding in string.
- * glib/dbus-glib-tool.c (pretty_print): Handle signature change to string. Remove usage of _dbus_gutils_type_to_string.
- * glib/dbus-gutils.c (_dbus_gutils_type_to_string): Delete.
- * glib/dbus-gutils.h (_dbus_gutils_type_to_string): Update for deletion.
- * glib/dbus-binding-tool-glib.c (compute_marshall)
(compute_marshall_name, generate_glue): Handle signature change to string.
(write_formal_parameters, write_args_for_direction): Ditto, and remove FIXME.
- * tools/dbus-tree-view.c (type_to_string): Delete.
(info_set_func_text): Update to print full signatures.
- * test/glib/test-service-glib.xml: Change types to new introspection format.

- * doc/TODO: remove the "guid" item
- * test/glib/test-profile.c (no_bus_thread_func): use open_private
(with_bus_thread_func): use open_private
- * dbus/dbus-connection.c (dbus_connection_open_private): new
function that works like the old dbus_connection_open()
(dbus_connection_open): now returns an existing connection if
possible
- * dbus/dbus-server-unix.c (handle_new_client_fd_and_unlock): pass
through the GUID to the transport
- * dbus/dbus-server.c (_dbus_server_init_base): keep around the
GUID in hex-encoded form.
- * dbus/dbus-server-debug-pipe.c (_dbus_transport_debug_pipe_new):
pass GUID argument in to the transport
- * dbus/dbus-transport-unix.c (_dbus_transport_new_for_fd): add
guid argument
- * dbus/dbus-transport.c (_dbus_transport_init_base): add guid
argument
- * dbus/dbus-auth.c (_dbus_auth_server_new): add guid argument

2005-02-25 Havoc Pennington <hp@redhat.com>

- * doc/dbus-specification.xml: document the GUID thing
- * dbus/dbus-server.c (_dbus_server_init_base): initialize a
globally unique ID for the server, and put a "guid=hexencoded"
field in the address
- * dbus/dbus-bus.c: fix missing #include of dbus-threads-
internal.h
- * dbus/dbus-message.c: ditto
- * dbus/dbus-dataslot.c: ditto
- * dbus/dbus-list.c: ditto
- * dbus/dbus-internals.h: wait, just include
dbus-threads-internal.h here
- * dbus/dbus-string.c (_dbus_string_copy_to_buffer): move back for
use in main library

* dbus/dbus-sysdeps.c (_dbus_generate_random_bytes_buffer): new function

2005-02-24 Colin Walters <walters@verbum.org>

* test/glib/Makefile.am (EXTRA_DIST): Add test-service-glib.xml

2005-02-24 John (J5) Palmieir <johnp@redhat.com>

* glib/Makefile.am: added dbus-gobject.h to sources list so distcheck doesn't fail

2005-02-24 Havoc Pennington <hp@redhat.com>

* dbus/dbus-server.c, dbus/dbus-server-unix.c: change semantics so

you must disconnect before unref, since locking and other things are screwed up otherwise. Fix assorted other locking stuff.

* dbus/dbus-signature.c (dbus_signature_iter_get_element_type): fix compilation

* dbus/dbus-threads-internal.h: move the mutex/condvar wrappers into a private header and don't export from the library

* throughout - call _dbus_thread_stuff vs. dbus_thread_stuff

2005-02-24 Colin Walters <walters@verbum.org>

* dbus/dbus-signature.c: New file; implements various functions related to type signatures. Includes an iterator for parsing, validation functions.

(dbus_type_is_basic): Moved here from dbus-marshal-basic.c:dbus_type_is_basic.

(dbus_type_is_container): Moved here from dbus-marshal-basic.c:_dbus_type_is_container.

All callers of _dbus_type_is_container and _dbus_type_is_basic updated, and include dbus-signature.h.

* dbus/dbus-signature.h: New file; prototypes for the above.

* dbus/Makefile.am (DBUS_LIB_SOURCES): Add dbus-signature.c, dbus-signature.h.

* dbus/dbus-marshal-basic.c (map_type_char_to_type): New utility function factored out of _dbus_first_type_in_signature.

first (_dbus_first_type_in_signature_c_str): New function; returns type code for a type signature character.

- * dbus/dbus-marshall-basic.h: Prototype `_dbus_first_type_in_signature_c_str`, handle function moves.
- * dbus/dbus-marshall-recursive.h: Export `_dbus_type_signature_next`.
- * dbus/dbus-marshall-recursive.c (`_dbus_type_signature_next`): New function; skips to next complete type in type signature. Implemented using previous `skip_one_complete_type`. Now `skip_one_complete_type` just delegates to `_dbus_type_signature_next`.
- * dbus/dbus-marshall-basic.c (`_dbus_type_is_basic`): Moved to `dbus-signature.c`
(`_dbus_type_is_container`): Ditto.
- * doc/dbus-specification.xml: Update introspection sample to use real type signatures.
- * dbus/dbus-test.h: Prototype signature test.
- * dbus/dbus-test.c (`dbus_internal_do_not_use_run_tests`): Run signature tests.
- * dbus/dbus-protocol.h (`DBUS_ERROR_INVALID_SIGNATURE`): New error.

2005-02-23 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx.in (`PendingCall::get_reply`):
`s/dbus_pending_call_get_reply/dbus_pending_call_steal_reply`

2005-02-21 Colin Walters <walters@verbum.org>

- * dbus/dbus-test-main.c (`main`): Take optional specific test argument.
- * dbus/dbus-test.c (`run_test`): New function, runs a test function with no data directory.
(`run_data_test`): Like above, but takes data directory.
(`dbus_internal_do_not_use_run_tests`): Take specific test argument. Replace lots of cut n' paste code with `run_test` and `run_data_test`.
- * dbus/dbus-test.h: Update prototype for `dbus_internal_do_not_use_run_tests`.

2005-02-20 Havoc Pennington <hp@redhat.com>

Fix bugs reported by Daniel P. Berrange

- * dbus/dbus-server.c (`_dbus_server_unref_unlocked`): new function

(protected_change_watch): new function
(_dbus_server_toggle_watch, _dbus_server_remove_watch)
(_dbus_server_add_watch): change to work like the
dbus-connection.c equivalents; like those, probably kind of
busted, but should at least mostly work for now
(dbus_server_disconnect): drop the lock if we were already
disconnected, patch from Daniel P. Berrange

* dbus/dbus-server.c (_dbus_server_toggle_timeout)
(_dbus_server_remove_timeout, _dbus_server_add_timeout): all the
same stuff

* doc/TODO: todo about unscrewing this mess

2005-02-19 Colin Walters <walters@verbum.org>

* glib/dbus-binding-tool-glib.c
(dbus_binding_tool_output_glib_server): Fix iochannel
refcounting.

* glib/dbus-glib-tool.c: Include dbus-glib-tool.h, as well
as errno.h and sys/stat.h.
(lose): New function, prints error with
newline and exits.
(lose_gerror): Similar, but takes GError for message.
(main): Add --output argument to specify output file to write to,
instead of always printing to stdout. In this mode, determine
timestamps on source files to see whether any are newer than the
target file. If not, exit. Also convert a number of error
messages to use lose (since it's shorter), and switch to using
g_io_channel_shutdown.

2005-02-19 Havoc Pennington <hp@redhat.com>

* glib/dbus-gobject.c
(_dbus_glib_marshal_dbus_message_to_gvalue_array): add docs

* glib/dbus-glib.c: fix doxygen warnings

* glib/dbus-gparser.c (parse_annotation): error if an annotation
is found on an <arg>

2005-02-17 Colin Walters <walters@verbum.org>

* glib/dbus-gobject.h: Don't export
_dbus_glib_marshal_dbus_message_to_gvalue_array.

* glib/dbus-gobject.c
(_dbus_glib_marshal_dbus_message_to_gvalue_array): Do rename.
(invoke_object_method): Handle it.

* glib/dbus-gproxy.c (marshal_dbus_message_to_g_marshaller):

Handle rename.

2005-02-17 Colin Walters <walters@verbum.org>

- * bus/.cvsignore, doc/.cvsignore
- * test/data/valid-service-files/.cvsignore, test/glib/.cvsignore: Update.

2005-02-17 Colin Walters <walters@verbum.org>

- * dbus/dbus-protocol.h (DBUS_SERVICE_ORG_FREEDESKTOP_DBUS): Rename to DBUS_SERVICE_DBUS.
- (DBUS_PATH_ORG_FREEDESKTOP_DBUS): Rename to DBUS_PATH_DBUS.
- (DBUS_PATH_ORG_FREEDESKTOP_LOCAL): Rename to DBUS_PATH_LOCAL. Change the value from "org.freedesktop.Local" to "org.freedesktop.DBus.Local".
- (DBUS_INTERFACE_ORG_FREEDESKTOP_DBUS): Rename to DBUS_INTERFACE_DBUS.
- (DBUS_INTERFACE_ORG_FREEDESKTOP_INTROSPECTABLE): Rename to DBUS_INTERFACE_INTROSPECTABLE. Change the value from "org.freedesktop.Introspectable" to "org.freedesktop.DBus.Introspectable".
- (DBUS_INTERFACE_ORG_FREEDESKTOP_PROPERTIES): Rename to DBUS_INTERFACE_PROPERTIES. Change the value from "org.freedesktop.Properties" to "org.freedesktop.DBus.Properties".
- (DBUS_INTERFACE_ORG_FREEDESKTOP_PEER): Rename to DBUS_INTERFACE_PEER. Change the value from "org.freedesktop.Peer" to "org.freedesktop.DBus.Peer".
- (DBUS_INTERFACE_ORG_FREEDESKTOP_LOCAL): DBUS_INTERFACE_LOCAL. Change the value from "org.freedesktop.Local" to "org.freedesktop.DBus.Local".

All other users of those constants have been changed.

- * bus/driver.c (bus_driver_handle_introspect): Use constants.
- * glib/dbus-gobject.c (handle_introspect): Use constants.
- * doc/dbus-faq.xml, doc/dbus-specification.xml: Update for rename.

2005-02-17 Colin Walters <walters@verbum.org>

- * glib/dbus-gparser.c (struct Parser): Add in_annotation boolean. (parse_node, parse_interface, parse_method, parse_signal) (parse_property, parse_annotation): Lose if we're currently in an annotation.
- (parse_annotation): New function.
- (parser_start_element, parser_end_element): Handle annotation.

(parse_method, parse_interface): Remove support for c_name attribute, switch to annotations.

* glib/dbus-gidl.h (interface_info_get_binding_names) (method_info_get_binding_names) (interface_info_get_binding_name, method_info_get_binding_name) (interface_info_set_binding_name, method_info_set_binding_name): Remove. (interface_info_get_annotations, method_info_get_annotations) (interface_info_get_annotation, method_info_get_annotation) (interface_info_add_annotation, method_info_add_annotation): Prototype.

* glib/dbus-gidl.c (struct InterfaceInfo): Substitute "annotations" for "bindings". (struct MethodInfo): Ditto. Straightfoward conversion of binding methods into annotation methods as prototyped.

* glib/dbus-glib-tool.c (pretty_print): Print annotations.

* glib/dbus-binding-tool-glib.h (DBUS_GLIB_ANNOTATION_C_SYMBOL): Define.

* glib/dbus-binding-tool-glib.c (gather_marshallers, generate_glue): Use new annotation API.

* doc/introspect.dtd: Fix a number of DTD syntax errors. Add annotation element.

* doc/dbus-specification.xml: Discuss introspection annotations, include list of well-known annotations.

* test/glib/test-service-glib.xml: Make validate against new DTD.

2005-02-17 Colin Walters <walters@verbum.org>

This patch is based on initial work from Paul Kuliniewicz <kuliniew@purdue.edu>.

* glib/dbus-gvalue.c (dbus_gvalue_init): New function; move initialization of GValue from dbus type to here. (dbus_gvalue_genmarshal_name_from_type): New function; generates a string for the "glib-genmarshal" program from a DBus type. (dbus_gvalue_binding_type_from_type): New function; turns a DBus type into the C name for it we use in the glib bindings.

(dbus_gvalue_ctype_from_type): New function; maps a Dbus type into a glib C type (not GValue).
(dbus_gvalue_demarshal): invoke dbus_gvalue_init.

* glib/dbus-gutils.c (_dbus_gutils_wincaps_to_uscore): Moved here from dbus-gobject.c.

* glib/dbus-gutils.h: Prototype it.

* glib/dbus-gproxy.c: Include new dbus-gobject.h.
(marshal_dbus_message_to_g_marshaller): Use new shared function dbus_glib_marshal_dbus_message_to_gvalue_array.

* glib/dbus-gparser.c (parse_interface, parse_method): Handle c_name attribute.
Will be changed once we have annotations.

* glib/dbus-gobject.c: Change info_hash_mutex from GStaticMutex to GStaticRWLock. Callers updated.
(wincaps_to_uscore): Move to dbus-gutils.c. Callers updated.
(string_table_next): New function for iterating over zero-terminated string value array.
(string_table_lookup): New function; retrieves specific entry in array.
(get_method_data): New function; look up method data in object data chunk.
(object_error_domain_prefix_from_object_info)
(object_error_code_from_object_info): New functions, but not implemented yet.
(method_interface_from_object_info): New function; retrieve interface name.
(method_name_from_object_info): New function; retrieve method name.
(method_arg_info_from_object_info): New function; retrieve argument data.
(arg_iterate): New function; iterates over serialized argument data.
(method_dir_signature_from_object_info): New function; returns a GString holding type signature for arguments for just one direction (input or output).
(method_input_signature_from_object_info)
(method_output_signature_from_object_info): New functions.
(dbus_glib_marshal_dbus_message_to_gvalue_array): New shared function;
converts dbus message arguments into a GValue array. Used for both signal handling and method invocation.
(struct DbusGlibWriteInterfaceData): New utility structure.
(write_interface): New function; generate introspection XML for

```

    an interface.
    (introspect_interfaces): New function; gathers all interface-
>methods,
    generates introspection XML for them.
    (handle_introspect): Invoke introspect_interfaces.
    (get_object_property): Be sure to zero-initialize stack-allocated
GValue.
    (lookup_object_and_method): New function; examines an incoming
message
    and attempts to match it up (via interface, method name, and
argument
    signature) with a known object and method.
    (gerror_domaincode_to_dbus_error_name): New function; converts a
GError domain and code into a DBus error name. Needs GError data
added to object introspection to work well.
    (gerror_to_dbus_error_message): Creates a DBusMessage error
return from
    GError.
    (invoke_object_method): New function to invoke an object method
looked up via lookup_object_and_method. Parses the incoming
message, turns it into a GValue array, then invokes the
marshaller
    specified in the DBusGMethodInfo. Creates a new message with
either return values or error message as appropriate.
    (gobject_message_function): Invoke lookup_object_and_method and
invoke_object_method.

* glib/dbus-glib-tool.c: Include dbus-binding-tool-glib.h.
(enum DBusBindingOutputMode): New enum for binding output modes.
(pretty_print): Print binding names.
(dbus_binding_tool_error_quark): GError bits.
(version): Fix typo.
(main): Create GIOChannel for output. Parse new --mode argument,
possible values are "pretty-print", "glib-server", "glib-client".
Use mode to invoke appropriate function.

* glib/dbus-gobject.h: Prototype
dbus_glib_marshal_dbus_message_to_gvalue_array.

* glib/dbus-glib-tool.h: New header, just includes GError bits
for now.

* glib/dbus-gidl.c (struct InterfaceInfo): Add bindings
hashtable;
    maps binding style to name.
    (struct MethodInfo): Ditto.
    (get_hash_keys, get_hash_key): Utility function, returns keys for
a GHashTable.
    (interface_info_new, method_info_new): Initialize bindings.
    (interface_info_unref, method_info_unref): Destroy bindings.
    (method_info_get_binding_names, method_info_get_binding_name)

```



```
(interface_info_get_binding_names,  
interface_info_get_binding_name):  
    Functions for retrieving binding names.  
(method_info_set_binding_name, interface_info_set_binding_name):  
    Functions for setting binding names.  
  
* glib/dbus-binding-tool-glib.h: New file, has prototypes  
for glib binding generation.  
  
* glib/dbus-binding-tool-glib.c: New file, implements server-side  
and client-side glib glue generation.  
  
* glib/Makefile.am (dbus_binding_tool_SOURCES): Add  
dbus-binding-tool-glib.c, dbus-binding-tool-glib.h,  
dbus-glib-tool.h.  
  
* dbus/dbus-glib.h (struct DBusGMethodMarshaller): Remove in  
favor  
of using GClosureMarshal directly.  
(struct DBusGObjectInfo): Add n_infos member.  
  
* test/glib/test-service-glib.xml: New file; contains  
introspection data  
for MyTestObject used in test-service-glib.c.  
  
* test/glib/test-service-glib.c (enum MyObjectError): New GError  
enum.  
(my_object_do_nothing, my_object_increment,  
my_object_throw_error)  
(my_object_uppercase, my_object_many_args): New test methods.  
(main): Use dbus_g_object_class_install_info to include generated  
object  
info.  
  
* test/glib/Makefile.am: Generate server-side glue for test-  
service-glib.c,  
as well as client-side bindings.  
  
* test/glib/test-dbus-glib.c: Include test-service-glib-  
bindings.h.  
(main): Activate TestSuiteGLibService; test invoke a bunch of its  
methods  
using both the dbus_gproxy stuff directly as well as the  
generated bindings.
```

2005-02-15 Havoc Pennington <hp@redhat.com>

```
* dbus/dbus-connection.c (dbus_connection_dispatch): always  
complete a pending call, don't run filters first.  
  
* glib/dbus-gproxy.c (dbus_g_proxy_end_call): change to use  
dbus_pending_call_steal_reply
```

* dbus/dbus-pending-call.c (dbus_pending_call_block): just call
_dbus_connection_block_pending_call
(dbus_pending_call_get_reply): change to steal_reply and return a
ref

* dbus/dbus-connection.c
(dbus_connection_send_with_reply_and_block): port to work in
terms
of DbusPendingCall
(_dbus_connection_block_pending_call): replace block_for_reply
with this

2005-02-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-userdb-util.c (_dbus_user_database_lookup_group):
properly handle looking up group information by name; fix
from j@bootlab.org

2005-02-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_return_message)
(dbus_connection_borrow_message): hold dispatch lock while
message
is outstanding
(_dbus_connection_block_for_reply): hold dispatch lock while we
block for the reply, so nobody steals our reply
(dbus_connection_pop_message): hold the dispatch lock while we
pluck the message

2005-02-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c (_dbus_connection_acquire_dispatch)
(_dbus_connection_release_dispatch)
(_dbus_connection_acquire_io_path)
(_dbus_connection_release_io_path): make the mutex and condvar
control access to the "acquired" flag. Drop the connection lock
while waiting on the condvar. Hopefully these are baby steps in
roughly the right direction.

2005-02-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c: use separate mutexes for the condition
variables; this is some kind of baseline for sanity, but the
condition variables still aren't used correctly afaict

2005-02-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-object-tree.c (handle_default_introspect_and_unlock):
fix a double-unlock

* dbus/dbus-connection.c

(`_dbus_connection_detach_pending_call_unlocked`): add this

Initial semi-correct pass through to fix thread locking; there are

still some issues with the condition variable paths I'm pretty sure

* `dbus/dbus-server.c`: add a mutex on `DBusServer` and appropriate lock/unlock calls

* `dbus/dbus-connection.c`

(`_dbus_connection_do_iteration_unlocked`):

rename to add `_unlocked`

(`struct DBusConnection`): move "`dispatch_acquired`" and "`io_path_acquired`" to use only one bit each.

(`CONNECTION_LOCK`, `CONNECTION_UNLOCK`): add checks with

`!DBUS_DISABLE_CHECKS`

(`dbus_connection_set_watch_functions`): hacky fix to reentrancy

(`_dbus_connection_add_watch`, `_dbus_connection_remove_watch`)

(`_dbus_connection_toggle_watch`, `_dbus_connection_add_timeout`)

(`_dbus_connection_remove_timeout`)

(`_dbus_connection_toggle_timeout`): drop lock when calling out to user functions; done in a hacky/bad way.

(`_dbus_connection_send_and_unlock`): add a missing unlock

(`_dbus_connection_block_for_reply`): add a missing unlock

* `dbus/dbus-transport.c` (`_dbus_transport_get_is_authenticated`): drop lock in a hacky probably unsafe way to call out to user function

2005-02-12 Havoc Pennington <hp@redhat.com>

* `tools/dbus-tree-view.c` (`info_set_func_text`): display more details on args

* `bus/driver.c` (`bus_driver_handle_list_services`): list the bus driver

* `glib/dbus-gparser.c` (`parse_arg`): generate an arg name if none is supplied

* `glib/dbus-gidl.c` (`signal_info_get_n_args`): new function

(`method_info_get_n_args`): new function

2005-02-12 Havoc Pennington <hp@redhat.com>

* `bus/driver.c` (`bus_driver_handle_introspect`): add introspection for bus driver

2005-02-12 Havoc Pennington <hp@redhat.com>

* bus/driver.c: put the signature of each bus driver method in the table of handlers and check it on incoming calls; this isn't really useful, but going to add introspect support in a minute.

2005-02-11 Joe Shaw <joeshaw@novell.com>

* mono/Connection.cs: The unpredictability of finalizers in mono prevents us from deterministically disconnecting the filters from the Service class's finalizer, so move tracking of filters and matches here. Add API for that.

* mono/Service.cs: Remove the code, add code which calls the methods now on the Connection class.

2005-02-11 John (J5) Palmieri <johnp@redhat.com>

* python/dbus.py (class Sender): added to support dbus signals better
(Bus::add_signal_receiver): added expand_args parameter which defaults to True. When expand args is True the signal handler will pass the message arguments as parameters to the signal handler. If False revert to previous behavior where the signal handler must get the argument list from the message. This is to help port applications like HAL that have a tendency to send variable length argument lists.
self._match_rule_to_receivers is now a dict of dicts.
(Bus::remove_signal_receiver): pop handler off the dict instead of removing it from a list
(Bus::signal_func): change signal handlers so that interface, signal_name, service, path and message are packed into a Sender object and that is passed to the handler. If expand_args is True extract the args list from the message and append it to the parameter list

* python/dbus_bindings.pyx.in (class Signature): added to support signature types
(MessageIter::__init__): changed iteration limit to match D-BUS
(MessageIter::get*): added INT16, UINT16, SIGNATURE, DICT_ENTRY, STRUCT and VARIANT type support
(MessageIter::python_value_to_dbus_sig): made recursive to support recursive types
(MessageIter::append*): added Signature, dict, tuple support

* python/examples/example-client.py: added examples of getting tuples

and dicts

- * python/examples/example-service.py: added examples of sending tuples and dicts
- * python/examples/example-signal-recipient.py: Fixed to handle new signal callback format

2005-02-10 Havoc Pennington <hp@redhat.com>

- * test/glib/test-dbus-glib.c (main): fix so this test doesn't fail (call dbus_g_proxy_add_signal)
- * dbus/dbus-server-unix.c (_dbus_server_new_for_tcp_socket): escape the hostname (_dbus_server_new_for_domain_socket): escape the path
- * dbus/dbus-address.c (dbus_address_escape_value): new (dbus_address_unescape_value): new (dbus_parse_address): unescape values
- * dbus/dbus-string.c (_dbus_string_append_byte_as_hex): new function
- * doc/dbus-specification.xml: explain how to escape values in addresses

2005-02-10 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message-factory.c (generate_special): modify test to avoid using a non-basic dict key
- * dbus/dbus-marshal-validate-util.c: add test for the below
- * doc/dbus-specification.xml: require that dict keys are a basic type
- * dbus/dbus-marshal-validate.c (_dbus_validate_signature_with_reason): require that dict key is a basic type

2005-02-10 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-object-tree.c (handle_default_introspect_and_unlock): change to be _and_unlock instead of _unlocked
- * dbus/dbus-connection.c

(`_dbus_connection_send_preallocated_unlocked_no_update`): rename to have `no_update` so we can find this bug quickly in future

2005-02-10 Havoc Pennington <hp@redhat.com>

* `dbus/dbus-message-util.c` (`verify_test_message`): tests for string array

* `dbus/dbus-message.c` (`dbus_message_append_args_valist`): add support for arrays of string/signature/path

2005-02-10 Joe Shaw <joeshaw@novell.com>

* `dbus/dbus-connection.c`
(`_dbus_connection_queue_received_message_link`, `_dbus_connection_message_sent`): Add the path to the verbose output.
(`_dbus_connection_send_preallocated_and_unlock`): Added. Calls `_dbus_connection_send_preallocated_unlocked()`, updated the dispatch status, and unlocks. Fixes a bug where certain situations (like a broken pipe) could cause a Disconnect message to not be sent, tricking the bus into thinking a service was still there when the process had quit.
(`_dbus_connection_send_preallocated`): Call `_dbus_connection_send_preallocated_and_unlock()`.
(`_dbus_connection_send_and_unlock`): Added. Calls `_dbus_connection_send_preallocated_and_unlock()`.
(`dbus_connection_send`): Call `_dbus_connection_send_and_unlock()`.
(`dbus_connection_send_with_reply`): Update the dispatch status and unlock.

* `mono/Service.cs` (`~Service`): Added. Removes the filter so that we don't get unmanaged code calling back into a GCed delegate.
(`RemoveFilter`); Added.

2005-02-09 John (J5) Palmieri <johnp@redhat.com>

* `dbus/dbus-message.c` (`dbus_message_iter_open_container`):
- Removed check for iterator type being an array because `get_arg_type` does not work with writer iterators
- Pass NULL to `_dbus_type_writer_recurse` if signature is NULL

2005-02-07 Havoc Pennington <hp@redhat.com>

* `doc/dbus-specification.xml`: some more language cleanups; add stuff about how to deal with invalid protocol and extension points; add `_` to allowed chars in auth commands; add `EXTENSION_` auth command prefix

2005-02-06 Havoc Pennington <hp@redhat.com>

- * s/expected/required/ in a couple places for clarity

2005-02-07 Colin Walters <walters@verbum.org>

- * bus/selinux.c (bus_selinux_allows_send): Handle NULL for sender or proposed_recipient.

2005-02-06 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message-factory.c (generate_special): more tests

- * dbus/dbus-marshall-validate.c (validate_body_helper): detect array length that exceeds the maximum

2005-02-05 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message-factory.c (generate_special): more test cases, increasing coverage

- * dbus/dbus-marshall-validate.c (validate_body_helper): return the reason why a signature was invalid

- * dbus/dbus-marshall-header.c (load_and_validate_field): fix to skip the length of the string before we look at it in validation

- * dbus/dbus-string-util.c (_dbus_string_test): add tests for equal_substring

- * dbus/dbus-message.c (_dbus_message_loader_new): default max_message_length to DBUS_MAXIMUM_MESSAGE_LENGTH

2005-02-05 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-marshall-validate.c (validate_body_helper): fix crash if the signature of a variant was empty
(_dbus_validate_signature_with_reason): catch "(a)" (array inside struct with no element type)

- * dbus/dbus-message-factory.c (generate_uint32_changed): add more mangled messages to break things

2005-02-04 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gproxy.c (dbus_g_proxy_disconnect_signal): use g_quark_try_string() so it actually can return 0
(dbus_g_proxy_connect_signal): ditto

2005-02-04 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c (dbus_g_proxy_emit_remote_signal): fix a bogus warning
(tristring_from_message): assert cleanly on null path/interface (should not be possible though I decided later)
(dbus_g_proxy_dispose): move proxy manager unregistration here
(DBUS_G_PROXY_DESTROYED): add this macro, and use it in a bunch of
g_return_if_fail() checks

2005-02-04 Havoc Pennington <hp@redhat.com>

* doc/Makefile.am (EXTRA_DIST): add DTDs to makefile
* doc/introspect.dtd: add introspect.dtd from David A. Wheeler (with some minor changes)
* doc/dbus-specification.xml: add deprecated attribute to introspection format

2005-01-31 Havoc Pennington <hp@redhat.com>

* glib/dbus-gproxy.c: rewrite how signals work again, this time I think it's sort of right

2005-01-30 Havoc Pennington <hp@redhat.com>

* tools/dbus-viewer.c: kind of half-ass hook up the option menu.

2005-01-30 Havoc Pennington <hp@redhat.com>

* tools/dbus-names-model.c: dynamically watch NameOwnerChanged
* autogen.sh: change to autotools 1.9
* glib/dbus-gproxy.c: completely change how signals work
(dbus_g_proxy_add_signal): new function to specify signature of a signal
(dbus_g_proxy_emit_received): marshal the dbus message to GValues,
and g_warning if the incoming message has the wrong signature.

2005-01-30 Havoc Pennington <hp@redhat.com>

* tools/dbus-names-model.c (have_names_notify): fix this
* dbus/dbus-message.c (_dbus_message_iter_get_args_valist): clean up the string array handling a bit

2005-01-30 Havoc Pennington <hp@redhat.com>

* glib/dbus-glib.c (dbus_g_pending_call_set_notify): new function
(dbus_g_pending_call_cancel): new function

- * dbus/dbus-glib.h: move GType decls for connection/message here;
- * dbus/dbus-glib.c: move all the g_type and ref/unref stuff in here, just kind of rationalizing how we handle all that

- * tools/dbus-names-model.c: new file for a tree model listing the services on a bus

- * tools/dbus-tree-view.c (model_new): use proper typing on the model rows

2005-01-30 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gmain.c: add a custom GSource back that just checks whether the message queue has anything in it; otherwise, there are cases where we won't see messages in the queue since there was no IO visible to the glib main loop

- * dbus/dbus-connection-internal.h (_DBUS_DEFAULT_TIMEOUT_VALUE): increase default message timeout to 25 seconds

2005-01-30 Havoc Pennington <hp@redhat.com>

- * test/glib/test-profile.c (no_bus_stop_server): remove the warning about the g_warning that I just fixed

- * glib/dbus-gmain.c: rewrite the main loop stuff to avoid the custom source, seems to be a lot easier to understand and work better.

2005-01-30 Havoc Pennington <hp@redhat.com>

I think this main loop thing is conceptually broken, but here are some band aids. I'll maybe rewrite it in a minute.

- * glib/dbus-gmain.c (add_timeout): timeout stuff doesn't use the custom GSource, so don't pass it in; confusing (gsource_server_finalize, gsource_connection_finalize): add finalize handlers that remove all the watches.

2005-01-30 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gobject.c (introspect_properties): fix the XML generated

- * dbus/dbus-message.c (dbus_message_unref): add an in_cache flag which effectively detects the use of freed messages

- * glib/dbus-gobject.c (handle_introspect): modify and return the reply message instead of the incoming message

* dbus/dbus-object-tree.c (handle_default_introspect_unlocked):
gee, maybe it should SEND THE XML instead of just making a string
and freeing it again ;-)

* tools/dbus-print-message.c (print_message): improve printing of
messages

* configure.in: add debug-glib.service to the output

2005-01-30 Havoc Pennington <hp@redhat.com>

dbus-viewer introspected and displayed the bus driver

* dbus/dbus-object-tree.c
(object_tree_test_iteration): add tests for a handler registered
on "/"

* dbus/dbus-object-tree.c
(_dbus_decompose_path): fix to handle path "/" properly
(run_decompose_tests): add tests for path decomposition

* glib/dbus-gutils.c (_dbus_gutils_split_path): fix to handle "/"
properly

* glib/dbus-gobject.c (handle_introspect): fix quotes

* test/glib/run-test.sh: support launching the bus, then running
dbus-viewer

* test/glib/test-service-glib.c (main): put in a trivial gobject
subclass and register it on the connection

* bus/driver.c (bus_driver_handle_introspect): implement
introspection of the bus driver service

* dbus/dbus-protocol.h: add #defines for the XML namespace,
identifiers, doctype decl

* bus/driver.c (bus_driver_handle_get_service_owner): handle
attempts to get owner of DBUS_SERVICE_ORG_FREEDESKTOP_DBUS by
returning the service unchanged.
(bus_driver_handle_message): remove old check for reply_serial in
method calls, now the message type deals with that
(bus_driver_handle_message): handle NULL interface

* glib/dbus-gproxy.c (dbus_g_proxy_get_bus_name): new function

* glib/dbus-gloader-expat.c (description_load_from_string): allow
-1 for len

* tools/dbus-viewer.c: add support for introspecting a service on

a bus

- * glib/dbus-gproxy.c (dbus_g_pending_call_ref): add
(dbus_g_pending_call_unref): add

2005-01-29 Havoc Pennington <hp@redhat.com>

- * tools/dbus-tree-view.c: add support for displaying properties.
(run dbus-viewer with an introspect xml file as arg, then resize
the window so the tree elements show up, not sure what that is)

- * glib/dbus-gobject.c (handle_introspect): return
org.freedesktop.Properties and org.freedesktop.Introspectable
interfaces when we are introspected.

- * doc/dbus-specification.xml: allow empty interface name when
Get/Set a property

2005-01-29 Havoc Pennington <hp@redhat.com>

- * glib/Makefile.am: rename dbus-glib-tool to dbus-binding-tool;
though it uses glib, it could be extended for any binding in
principle

- * glib/dbus-gobject.c (gobject_message_function): change to the
new way properties work

- * dbus/dbus-protocol.h: add the new interfaces

- * doc/dbus-specification.xml: document the introspection format,
Introspectable interface, and add an org.freedesktop.Properties
interface.

- * glib/dbus-gparser.c: add support for a <property> element

- * glib/dbus-gidl.c: add PropertyInfo

- * glib/dbus-gobject.c (handle_introspect): put the outermost
<node> outside the signal and property descriptions.
(introspect_properties): export properties as <property> rather
than as method calls

2005-01-28 Havoc Pennington <hp@redhat.com>

- * doc/TODO, doc/dbus-specification.xml: spec and TODO tweaks
related to authentication protocol

2005-01-28 John (J5) Palmieri <johnp@redhat.com>

- * python/dbus_bindings.pyx.in: Updated to handle new D-BUS type
system

 - BUS_ACTIVATION -> BUS_STARTER

```

- DBUS_BUS_ACTIVATION -> DBUS_BUS_STARTER
- class MessageIter (__init__): Added recursion checking
  so we throw a nice error instead of just disconnecting from
the
  bus.
  (get): Added arg_type parameter for recursion.
  Removed the nil type
  Added signiture type placeholder (not implemented)
  Added struct type placeholder (not implemented)
  Added varient type placeholder (not implemented)
  Commented out dict type for now
  (get_element_type): renamed from get_array_type
  (get_*): changed to use the dbus_message_iter_get_basic API
  (get_*_array): removed in favor of recursive get_array
method
  (get_array): new recursive method which calls get to
marshal
  the elements of the array
  (value_to_dbus_sig): New method returns the corrasponding
  dbus signiture to a python value
  (append): Comment out dict handling for now
  Handle lists with the new recursive API
  Comment out None handling for now
  (append_nil): removed
  (append_*): changed to use dbus_message_iter_append_basic
API
  (append_*_array): removed in favor of recursive
append_array
  method
  (__str__): Make it easier to print out recursive iterators
  for debugging
  - class Message (__str__): moved type inspection to the
  MessageIter class' __str__ method
  (get_iter): Added an append parameter wich defaults to
False
  If True use the new API's to create an append iterator

* python/dbus.py: Update to use new bindings API
  - TYPE_ACTIVATION -> TYPE_STARTER
  - class Bus (_get_match_rule): GetServiceOwner ->
GetNameOwner
  - class ActivationBus -> class StarterBus
  - class RemoteObject (__call__): get an append iterator
  - (_dispatch_dbus_method_call): get an append iterator
  - class Object (emit_signal): get an append iterator

* python/examples/: Fixed up the examples to work with the new
API

2005-01-28 Joe Shaw <joeshaw@novell.com>

* configure.in: Bump version up to 0.30.

```

* HACKING: Add a release item to bump the version number up after a release.

2005-01-28 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: update to describe 16-bit types and dict entries

* dbus/dbus-marshal-basic.c (_dbus_unpack_uint16): fix broken assertion

* dbus/dbus-protocol.h (DBUS_TYPE_DICT_ENTRY): add DICT_ENTRY as
a
type

* dbus/dbus-marshal-recursive.c: implement

2005-01-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-arch-deps.h.in: add 16/32-bit types

* configure.in: find the right type for 16 and 32 bit ints as
well
as 64

* dbus/dbus-protocol.h (DBUS_TYPE_INT16, DBUS_TYPE_UINT16): add the 16-bit types so people don't have to stuff them in 32-bit or byte arrays.

2005-01-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c: byteswap the message if you init an iterator to read/write from it

* dbus/dbus-marshal-byteswap.c: new file implementing
_dbus_marshal_byteswap()

* dbus/dbus-marshal-basic.c: add _dbus_swap_array()

2005-01-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-marshal-validate-util.c: break this out (and fix build, apparently - nobody noticed?)

2005-01-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-marshal-recursive.h: remove todo comment

2005-01-25 Joe Shaw <joeslaw@novell.com>

* Land the mono binding changes to conform to the new APIs.

* mono/Makefile.am: Remove Custom.cs, DBusType/Custom.cs, DBusType/Dict.cs, and DBusType/Nil.cs from the build.

* mono/Arguments.cs (GetCodeAsString): Added. Returns the dbus type code as a string.
(InitAppending): Rename dbus_message_append_iter_init() to dbus_message_iter_init_append().

* mono/BusDriver.cs: Rename ServiceEventHandler to NameOwnerChangedHandler. Rename GetServiceOwner to GetOwner. Rename ServiceOwnerChanged to NameOwnerChanged.

* mono/Connection.cs: Rename BaseService to UniqueName, and the underlying C call.

* mono/Custom.cs: Removed. The CUSTOM type has been removed.

* mono/Service.cs: Rename Exists to HasOwner, internally rename dbus_bus_acquire_service() to dbus_bus_request_name().

* mono/DBusType/Array.cs (ctor): Use Type.GetElementType() instead of Type.UnderlyingSystemType to get the correct element type for the array.
(ctor): Update code for new APIs: use
dbus_message_iter_recurse(),
dbus_message_get_{element|arg}_type() instead of
dbus_message_iter_init_array_iterator().
(Append): Replace dbus_message_iter_append_array() with
dbus_message_iter_open_container() and
dbus_message_iter_close_container().

* mono/DBusType/Custom.cs, mono/DBusType/Nil.cs: Removed. These types have been removed.

* mono/DBusType/*.cs: Replace calls of
dbus_message_iter_get_[type]() to dbus_message_iter_get_basic(),
but specify the type in the DllImport extern declaration. Ditto
for dbus_message_iter_append_[type]() ->
dbus_message_iter_append_basic().

* mono/example/BusListener.cs: Update for ServiceEventHandler -> NameOwnerChangedHandler.

2005-01-25 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in: Rename of methods and bindings
- get_base_service -> get_unique_name
- bus_get_base_service -> bus_get_unique_name
- dbus_bus_get_base_service -> dbus_bus_get_unique_name
- ACTIVATION_REPLY_ACTIVATED -> DBUS_START_REPLY_SUCCESS

```
    - ACTIVATION_REPLY_ALREADY_ACTIVE ->
DBUS_START_REPLY_ALREADY_RUNNING
    - bus_activate_service -> bus_start_service_by_name
    - dbus_bus_activate_service ->
dbus_bus_start_service_by_name
    - bus_acquire_service -> bus_request_name
    - dbus_bus_acquire_service -> dbus_bus_request_name
    - bus_service_exists -> bus_name_has_owner
    - dbus_bus_service_exists -> dbus_bus_name_has_owner

* python/dbus.py: Rename of methods
    - activate_service -> start_service_by_name
    - bus_acquire_service -> bus_request_name
    - ACTIVATION_REPLY_ACTIVATED -> START_REPLY_SUCCESS
    - ACTIVATION_REPLY_ALREADY_ACTIVE ->
START_REPLY_ALREADY_RUNNING
```

2005-01-24 Joe Shaw <joeshaw@novell.com>

* dbus/dbus-connection.c (dbus_connection_dispatch): Print out the signature for the method that can't be found.

* dbus/dbus-message.c (dbus_message_iter_init): To check to see if the message has any arguments, we need to call `_dbus_type_reader_get_current_type()`, not `_dbus_type_reader_has_next()`.

2005-01-24 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message-factory.c: more testing of message validation

* dbus/dbus-protocol.h (DBUS_MINIMUM_HEADER_SIZE): move to this header

2005-01-23 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message-factory.c, dbus/dbus-message-util.c: get this all working, not many tests in the framework yet though

2005-01-22 Havoc Pennington <hp@redhat.com>

* doc/dbus-faq.xml, doc/dbus-tutorial: add a FAQ and update tutorial, based on work from David Wheeler.

2005-01-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-bus.c: add more `return_if_fail` checks

* dbus/dbus-message.c (load_message): have the "no validation"

mode (have to edit the code to toggle the mode for now though)

- * dbus/dbus-marshal-header.c (_dbus_header_load): have a mode that skips all validation; I want to use this at least for benchmark baseline, I'm not sure if it should be a publicly-available switch.

2005-01-21 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gmain.c: don't put the GLib bindings in the same toplevel doxygen group as the low-level API stuff
- * dbus/dbus.h: note that libdbus is the low-level API

2005-01-20 Havoc Pennington <hp@redhat.com>

- * update-dbus-docs.sh: script to update docs on the web site, only works for me though. neener.

2005-01-20 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps.c (_dbus_poll): amazingly, trying to compile code can reveal bugs in it

2005-01-20 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps.c (_dbus_poll): fix several bugs in the select() version, patches from Tor Lillqvist

2005-01-20 Havoc Pennington <hp@redhat.com>

- * doc/dbus-tutorial.xml: replace > with >
- * bus/services.c (bus_registry_acquire_service): validate the name and return a better error if it's no good.
- * doc/dbus-specification.xml: note NO_AUTO_START change
- * dbus/dbus-protocol.h (DBUS_HEADER_FLAG_NO_AUTO_START): change from AUTO_START, we're toggling the default
- * bus/dispatch.c: adapt the tests to change of auto-start default

2005-01-18 Havoc Pennington <hp@redhat.com>

- * rename dbus-daemon-1 to dbus-daemon throughout

2005-01-18 Havoc Pennington <hp@redhat.com>

* Throughout, grand renaming to strip out the use of "service", just say "name" instead (or "bus name" when ambiguous). Did not change the internal code of the message bus itself, only the programmer-facing API and messages.

* doc/dbus-specification.xml: further update the message bus section

* bus/config-parser.c (all_are_equiv): fix bug using freed string in error case

2005-01-17 Havoc Pennington <hp@redhat.com>

* dbus/dbus-types.h: remove 16-bit types since we don't use them ever

* dbus/dbus-marshall-validate.c (_dbus_validate_path): disallow any "invalid name character" not only non-ASCII

* doc/dbus-specification.xml: further update spec, message bus parts are still out-of-date but the marshaling etc. stuff is now accurate-ish

2005-01-17 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: partially update spec

2005-01-17 Havoc Pennington <hp@redhat.com>

* Throughout, align variant bodies according to the contained type, rather than always to 8. Should save a fair bit of space in message headers.

* dbus/dbus-marshall-validate.c (_dbus_validate_body_with_reason): fix handling of case where p == end

* doc/TODO: remove the dbus_bool_t item and variant alignment items

2005-01-17 Havoc Pennington <hp@redhat.com>

* dbus/dbus-types.h: hardcode dbus_bool_t to 32 bits

* Throughout: modify DBUS_TYPE_BOOLEAN to be a 32-bit type instead of an 8-bit type. Now dbus_bool_t is the type to use whenever you are marshaling/unmarshaling a boolean.

2005-01-16 Havoc Pennington <hp@redhat.com>

This is about it on what can be disabled/deleted from libdbus easily, back below 150K anyhow. Deeper cuts are more work than just turning the code off as I've done here.

- * dbus/dbus-marshal-basic.c (`_dbus_pack_int32`): we don't need the signed int convenience funcs

- * dbus/dbus-internals.c (`_dbus_verbose_real`): omit when not in verbose mode

- * dbus/dbus-string-util.c, dbus/dbus-string.c: more breaking things out of libdbus

- * dbus/dbus-sysdeps.c, dbus/dbus-sysdeps-util.c: same

- * dbus/dbus-hash.c: purge the `TWO_STRINGS` crap (well, make it tests-enabled-only, though it should probably be deleted)

- * dbus/dbus-message-util.c: same stuff

- * dbus/dbus-auth-util.c: same stuff

2005-01-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-userdb-util.c: split out part of dbus-userdb.c

- * dbus/dbus-sysdeps.c (`_dbus_uid_from_string`): move here to pave way for stripping down dbus-userdb.c stuff included in libdbus. Rename `_dbus_parse_uid` for consistency.

2005-01-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-internals.c (`_dbus_real_assert`): print the function name the assertion failed in

- * dbus/dbus-internals.h (`_dbus_return_if_fail`)
(`_dbus_return_val_if_fail`): assert that the name of the function containing the check doesn't start with `'_'`, since we only want

to

use checks on public functions

- * dbus/dbus-connection.c (`_dbus_connection_ref_unlocked`): change checks to assertions

- * dbus/dbus-marshal-header.c (`_dbus_header_set_field_basic`): change checks to asserts for private function

- * dbus/dbus-message.c (`_dbus_message_set_serial`): checks to asserts for private function

- * dbus/dbus-marshal-recursive.c (`skip_one_complete_type`): remove broken assertion that was breaking make check

(`_dbus_type_reader_array_is_empty`): remove this rather than fix it, was only used in assertions

2005-01-16 Havoc Pennington <hp@redhat.com>

* `test/unused-code-gc.py`: hacky script to find code that's used only by the bus (not `libdbus`) or used only by tests or not used at all. It has some false alarms, but looks like we can clean up a lot of size from `libdbus`.

* `dbus/dbus-sysdeps.c`, `dbus/dbus-sysdeps-utils.c`, `dbus/Makefile.am`: initially move 10K of binary size out of `libdbus`

2005-01-16 Havoc Pennington <hp@redhat.com>

* Add and fix docs according to Doxygen warnings throughout source.

* `dbus/dbus-marshal-recursive.c`
(`_dbus_type_reader_array_is_empty`): change this to just call `array_reader_get_array_len()` and make it static

* `dbus/dbus-message.c` (`dbus_message_iter_get_element_type`):
rename
from `get_array_type`
(`dbus_message_iter_init_append`): rename from `append_iter_init`

* `dbus/dbus-marshal-recursive.c`
(`_dbus_type_reader_get_element_type`): rename from
`_dbus_type_reader_get_array_type`

2005-01-15 Havoc Pennington <hp@redhat.com>

* `test/glib/test-profile.c` (`with_bus_server_filter`): fix crash

* `dbus/dbus-marshal-basic.c` (`_dbus_unpack_uint32`): inline as
macro
when `DBUS_DISABLE_ASSERT`
(`_dbus_marshall_set_basic`): be sure we align for the string length

* `dbus/dbus-marshal-recursive.c` (`skip_one_complete_type`): make this look faster

* `dbus/dbus-string.c` (`_dbus_string_get_const_data_len`): add an inline macro version
(`_dbus_string_set_byte`): provide inline macro version

2005-01-15 Havoc Pennington <hp@redhat.com>

* Land the new message args API and type system.

This patch is huge, but the public API change is not really large. The set of D-BUS types has changed somewhat, and the arg "getters" are more geared toward language bindings; they don't make a copy, etc.

There are also some known issues. See these emails for details on this huge patch:

<http://lists.freedesktop.org/archives/dbus/2004-December/001836.html>

<http://lists.freedesktop.org/archives/dbus/2005-January/001922.html>

- * dbus/dbus-marshal-*: all the new stuff
- * dbus/dbus-message.c: basically rewritten
- * dbus/dbus-memory.c (check_guards): with "guards" enabled, init freed blocks to be all non-nul bytes so using freed memory is less likely to work right
- * dbus/dbus-internals.c (_dbus_test_oom_handling): add DBUS_FAIL_MALLOC=N environment variable, so you can do DBUS_FAIL_MALLOC=0 to skip the out-of-memory checking, or DBUS_FAIL_MALLOC=10 to make it really, really, really slow and thorough.
- * qt/message.cpp: port to the new message args API (operator<<): use str.utf8() rather than str.unicode() (pretty sure this is right from the Qt docs?)
- * glib/dbus-gvalue.c: port to the new message args API
- * bus/dispatch.c, bus/driver.c: port to the new message args API
- * dbus/dbus-string.c (_dbus_string_init_const_len): initialize the "locked" flag to TRUE and align_offset to 0; I guess we never looked at these anyhow, but seems cleaner.
- * dbus/dbus-string.h (_DBUS_STRING_ALLOCATION_PADDING): move allocation padding macro to this header; use it to implement (_DBUS_STRING_STATIC): ability to declare a static string.
- * dbus/dbus-message.c (_dbus_message_has_type_interface_member): change to return TRUE if the interface is not set.
- * dbus/dbus-string.[hc]: move the D-BUS specific validation stuff to dbus-marshal-validate.[hc]

from * dbus/dbus-marshall-basic.c (_dbus_type_to_string): move here
dbus-internals.c

* dbus/Makefile.am: cut over from dbus-marshall.[hc]
to dbus-marshall-*. [hc]

* dbus/dbus-object-tree.c (_dbus_decompose_path): move this
function here from dbus-marshall.c

2005-01-12 Joe Shaw <joeshaw@novell.com>

* NEWS: Update for 0.23.

* configure.in: Release 0.23.

2005-01-12 Joe Shaw <joeshaw@novell.com>

* mono/Makefile.am, mono/example/Makefile.am: Always build the
dbus DLL with --debug. Clean up after the .mdb files this leaves
behind.

* mono/doc/Makefile.am: Need to uninstall the docs on "make
uninstall"

* mono/Arguments.cs (GetDBusTypeConstructor): If the type
is an enum, get the enum's underlying type. Another mono
1.1.3 fix.

2005-01-11 Joe Shaw <joeshaw@novell.com>

Patch from Sjoerd Simons <sjoerd@luon.net>

* mono/Makefile.am, mono/example/Makefile.am: Don't redefine
DESTDIR. It breaks stuff.

2005-01-11 Joe Shaw <joeshaw@novell.com>

Patch from Tabet Ingo <tabet@ximian.com>

* mono/DBusType/Array.cs (Get): Get the underlying element type
by calling type.GetElementType(). The code previously depended on
broken Mono behavior, which was fixed in Mono 1.1.3.

* mono/DBusType/Dict.cs (constructor): Fix the parameters for
Activator.CreateInstance() so that the class's constructor is
called with the right parameters.

2005-01-11 Joe Shaw <joeshaw@novell.com>

Patch from Timo Teras <ext-timo.teras@nokia.com>

* dbus/dbus-connection.c
(_dbus_connection_queue_received_message_link): Call
_dbus_connection_remove_timeout() instead of the _locked()
variant, since it's always called from
_dbus_connection_handle_watch(), which handles the locking.
Removed the _locked() variant since it's no longer used.

2005-01-03 Havoc Pennington <hp@redhat.com>

* dbus/dbus-internals.h: I'm an idiot, _dbus_assert certainly can
return

2004-12-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-internals.h: add _DBUS_GNUC_NORETURN to _dbus_assert

2005-01-03 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_sysdeps_test): fix using == on
floating point

* dbus/dbus-string.c (_dbus_string_insert_alignment): new
function

2005-01-02 Havoc Pennington <hp@redhat.com>

* dbus/dbus-internals.h (_DBUS_ALIGN_OFFSET): new macro

2005-01-01 Havoc Pennington <hp@redhat.com>

* configure.in: add -Wfloat-equal

2005-01-01 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.h: add _DBUS_DOUBLES_BITWISE_EQUAL macro,
for a variety of reasons '==' doesn't do this.

2004-12-31 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (_dbus_string_equal_substrings): new
function
I keep wishing I had

2004-12-30 John (J5) Palmieri <johnp@redhat.com>

* python/dbus.py:
s/ACTIVATION_REPLY_ACTIVE/ACTIVATION_REPLY_ACTIVATED

2004-12-30 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in: Change
DBUS_ACTIVATION_REPLY_ACTIVATED
and DBUS_ACTIVATION_REPLY_ALREADY_ACTIVE to match the values in
dbus-protocol.h. Because they are defines and not enums they are
not
autogenerated.

2004-12-26 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in (bus_activate_service): Bind
dbus_bus_activate_service

* python/dbus.py (Bus.activate_service): activate a service on
the
bus.

2004-12-24 Havoc Pennington <hp@redhat.com>

* test/decode-gcov.c: change to use .gcno and .gcda files, but
the
file format has also changed and I haven't adapted to that yet

* Makefile.am: load .gcno files from latest gcc

2004-12-23 John (J5) Palmieri <johnp@redhat.com>

* Patch from Rob Taylor <robtaylor@fastmail.fm>

* python/dbus_bindings.pyx.in (bus_get_unix_user): New
lowlevel binding

* python/dbus.py (get_unix_user): Added binding to
call dbus_bindings.bus_get_unix_user

* python/extract.py: Modified the proto_pat regex to
handle unsigned long

2004-12-21 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/make-dbus-glib-error-enum.sh: omit the function keyword
for
better POSIX compliance.

2004-12-19 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (_dbus_string_insert_4_aligned)
(_dbus_string_insert_8_aligned): new functions

* dbus/dbus-string.c (_dbus_string_alloc_space): new function

2004-12-18 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (_dbus_string_validate_ascii): use ISASCII

macro

* dbus/dbus-message.c: fix a comment, and add a still-unused not-implemented function

* dbus/dbus-marshal.h: fix comment

* dbus/dbus-internals.h (_DBUS_ISASCII): new macro

2004-12-17 Joe Shaw <joeshaw@novell.com>

* mono/DBusType/Byte.cs, mono/DBusType/Int32.cs, mono/DBusType/Int64.cs, mono/DBusType/UInt32.cs, mono/DBusType/UInt64.cs: Use Enum.GetUnderlyingType() instead of Type.UnderlyingSystemType to get the actual system type underneath. This code previously depended on the broken Mono behavior, which was fixed in 1.1.3.

2004-11-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.h (_dbus_string_get_byte): inline when asserts are disabled

(_dbus_string_get_const_data): inline when asserts are disabled

* dbus/dbus-message.c: record the _dbus_current_generation of creation so we can complain if dbus_shutdown() is used improperly.

Do this only if checks are enabled.

* dbus/dbus-connection.c: ditto

2004-11-26 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c: add with_bus mode to profile echoes that go through the bus.

* test/glib/run-test.sh: add ability to run test-profile

* bus/dbus-daemon-1.1.in: fix to say that SIGHUP causes partial config file reload.

2004-11-26 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c: clean up how the fake_malloc_overhead thing was implemented

2004-11-26 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c: tweak a bit, add support for some made-up minimal malloc overhead with plain sockets, since in real life some sort of buffers are unavoidable thus we could count them in the theoretical best case

2004-11-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_cache_or_finalize): fix bug where I was trying to cache one too many messages

2004-11-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c: reimplement message cache as an array which makes the cache about twice as fast and saves maybe 1.5% overall

2004-11-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-threads.c (init_global_locks): forgot to put the message cache lock here

2004-11-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (struct DBusMessage): put the locked bit and the "char byte_order" next to each other to save 4 bytes (dbus_message_new_empty_header): reduce preallocation, since the message cache should achieve a similar effect (dbus_message_cache_or_finalize, dbus_message_get_cached): add a message cache that keeps a few DBusMessage around in a pool, another 8% speedup or so.
* dbus/dbus-dataslot.c (_dbus_data_slot_list_clear): new function

2004-11-25 Havoc Pennington <hp@redhat.com>

* dbus/dbus-transport-unix.c (unix_do_iteration): if we're going to write, without reading or blocking, try it before the poll() and skip the poll() if nothing remains to write. This is about a 3% speedup in the echo client/server

2004-11-25 Havoc Pennington <hp@redhat.com>

The primary change here is to always write() once before adding the write watch, which gives us about a 10% performance increase.

* dbus/dbus-transport-unix.c: a number of modifications to cope with removing messages_pending (check_write_watch): properly handle DBUS_AUTH_STATE_WAITING_FOR_MEMORY; adapt to removal of messages_pending stuff (check_read_watch): properly handle WAITING_FOR_MEMORY and AUTHENTICATED cases (unix_handle_watch): after writing, see if the write watch can be removed

```

(unix_do_iteration): assert that write_watch/read_watch are
non-NULL rather than testing that they aren't, since they
aren't allowed to be NULL. check_write_watch() at the end so
we add the watch if we did not finish writing (e.g. got EAGAIN)

* dbus/dbus-transport-protected.h: remove messages_pending call,
since it resulted in too much inefficient watch adding/removing;
instead we now require that the transport user does an iteration
after queueing outgoing messages, and after trying the first
write() we add a write watch if we got EAGAIN or exceeded our
max bytes to write per iteration setting

* dbus/dbus-string.c (_dbus_string_validate_signature): add this
function

* dbus/dbus-server-unix.c (unix_finalize): the socket name was
freed and then accessed, valgrind flagged this bug, fix it

* dbus/dbus-message.c: fix several bugs where HEADER_FIELD_LAST
was taken
as the last valid field plus 1, where really it is equal to the
last valid field. Corrects some message corruption issues.

* dbus/dbus-mainloop.c: verbosity changes

* dbus/dbus-keyring.c (_dbus_keyring_new_homedir): handle OOM
instead of aborting in one of the test codepaths

* dbus/dbus-internals.c (_dbus_verbose_real): fix a bug that
caused not printing the pid ever again if a verbose was missing
the newline at the end
(_dbus_header_field_to_string): add HEADER_FIELD_SIGNATURE

* dbus/dbus-connection.c: verbosity changes;
(dbus_connection_has_messages_to_send): new function
(_dbus_connection_message_sent): no longer call transport-
>messages_pending
(_dbus_connection_send_preallocated_unlocked): do one iteration
to
try to write() immediately, so we can avoid the write watch. This
is the core purpose of this patchset
(_dbus_connection_get_dispatch_status_unlocked): if disconnected,
dump the outgoing message queue, so nobody will get confused
trying to send them or thinking stuff is pending to be sent

* bus/test.c: verbosity changes

* bus/driver.c: verbosity/assertion changes

* bus/dispatch.c: a bunch of little tweaks to get it working
again
because this patchset changes when/where you need to block.

```

2004-11-23 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c: modify to accept a plain_sockets argument in which case it will bench plain sockets instead of libdbus, for comparison purposes.

2004-11-22 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c (N_CLIENT_THREADS): run multiple threads for more time, so sysprof can get a grip on it.

* dbus/dbus-string.c (_dbus_string_validate_utf8): remove pointless variable

2004-11-13 Havoc Pennington <hp@redhat.com>

* test/glib/test-profile.c: fix this thing up a bit

* dbus/dbus-message.c (dbus_message_new_empty_header): increase preallocation sizes by a fair bit; not sure if this will be an overall performance win or not, but it does reduce reallocs.

* dbus/dbus-string.c (set_length, reallocate_for_length): ignore the test hack that forced constant realloc if asserts are disabled, so we can profile sanely. Sprinkle in some _DBUS_UNLIKELY() which are probably pointless, but before I noticed the real performance problem I put them in.

(_dbus_string_validate_utf8): micro-optimize this thing a little bit, though callgrind says it didn't help; then special-case ascii, which did help a lot; then be sure we detect nul bytes as invalid, which is a bugfix.

(align_length_then_lengthen): add some more _DBUS_UNLIKELY superstition; use memset to nul the padding instead of a manual loop.

(_dbus_string_get_length): inline this as a macro; it showed up in the profile because it's used for loop tests and so forth

2004-11-10 Colin Walters <walters@verbum.org>

* dbus/dbus-spawn.c (check_babysit_events): Handle EINTR, for extra paranoia.

2004-11-09 Colin Walters <walters@verbum.org>

* dbus/dbus-string.c (_dbus_string_get_length): New function, writes DBusString to C buffer.

* dbus/dbus-string.h: Prototype it.

* dbus/dbus-message.c (dbus_message_type_to_string): New

function, converts message type into C string.

* dbus/dbus-message.h: Prototype it.

* bus/selinux.c (bus_selinux_check): Take source pid, target pid, and audit data. Pass audit data to avc_has_perm.

(log_audit_callback): New function, appends extra audit information.

(bus_selinux_allows_acquire_service): Also take service name, add it to audit data.

(bus_selinux_allows_send): Also take message type, interface, method member, error name, and destination, and add them to audit data.

(log_cb): Initialize func_audit.

* bus/selinux.h (bus_selinux_allows_acquire_service) (bus_selinux_allows_send): Update prototypes

* bus/services.c (bus_registry_acquire_service): Pass service name to bus_selinux_allows_acquire_service.

* bus/bus.c (bus_context_check_security_policy): Pass additional audit data. Move assignment of dest to its own line.

2004-11-07 Colin Walters <walters@verbum.org>

* dbus/dbus-transport-unix.c (do_authentication): Always initialize auth_completed.

2004-11-07 Colin Walters <walters@verbum.org>

* bus/bus.c (load_config): Break into three separate functions: process_config_first_time_only, process_config_every_time, and process_config_postinit.

(process_config_every_time): Move call of bus_registry_set_service_context_table into process_config_postinit.

(process_config_postinit): New function, does any processing that needs to happen late in initialization (and also on reload).

(bus_context_new): Instead of calling load_config, open config parser here and call process_config_first_time_only and process_config_every_time directly. Later, after we have forked but before changing UID, invoke bus_selinux_full_init, and then call process_config_postinit.

(bus_context_reload_config): As in bus_context_new, load parse file inside here, and call process_config_every_time and process_config_postinit.

* bus/services.h, bus/services.c
(bus_registry_set_service_context_table): Rename from bus_registry_set_sid_table. Take string hash from config parser, and convert them here into SIDs.

* bus/config-parser.c (struct BusConfigParser): Have config parser only store a mapping of service->context string.
(merge_service_context_hash): New function.
(merge_included): Merge context string hashes instead of using bus_selinux_id_table_union.
(bus_config_parser_new): Don't use bus_selinux_id_table_new; simply create a new string hash.
(bus_config_parser_unref): Unref it.
(start_selinux_child): Simply insert strings into hash, don't call bus_selinux_id_table_copy_over.

* bus/selinux.h, bus/selinux.c (bus_selinux_id_table_union)
(bus_selinux_id_table_copy_over): Delete.

2004-11-03 Colin Walters <walters@verbum.org>

* bus/selinux.c (bus_selinux_pre_init): Kill some unused variables.

2004-11-03 Colin Walters <walters@verbum.org>

* bus/test-main.c (test_pre_hook): Fix test logic, thanks Joerg Barfurth <Joerg.Barfurth@Sun.COM>.

2004-11-02 Colin Walters <walters@redhat.com>

* bus/selinux.c (bus_selinux_init): Split into two functions, bus_selinux_pre_init and bus_selinux_post_init.
(bus_selinux_pre_init): Just determine whether SELinux is enabled.
(bus_selinux_post_init): Do everything else.

* bus/main.c (main): Call bus_selinux_pre_init before parsing config file, and bus_selinux_post_init after. This ensures that we don't lose the policyreload notification thread that bus_selinux_init created before forking previously.

* bus/test-main.c (test_pre_hook): Update for split.

2004-10-31 Owen Fraser-Green <owen@discobabe.net>

Patch from Johan Fischer <linux@fischaz.com>

* mono/doc/Makefile.am (install-data-local): Added directory install for DESTDIR

2004-10-29 Colin Walters <walters@redhat.com>

* dbus/dbus-sysdeps.h (_dbus_become_daemon): Also take parameter for fd to write pid to.

* dbus/dbus-sysdeps.c (_dbus_become_daemon): Implement it.

* bus/bus.c (bus_context_new): Pass print_pid_fd to _dbus_become_daemon (bug #1720)

2004-10-29 Colin Walters <walters@redhat.com>

Patch from Ed Catmur <ed@catmur.co.uk>

* mono/doc/Makefile.am (install-data-local): Handle DESTDIR.

2004-10-29 Colin Walters <walters@redhat.com>

* bus/.cvsignore, qt/.cvsignore: Update.

2004-10-29 Colin Walters <walters@redhat.com>

Patch from Kristof Vansant <de_lupus@pandora.be>

* configure.in: Detect Slackware.

* bus/Makefile.am (SCRIPT_IN_FILES): Add rc.messagebus.in.

* bus/rc.messagebus.in: New file.

2004-10-29 Colin Walters <walters@redhat.com>

* tools/dbus-monitor.c (filter_func): Return DBUS_HANDLER_RESULT_HANDLED in filter function for now. See:

<http://freedesktop.org/pipermail/dbus/2004-August/001433.html>

2004-10-29 Colin Walters <walters@redhat.com>

Patch from Matthew Rickard <mjricka@epoch.ncsc.mil>

* bus/services.c (bus_registry_acquire_service): Correctly retrieve service name from DBusString for printing.

2004-10-29 Colin Walters <walters@redhat.com>

* dbus/dbus-glib.h: Update documentation to not refer to internal APIs.

2004-10-27 Joe Shaw <joesshaw@novell.com>

* mono/Arguments.cs (GetDBusTypeConstructor):

type.UnderlyingSystemType will return "System.Byte" if you do it on "byte[]", which is not what we want. So check the type.IsArray property and use System.Array instead.

2004-10-25 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps.c (fill_user_info): On errors do not free the DBusUserInfo structure since this is passed into the function.

This would cause a double free when the function that allocated the structure would try to free it when an error occurred.

* (bus/session.conf.in, bus/Makefile.am, dbus/configure.in): use /usr/share/dbus-1/services instead of /usr/lib/dbus-1.0/services for service activation to avoid 32bit/64bit parallel install issues

2004-10-21 Colin Walters <walters@verbum.org>

* AUTHORS: Fix my email address, the @gnu.org one has been bouncing for some time. Also add J5.

2004-10-21 Colin Walters <walters@verbum.org>

* dbus/dbus-transport-unix.c (do_authentication): Return authentication status to callers.
(unix_handle_watch): If we completed authentication this round, don't do another read. Instead wait until the next iteration, after we've read any pending data in the auth buffer.
(unix_do_iteration): Ditto.
(unix_handle_watch): Updated for new do_authentication prototype.

2004-10-18 Colin Walters <walters@verbum.org>

* bus/selinux.c (bus_selinux_enabled): Handle --disable-selinux case.

2004-10-18 Colin Walters <walters@verbum.org>

* bus/selinux.h: Add bus_selinux_enabled.

* bus/selinux.c (bus_selinux_enabled): Implement it.

* bus/config-parser.c (struct include): Add if_selinux_enabled member.
(start_busconfig_child): Parse if_selinux_enabled attribute for include.
(bus_config_parser_content): Handle it.

* bus/session.conf.in, bus/system.conf.in: Add

inclusion of context mapping to default config files;
conditional on SELinux being enabled.

* doc/busconfig.dtd: Add to if_selinux_enabled to default DTD.

* test/data/invalid-config-files/badselinux-1.conf,
test/data/invalid-config-files/badselinux-2.conf:
Test files for bad syntax.

2004-10-17 Colin Walters <walters@verbum.org>

* dbus/dbus-memory.c (_dbus_initialize_malloc_debug,
check_guards)
(dbus_malloc, dbus_malloc0, dbus_realloc): Fix up printf
format specifier mismatches.

2004-10-07 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-sysdeps.c (_dbus_file_get_contents): fix an incorrect
format string.

* glib/dbus-dbus-gmain.c (dbus_g_bus_get): do not mangle NULL
pointer (bug #1540, Leonardo Boiko).

2004-09-28 Jon Trowbridge <trow@ximian.com>

* mono/BusDriver.cs: Changed BusDriver struct to remove
the ServiceCreated and ServiceDeleted events and replace them
with the new ServiceOwnerChanged event.

* mono/example/BusListener.cs: Added a new example program,
which listens for and reports any ServiceOwnerChanged events
on the bus driver.

* mono/example/Makefile.am (DESTDIR): Build changes for the
new BusListener.cs example.

2004-09-27 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/signals.c (bus_match_rule_parse): validate the components
of
match rules (bug #1439).

* dbus/dbus-bus.c (dbus_bus_add_match): add a missing OOM test.

2004-09-24 Olivier Andrieu <oliv__a@users.sourceforge.net>

* doc/dbus-specification.xml: document ServiceOwnerChanged
signal.

* bus/driver.c, bus/driver.h, bus/services.c: Use
ServiceOwnerChanged signal instead of ServiceCreated and

ServiceDeleted.

* bus/dispatch.c: update testcase for the new signal.

2004-09-20 Jon Trowbridge <trow@ximian.com>

Patch from Nat Friedman <nat@novell.com>

* mono/Makefile.am: A number of small build fixes to allow "make distcheck" to succeed.

* mono/example/Makefile.am: "make distcheck" fixes.

* mono/AssemblyInfo.cs.in: When signing the assembly, look for the key in @srcdir@.

* test/Makefile.am: "make distcheck" fixes.

2004-09-17 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-sysdeps.c (_dbus_user_at_console): fix memleak in OOM.

* doc/busconfig.dtd: update the DTD for the at_console attribute.

* bus/driver.c (bus_driver_handle_hello): correctly handle Hello messages after the first one (bug #1389).

* bus/dispatch.c (check_double_hello_message): add a test case for the double hello message bug.
(check_existent_service_activation): fix check of spawning error.

2004-09-16 David Zeuthen <david@fubar.dk>

* python/dbus_bindings.pyx.in: Add support for int64 and uint64

2004-09-12 David Zeuthen <david@fubar.dk>

Patch from Kay Sievers <kay.sievers@vrfy.org>

* bus/bus.c (bus_context_new):

* bus/bus.h:

* bus/main.c (usage)

(main):

Add commandline option --nofork to override configuration file setting.

2004-09-09 Olivier Andrieu <oliv__a@users.sourceforge.net>

C * dbus/dbus-*.h: remove the ; after DBUS_(BEGIN|END)_DECLS. Some compilers don't like it (bug #974).

2004-09-04 Harald Fernengel <harry@kdevelop.org>

* qt/connection.*: Applied patch by JÃ©rÃ©me Lodewyck <lodewyck@clipper.ens.fr> to integrate an existing connection into the Qt eventloop

2004-08-30 Jon Trowbridge <trow@ximian.com>

* mono/BusDriver.cs: Added. This is a class for interacting with the org.freedesktop.DBus service.

is * mono/Message.cs: Added a mechanism to expose the message that currently being dispatched via the static Message.Current property. Added Message.Sender and Message.Destination properties.

* mono/Handler.cs: Expose the dispatched message via Message.Current when handling method calls.

* mono/Service.cs: Expose the dispatched message via Message.Current when handling signal emissions.

* mono/Connection.cs: Bind dbus_bus_get_base_service via the Connection.BaseService property.

2004-08-28 Havoc Pennington <hp@redhat.com>

* dbus/dbus-userdb.c (_dbus_is_console_user): remove unused variable

More fixes from Steve Grubb

* dbus/dbus-sysdeps.c (_dbus_connect_tcp_socket): fix fd leak (_dbus_listen_tcp_socket): fix fd leak

* dbus/dbus-spawn.c (read_pid, read_ints): move the "again:" for EINTR to a bit lower in the code

2004-08-26 Jon Trowbridge <trow@ximian.com>

* bus/driver.c (bus_driver_handle_service_exists): Respond with TRUE if we are inquiring about the existence of the built-in org.freedesktop.DBus service.

2004-08-25 John Palmieri <johnp@redhat.com>

* bus/config-parser.c: (struct PolicyType): Add POLICY_CONSOLE

(struct Element.d.policy): s/gid_or_uid/gid_uid_or_at_console
(start_busconfig_child): Sets up console element when
<policy at_console=""> is encountered in a policy file
(append_rule_from_element): Convert console elements to console
rules.

* bus/policy.c:

(bus_policy_create_client_policy): Add console rules to the
client

policy based on if the client is at the console

(bus_policy_append_console_rule): New function for adding a
console rule to a policy

(bus_policy_merge): Handle console rule merging

* dbus/dbus-sysdeps.h: Added the DBUS_CONSOLE_DIR constant
where we check for console user files

* dbus/dbus-sysdeps.c:

(_dbus_file_exists): New function which checks if the given
file exists

(_dbus_user_at_console): New function which does the system
specific process of checking if the user is at the console

* dbus/dbus-userdb.c:

(_dbus_is_console_user): New function converts a UID to user name
and then calls the system specific _dbus_user_at_console to
see if the user is at the console and therefor a console user

2004-08-25 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/config-parser.c (set_limit):

* bus/dbus-daemon-1.1.in:

* test/data/valid-config-files/many-rules.conf: set the
max_match_rules_per_connection limit from the config file.

* doc/busconfig.dtd: update the DTD.

* bus/driver.c: remove some unused variables.

2004-08-24 Mikael Hallendal <micke@imendio.com>

* dbus/dbus-glib-lowlevel.h: Removed dbus_bus_get_with_g_main
since

it's been replaced by dbus_g_bus_get

2004-08-23 Colin Walters <walters@redhat.com>

Updated SELinux support from Matthew Rickard
<mjricka@epoch.ncsc.mil>

* bus/selinux.h: Prototype bus_selinux_get_policy_root.

* bus/selinux.c: Create a thread for policy reload notification.
(bus_selinux_get_policy_root): Implement.

* bus/config-parser.c (start_busconfig_child)
(bus_config_parser_content): Support SELinux-root relative
inclusion.

* configure.in <HAVE_SELINUX>: Add -lpthread.

* bus/test-main.c (test_pre_hook, test_post_hook): New.
(test_post_hook): Move memory checking into here.
(test_pre_hook, test_post_hook): Move SELinux checks in
here, but conditional on a DBUS_TEST_SELINUX environment
variable. Unfortunately we can't run the SELinux checks
as a normal user, since they won't have any permissions
for /selinux. So this will have to be tested manually
for now, until we have virtualization for most of
libselinux.

2004-08-23 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_change_identity): add setgroups() to
drop supplementary groups, suggested by Steve Grubb

2004-08-20 Colin Walters <walters@redhat.com>

* bus/config-parser.c (start_busconfig_child): Remove some unused
variables.

* bus/selinux.c (bus_selinux_id_table_insert): Avoid compiler
warning.

2004-08-17 Joe Shaw <joeshaw@novell.com>

* configure.in: If --enable-mono is passed in, if we can't find
mono error out.

* mono/Makefile.am: Use /gacutil to install assemblies into the
GAC and not /root.

2004-08-12 Havoc Pennington <hp@redhat.com>

* NEWS: update for 0.22

* configure.in: release 0.22

2004-08-11 Colin Walters <walters@redhat.com>

* tools/dbus-send.c (main, usage): Add --reply-timeout
argument.

2004-08-10 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/bus.c (process_config_first_time_only): get rid of an unused
DBusError that was causing a memory leak (bug #989).

* dbus/dbus-keyring.c, dbus/dbus-message.c:
fix compilation on Solaris/Forte C (bug #974)

* bus/main.c (main): plug two minuscule memleaks.

2004-08-10 Havoc Pennington <hp@redhat.com>

* doc/dbus-tutorial.xml: add some more info on GLib bindings

2004-08-09 Havoc Pennington <hp@redhat.com>

* COPYING: switch to Academic Free License version 2.1 instead of
2.0, to resolve complaints about patent termination clause.

2004-07-31 John (J5) Palmieri <johnp@redhat.com>

* README: added documentation for the --enable-python
configure switch.

2004-07-31 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/config-parser.c (bus_config_parser_new): fix an invalid
_unref in the SELinux support.

* doc/busconfig.dtd: update DTD for SELinux support.

* bus/config-loader-libxml.c: fix error handler and parser
initialisation/cleanup. OOM test now works with libxml2 HEAD.

* configure.in: remove the warning about libxml2.

* dbus/dbus-bus.c: silence doxygen warning.

2004-07-31 Colin Walters <walters@redhat.com>

* configure.in: Move #error in SELinux check to its own line.

2004-07-31 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-internals.h (_DBUS_SET_OOM):
* bus/utls.h (BUS_SET_OOM): use dbus_error_set_const instead of
dbus_error_set.

* bus/dispatch.c (check_send_exit_to_service): fix the test case,
broken by the change in the _SET_OOM macros.

2004-07-31 Colin Walters <walters@redhat.com>

* bus/selinux.c <HAVE_SELINUX>: Include utils.h to get BUS_SET_OOM.

2004-07-31 Colin Walters <walters@redhat.com>

* configure.in: Use AC_TRY_COMPILE instead of AC_EGREP_HEADER to correctly detect DBUS__ACQUIRE_SVC. Also add an AC_MSG_CHECKING.

2004-07-24 Havoc Pennington <hp@redhat.com>

SELinux support from Matthew Rickard <mjricka@epoch.ncsc.mil>

* bus/selinux.c, bus/selinux.h: new file encapsulating selinux functionality

* configure.in: add --enable-selinux

* bus/policy.c (bus_policy_merge): add FIXME to a comment

* bus/main.c (main): initialize and shut down selinux

* bus/connection.c: store SELinux ID on each connection, to avoid repeated getting of the string context and converting it into an ID

* bus/bus.c (bus_context_get_policy): new accessor, though it isn't used
(bus_context_check_security_policy): check whether the security context of sender connection can send to the security context of recipient connection

* bus/config-parser.c: add parsing for <selinux> and <associate>

* dbus/dbus-transport.c (_dbus_transport_get_unix_fd): to implement dbus_connection_get_unix_fd()

* dbus/dbus-connection.c (dbus_connection_get_unix_fd): new function, used by the selinux stuff

2004-07-29 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/config-loader-libxml.c: complete the implementation of libxml backend for config file loader. Doesn't work with full OOM test yet.

* configure.in: change error when selecting libxml into a warning.

* test/data/invalid-config-files: add two non-well-formed XML files.

- * glib/Makefile.am: libdbus_gtool always uses expat, not libxml.
- * dbus/dbus-transport-unix.c (unix_handle_watch): do not disconnect in case of DBUS_WATCH_HANGUP, several do_reading() may be necessary to read all the buffer. (bug #894)
- * bus/activation.c (bus_activation_activate_service): fix a potential assertion failure (bug #896). Small optimization in the case of auto-activation messages.
- * dbus/dbus-message.c (verify_test_message, _dbus_message_test): add test case for byte-through-vararg bug (#901). patch by Kimmo Härmäläinen.

2004-07-28 Anders Carlsson <andersca@gnome.org>

- * python/dbus.py:
- * python/dbus_bindings.pyx.in:
Add dbus.init_gthreads (), allow emit_signal to pass arguments to the signal.

2004-07-24 Havoc Pennington <hp@redhat.com>

- * AUTHORS: add some people, not really comprehensively, let me know if I missed you

2004-07-24 Havoc Pennington <hp@redhat.com>

- * Makefile.am (DIST_SUBDIRS): add DIST_SUBDIRS, problem solved by Owen
- * test/Makefile.am (DIST_SUBDIRS): here also

2004-07-22 Olivier Andrieu <oliv__a@users.sourceforge.net>

- * dbus/dbus-sysdeps.c (fill_user_info): fix inexistent label name,
breaking build on Solaris, reported by Farhad Saberi on the ML.
- * dbus/dbus-message.c (dbus_message_append_args_valist): fix the va_arg invocation to account for integer promotion in the case of DBUS_TYPE_BYTE (unsigned char is promoted to int). (bug #901)
- * bus/services.c (bus_service_remove_owner): fix bug #902, use _dbus_list_get_first_link, not _dbus_list_get_first.
- * dbus/dbus-bus.c (dbus_bus_service_exists): plug a memory leak.
- * dbus/dbus-object-tree.c (free_subtree_recurse): always null handler functions so that the asserts in _dbus_object_subtree_unref

do not fail.

* dbus/dbus-transport-unix.c (do_reading):
_dbus_transport_queue_messages return value is of type
dbus_bool_t, not DBusDispatchStatus.

2004-07-19 David Zeuthen <david@fubar.dk>

* dbus/dbus-protocol.h: Add DBUS_ERROR_UNIX_PROCESS_ID_UNKNOWN

* bus/dispatch.c:
(check_get_connection_unix_user): Debug says GetProperty; but the
method is called GetConnectionUnixUser
(check_get_connection_unix_process_id): New function
(bus_dispatch_test): Actually call
check_get_connection_unix_user();
also call check_get_connection_unix_process_id()

* bus/driver.c:
(bus_driver_handle_get_connection_unix_process_id): New function,
handles GetConnectionUnixProcessID on the org.freedesktop.DBus
interface

* dbus/dbus-auth.c:
(handle_server_data_external_mech): Set pid from the credentials
obtained from the socket

* dbus/dbus-connection.c:
(dbus_connection_get_unix_process_id): New function

* dbus/dbus-connection.h:
Add prototype for dbus_connection_get_unix_process_id

* dbus/dbus-transport.c:
(_dbus_transport_get_unix_process_id): New function

* dbus/dbus-transport.h:
Add prototype for _dbus_transport_get_unix_process_id

2004-07-19 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-message.c: Message counter fix, patch by Christian
Hammond <chipx86@gnupdate.org>

2004-07-18 Seth Nickell <seth@gnome.org>

* python/dbus.py:
* python/dbus_bindings.pyx.in:
* python/tests/test-client.py:

Add dbus.ByteArray and dbus_bindings.ByteArray
types so that byte streams can be passed back.

Give jdahlin the heaps of credit that are so rightfully his.

2004-07-12 Seth Nickell <seth@gnome.org>

* python/dbus.py:

Add message argument to the default object_method_handler function.

* python/dbus_bindings.pyx.in:

Automatically return NIL when passed an empty list (we can't pass back a list since lists are typed and we don't have any idea what type the the client intended the list to be... :- ()

2004-07-10 Seth Nickell <seth@gnome.org>

* python/examples/Makefile.am:

Fix distcheck breakage caused by new examples.

2004-07-10 Seth Nickell <seth@gnome.org>

* python/dbus.py:

Add "message" argument to service-side dbus.Object methods. This will break existing services written using the python bindings, but will allow extraction of all the message information (e.g. who its from).

Add improved "object oriented" signal handling/emission.

* python/examples/example-service.py:

Nix this example.

* python/examples/example-signal-emitter.py:

* python/examples/example-signal-recipient.py:

Two new examples that show how to emit and receive signals using the new APIs.

* python/examples/example-signals.py:

* python/examples/gconf-proxy-service.py:

* python/examples/gconf-proxy-service2.py:

Add "message" argument to service methods.

2004-06-28 Kay Sievers <kay.sievers@vrfy.org>

- * bus/driver.c (bus_driver_handle_get_connection_unix_user)
- * dbus/bus.c (dbus_bus_get_unix_user)
- * doc/dbus-specification.xml: implement GetConnectionUnixUser method of org.freedesktop.DBus interface.

- * bus/dispatch.c: test case

2004-06-23 John (J5) Palmieri <johnp@redhat.com>

- * python/Makefile.am: switched include directory from glib/ to dbus/
since dbus-glib.h moved

2004-06-22 Olivier Andrieu <oliv__a@users.sourceforge.net>

- * configure.in: prevent building the gcj stuff and libxml loader since they are broken.

2004-06-20 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-glib-error-enum.h: autogenerate the GError enum codes from the dbus error names

- * glib/dbus-glib.h: move to subdir dbus/ since it's included as dbus/dbus-glib.h and that breakage is now visible due to including dbus/dbus-glib.h in dbus-glib-lowlevel.h

- * glib/dbus-glib.h: s/gproxy/g_proxy/

- * dbus/dbus-shared.h: new header to hold stuff shared with binding APIs

- * dbus/dbus-protocol.h (DBUS_ERROR_*): move errors here rather than dbus-errors.h

- * glib/dbus-glib.h (dbus_set_g_error): move to dbus-glib-lowlevel.h

- * glib/dbus-glib.h: remove dbus/dbus.h from here; change a bunch of stuff to enable this

- * dbus/dbus-glib-lowlevel.h: put dbus/dbus.h here

- * a bunch of other changes with the same basic "separate glib bindings from dbus.h" theme

2004-06-10 Owen Fraser-Green <owen@discobabe.net>

- * dbus-sharp.pc.in: Removed glib-sharp inclusion in Libs.

- * python/examples/Makefile.am: Fixed typo in EXTRA_DIST.

2004-06-09 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/driver.c, dbus/dbus-bus.c: use BOOLEAN instead of UINT32 for the reply value of the ServiceExists message.

2004-06-07 John (J5) Palmieri <johnp@redhat.com>

* python/dbus_bindings.pyx.in: No longer need to parse path elements and pass them as arrays of strings. The C API now accepts plain path strings.
(_build_parsed_path): removed

2004-06-07 Havoc Pennington <hp@redhat.com>

* doc/TODO: remove auto-activation item since it's done; sort items by importance/milestone

2004-06-07 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message-builder.c (_dbus_message_data_load): append random signature when using REQUIRED_FIELDS (this hack won't work in the long term)

* dbus/dbus-message.c: change the signature to be a header field, instead of message->signature special-case string. Incremental step forward. Then we can fix up code to send the signature in the message, then fix up code to validate said signature, then fix up code to not put the typecodes inline, etc.
(load_one_message): don't make up the signature after the fact
(decode_header_data): require signature field for the known message types

* dbus/dbus-marshal.c (_dbus_marshall_string_len): new

* dbus/dbus-protocol.h: add DBUS_HEADER_FIELD_SIGNATURE

2004-06-07 Owen Fraser-Green <owen@discobabe.net>

* mono/DBusType/ObjectPath.cs: Renamed PathName argument to Path

* mono/Handler.cs: Updated to follow new path argument for (un-)registering objects.

* mono/example/Makefile.am:

* mono/Makefile.am:

* configure.in: Bumped required version for mono and use new -pkg syntax for deps

2004-06-05 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-connection.h, dbus/dbus-connection.c: have object path registration functions take the path argument as char* instead of char**.

* dbus/dbus-marshal.h, dbus/dbus-marshal.c
(_dbus_decompose_path):
split off the path decomposition part of
_dbus_demarshal_object_path. Some misc. fixes to silence compiler warnings.

* glib/dbus-gobject.c, test/test-service.c: update accordingly.

2004-06-02 Kristian HÃ¸gsberg <krh@redhat.com>

* dbus/dbus-auth.c: Rewrite auth protocol handling to use a state machine approach. A state is implemented as a function that handles incoming events as specified for that state.

* doc/dbus-specification.xml: Update auth protocol state machine specification to match implementation. Remove some leftover base64 examples.

2004-06-02 Kristian HÃ¸gsberg <krh@redhat.com>

* glib/dbus-gproxy.c, glib/dbus-gmain.c, dbus/dbus-string.c, dbus/dbus-object-tree.c, dbus/dbus-message.c: add comments to quiet doxygen.

* Doxyfile.in: remove deprecated options.

* dbus/dbus-message-handler.c, dbus/dbus-message-handler.h, glib/test-thread.h, glib/test-thread-client.c, glib/test-thread-server.c, glib/test-profile.c, glib/test-dbus-glib.c: remove these unused files.

2004-06-01 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-object-tree.c
(_dbus_object_tree_dispatch_and_unlock): fix dispatch for non-fallback handlers (bug #684).
(_dbus_object_subtree_new): initialize invoke_as_fallback field.
(find_subtree_recurse): report whether the returned subtree is an exact match or a "fallback" match higher up in the tree.
(object_tree_test_iteration): update test case.

2004-06-01 Seth Nickell <seth@gnome.org>

* python/dbus_bindings.pyx.in:
* python/tests/test-client.py:

Round off basic type support. Add dicts (yay!), and remaining array types.

Make MessageIter more general so it works for dicts too.

Mark all loop variables as C integers.

2004-05-31 Havoc Pennington <hp@redhat.com>

* glib/dbus-gidl.c (method_info_add_arg): keep args sorted with "in" before "out"

* glib/dbus-gobject.c (dbus_type_to_string): move to dbus-gutils.c

* glib/dbus-glib-tool.c (main): set up to have a --self-test option that runs the tests, and start filling in some code including for starters just dumping the interfaces to stdout

* glib/Makefile.am (INCLUDES): define DBUS_LOCALEDIR

* test/data/valid-introspection-files/lots-of-types.xml: test of an example introspection file

* glib/dbus-gparser.c (parser_check_doctype): doctype should be "node" (I think...)

2004-05-31 Seth Nickell <seth@gnome.org>

* python/dbus_bindings.pyx.in:
* python/tests/test-client.py:

Test Suite: 1
Python Bindings: 0

Fix string array memory trashing bug... oops...

2004-05-30 Seth Nickell <seth@gnome.org>

* python/dbus.py:

Add a nicer-but-less-flexible alternate API for handling calls to virtual objects in dbus.ObjectTree.

Screw up the argument order to the dbus.Object constructor for consistency with dbus.ObjectTree (and to make dbus_methods optional for future extension)

* python/examples/Makefile.am:
* python/examples/gconf-proxy-service.py:
* python/examples/gconf-proxy-service2.py:

Alternate implementation of gconf-proxy-service using the nicer dbus.ObjectTree API.

- * python/examples/example-service.py:
- * python/tests/test-server.py

Reverse the argument order to deal with dbus.Object constructor changes.

2004-05-30 Seth Nickell <seth@gnome.org>

- * python/examples/example-client.py:
- * python/examples/example-service.py:

Take it back. Lists seem to work but they're broken in the test suite. Make the base examples use lists (works fine).

2004-05-30 Seth Nickell <seth@gnome.org>

- * python/dbus_bindings.pyx.in:
- * python/tests/test-client.py:

Add some more tests and fix errors that crop up. Unfortunately, currently it seems like marshalling and unmarshalling of lists is completely broken :-)

2004-05-30 Seth Nickell <seth@gnome.org>

- * python/dbus_bindings.pyx.in:

Add support for ObjectPath type.

- * python/dbus.py:

Refactor message handling code to a common function.

- * python/tests/test-client.py:
- * python/tests/test-server.py:

Add tests that check to make sure values of all types can be echoed from a service w/o mangling.

2004-05-29 Seth Nickell <seth@gnome.org>

- * python/dbus.py:

Add ObjectTree class which allows implementation of trees of "virtual" objects. Basically the python wrapper for "register_fallback".

- * python/examples/Makefile.am

- * python/examples/gconf-proxy-client.py:
- * python/examples/gconf-proxy-service.py:

Implement a simple GConf proxy service that supports get/set on string and int GConf keys using the ObjectTree.

2004-05-29 Seth Nickell <seth@gnome.org>

- * python/dbus.py:
- * python/examples/example-client.py:
- * python/examples/example-service.py:
- * python/examples/list-system-services.py:

Add SessionBus, SystemBus and ActivationBus classes so you don't need to know the special little BUS_TYPE flag.

2004-05-29 Havoc Pennington <hp@redhat.com>

* bus/config-parser.c (process_test_valid_subdir): temporarily stop testing config parser OOM handling, since expat has issues <http://freedesktop.org/pipermail/dbus/2004-May/001153.html>

* bus/dbus-daemon-1.1.in: change requested_reply to send_requested_reply/receive_requested_reply so we can send the replies, not just receive them.

* bus/config-parser.c: parse the new send_requested_reply/receive_requested_reply

* bus/policy.c (bus_client_policy_check_can_send): add requested_reply argument and use it

* bus/bus.c (bus_context_check_security_policy): pass through requested_reply status to message send check

* bus/system.conf.in: adapt to requested_reply change

2004-05-28 Havoc Pennington <hp@redhat.com>

* test/glib/test-service-glib.c (main): remove unused variable

* glib/dbus-gidl.c (base_info_ref): fix a silly compiler warning

* dbus/dbus-auth.h (enum): remove AUTHENTICATED_WITH_UNUSED_BYTES from the enum, no longer in use.

* dbus/dbus-sysdeps.h: include config.h so DBUS_VA_COPY actually works right.

* dbus/dbus-message.c: add various _dbus_return_val_if_fail for whether error_name passed in is a valid error name.

2004-05-28 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-message.c (dbus_message_get_args): Added support for OBJECT_PATH and OBJECT_PATH_ARRAY

2004-05-28 Seth Nickell <seth@gnome.org>

* python/examples/Makefile.am:

Forget to add Makefile.am. Do not pass go.

2004-05-28 Michael Meeks <michael@ximian.com>

* glib/dbus-gvalue.c (dbus_gvalue_marshal, dbus_gvalue_demarshal):
fix no int64 case.

* dbus/dbus-string.c (_dbus_string_parse_basic_type): impl.

* dbus/dbus-message.c (_dbus_message_iter_get_basic_type),
(_dbus_message_iter_get_basic_type_array): impl.
drastically simplify ~all relevant _get methods to use these.
(_dbus_message_iter_append_basic_array),
(dbus_message_iter_append_basic): impl
drastically simplify ~all relevant _append methods to use these.

* dbus/dbus-message-builder.c (parse_basic_type)
(parse_basic_array, lookup_basic_type): impl.
(_dbus_message_data_load): prune scads of duplicate /
cut & paste coding.

* dbus/dbus-marshal.c (_dbus_demarshal_basic_type_array)
(_dbus_demarshal_basic_type): implement,
(demarshal_and_validate_len/arg): beef up debug.
(_dbus_marshal_basic_type, _dbus_marshal_basic_type_array): impl.

2004-05-27 Seth Nickell <seth@gnome.org>

* configure.in:
* python/Makefile.am:

Include the example python apps in the tarball.

* python/examples/list-system-services.py

Add a python new example that fetches the list of services
from the system bus.

2004-05-27 Seth Nickell <seth@gnome.org>

* python/dbus.py:

* python/dbus_bindings.pyx.in:

Fix failure to notify that a signal was not handled,
resulted in hung functions.

2004-05-25 Colin Walters <walters@redhat.com>

* tools/dbus-monitor.c (main): Monitor all types of messages.

2004-05-23 Owen Fraser-Green <owen@discobabe.net>

* mono/Handler.cs, mono/Service.cs: Added UnregisterObject method
which unregisters the object path and disposes the handler.

2004-05-23 Kristian HÃgsberg <krh@redhat.com>

Patch from Timo TerÃs <ext-timo.teras@nokia.com> (#614):

* dbus/dbus-message.c (dbus_message_iter_get_args_valist): Swap
operands to && so we call dbus_message_iter_next () for the
last
argument also.

2004-05-21 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-object-tree.c
(_dbus_object_tree_list_registered_unlock, lookup_subtree):
return
children even if the requested path isn't registered.
(object_tree_test_iteration): test object_tree_list_registered.

* configure.in: undefine HAVE_ABSTRACT_SOCKETS instead of
defining
it to 0.

2004-05-20 Kristian HÃgsberg <krh@redhat.com>

* doc/TODO: Remove resolved items.

* bus/expirelist.h (struct BusExpireList): remove unused n_items
field.

* bus/connection.c (bus_connections_expect_reply): Enforce the
per-connection limit on pending replies.

Patch from Jon Trowbridge <trow@ximian.com>:

* bus/main.c (setup_reload_pipe): Added. Creates a pipe and sets
up a watch that triggers a config reload when one end of the pipe
becomes readable.

(signal_handler): Instead of doing the config reload in our
SIGHUP

handler, just write to the reload pipe and let the associated watch handle the reload when control returns to the main loop.

* bus/driver.c (bus_driver_handle_reload_config): Added. Implements a ReloadConfig method for requesting a configuration file reload via the bus driver.

2004-05-19 Owen Fraser-Green <owen@discobabe.net>

* HACKING: Updated release instructions concerning the wiki page.

2004-05-18 Kristian HÃ,gsberg <krh@redhat.com>

* dbus/dbus-auth.c (client_try_next_mechanism): Remove logic to filter against auth->allowed_mechs; we only add allowed mechs in record_mechanisms().

* dbus/dbus-auth-script.c (_dbus_auth_script_run): Add an ALLOWED_MECHS to auth-script format so we can set the list of allowed mechanisms.

* data/auth/client-out-of-mechanisms.auth-script: New test to check client disconnects when it is out of mechanisms to try.

* dbus/dbus-auth.c (process_command): Remove check for lines longer than 1 MB; we only buffer up maximum 16 kB.

* dbus/dbus-transport.c, dbus/dbus-transport-unix.c, dbus/dbus-auth-script.c, dbus/dbus-auth.c, dbus/dbus-auth.h: Remove auth state AUTHENTICATED_WITH_UNUSED_BYTES, instead always assume there might be unused bytes.

* dbus/dbus-auth.c (_dbus_auth_do_work): Remove check for client-out-of-mechs, it is handled in process_reject(). Move check for max failures to send_rejected(), as it's a server-only thing.

* dbus/dbus-auth.c: Factor out protocol reply code into functions send_auth(), send_data(), send_rejected(), send_error(), send_ok(), send_begin() and send_cancel().

2004-05-17 Kristian HÃ,gsberg <krh@redhat.com>

Remove base64 encoding, replace with hex encoding. Original patch from trow@ximian.com, added error handling.

* dbus/dbus-string.c (_dbus_string_base64_encode) (_dbus_string_base64_decode): Remove. (_dbus_string_hex_decode): Add end_return argument so we can distinguish between OOM and invalid hex encoding. (_dbus_string_test): Remove base64 tests and add test case for invalid hex.

* dbus/dbus-keyring.c, dbus/dbus-auth-script.c, dbus/dbus-auth.c:
Replace base64 with hex.

* test/data/auth/invalid-hex-encoding.auth-script: New test case
for invalid hex encoded data in auth protocol.

2004-05-17 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-connection.c (check_for_reply_unlocked): plug a
memory
leak.

2004-05-15 Owen Fraser-Green <owen@discobabe.net>

* mono/dbus-sharp.dll.config.in: Added for GAC
* mono/dbus-sharp.snk: Added for GAC
* mono/Assembly.cs.in: Added for GAC
* mono/Makefile.am: Changes for GAC installation
* configure.in: Added refs for dbus-sharp.dll.config.in and
Assembly.cs.in. More fixes for mono testing
* mono/example/Makefile.am: Changed var to CSC
* Makefile.am: Changed flag name to DBUS_USE_CSC

2004-05-15 Owen Fraser-Green <owen@discobabe.net>

* mono/Makefile.am: Added SUBDIRS for docs. Changed SUBDIRS order
* mono/doc/*: Added documentation framework
* configure.in: Added monodoc check
* README: Added description of mono configure flags

2004-05-11 John (J5) Palmieri <johnp@redhat.com>:

* doc/dbus-specification.xml: Added a "Required" column to the
header fields table and changed the "zero or more" verbage in
the above paragraph to read "The header must contain the required
named header fields and zero or more of the optional named header
fields".
* test/data/invalid-messages/*.message: Added the required PATH
named header field to the tests so that they don't fail on
'Missing path field'

2004-05-07 John (J5) Palmieri <johnp@redhat.com>

* python/dbus-bindings.pyx.in: Stopped the bindings from trashing
the stack by implicitly defining variable and parameter types and
removing the hack of defining C pointers as python objects and
later
casting them.

2004-05-02 Owen Fraser-Green <owen@discobabe.net>

* mono/Makefile.am: Removed test-dbus-sharp.exe from all target

2004-05-01 Owen Fraser-Green <owen@discobabe.net>

- * mono/DBusType/Dict.cs: Handle empty dicts
- * mono/DBusType/Array.cs: Handle empty arrays
- * mono/Arguments.cs: Handle empty arguments

2004-04-30 Owen Fraser-Green <owen@discobabe.net>

- * dbus-sharp.pc.in: Modified to include include Libs and Requires field

2004-04-25 Kristian HÃ,gsberg <krh@redhat.com>

- * test/data/valid-messages/standard-*.message: Update message test scripts to new header field names.

2004-04-22 John (J5) Palmieri <johnp@redhat.com>

- * test/break-loader.c (randomly_do_n_things): tracked down buffer overflow to times_we_did_each_thing array which would chop off the first character of the failure_dir string. Increased the size of the array to 7 to reflect the number of random mutation functions we have.

2004-04-21 Kristian HÃ,gsberg <krh@redhat.com>

- * dbus/dbus-server-unix.c (unix_finalize): Don't unref unix_server->watch here, it is unreffed in disconnect. (_dbus_server_new_for_tcp_socket): convert NULL host to "localhost" here so we don't append NULL to address.
- * dbus/dbus-server.c (_dbus_server_test): Add test case for various addresses, including tcp with no explicit host.

2004-04-21 Olivier Andrieu <oliv__a@users.sourceforge.net>

- * dbus/dbus-message.c (decode_header_data, decode_string_field): fix incorrect setting of .name_offset in the HeaderField (it was off by two bytes, positioned right after the name and typecode)
- * bus/bus.c (bus_context_new, bus_context_unref): test before calling dbus_server_free_data_slot and _dbus_user_database_unref in case of an error.
- * tools/Makefile.am: add \$(DBUS_GLIB_TOOL_LIBS), xml libs needed by libdbus-gtool.

2004-04-19 Kristian HÃ,gsberg <krh@redhat.com>

* dbus/dbus-transport-unix.c (unix_do_iteration): Rewrite to use _dbus_poll() instead of select().

2004-04-15 Jon Trowbridge <trow@ximian.com>

* bus/main.c (signal_handler): Reload the configuration files on SIGHUP.
(main): Set up our SIGHUP handler.

* bus/bus.c (struct BusContext): Store the config file, user and fork flag in the BusContext.
(process_config_first_time_only): Added. Contains the code (previously in bus_context_new) for setting up the BusContext

from

the BusConfigParser that should only be run the first time the config files are read.

(process_config_every_time): Added. Contains the code (previously in bus_context_new) for setting up the BusContext from the BusConfigParser that should be run every time the config files are read.

(load_config): Added. Builds a BusConfigParser from the config files and passes the resulting structure off to process_config_first_time_only (assuming this is the first time) and process_config_every_time.

(bus_context_new): All of the config-related code has been moved to process_config_first_time_only and process_config_every_time. Now this function just does the non-config-related initializations

and calls load_config.

(bus_context_reload_config): Added.

2004-04-15 Olivier Andrieu <oliv__a@users.sourceforge.net>

* bus/driver.c (bus_driver_handle_get_service_owner): implement a GetServiceOwner method.

* doc/dbus-specification.xml: document it.

* dbus/dbus-errors.h: add a 'ServiceHasNoOwner' error.

* glib/dbus-gproxy.c (dbus_gproxy_new_for_service_owner): implement, using the bus GetServiceOwner method.

* test/glib/test-dbus-glib.c:

use dbus_gproxy_new_for_service_owner so that we can receive the signal.

2004-04-15 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-internals.c, dbus/dbus-message-builder.c, dbus/dbus-message.c, dbus/dbus-protocol.h

(DBUS_HEADER_FIELD_SERVICE): renamed
DBUS_HEADER_FIELD_DESTINATION

* dbus/dbus-internals.c, dbus/dbus-message-builder.c,
dbus/dbus-message.c, dbus/dbus-protocol.h
(DBUS_HEADER_FIELD_SENDER_SERVICE): renamed
DBUS_HEADER_FIELD_SENDER

* dbus/dbus-internals.c (_dbus_header_field_to_string):
DBUS_HEADER_FIELD_DESTINATION resolves to "destination"
DBUS_HEADER_FIELD_SENDER resolves to "sender"

* doc/dbus-specification.xml (Header Fields Table):
s/SERVICE/DESTINATION
s/SENDER_SERVICE/SENDER

2004-04-14 Olivier Andrieu <oliv__a@users.sourceforge.net>

* test/glib/test-dbus-glib.c (timed_exit): fail the test after
a few seconds.

2004-04-13 Michael Meeks <michael@ximian.com>

* glib/dbus-gobject.c (handle_introspect): split out
(introspect_properties): this.
(handle_introspect): implement this.

* test/glib/Makefile.am: use the absolute path so the bus
daemon's chdir ("/") doesn't kill us dead.

* configure.in: subst ABSOLUTE_TOP_BUILDDIR

2004-04-12 Jon Trowbridge <trow@ximian.com>

* bus/config-parser.c (struct BusConfigParser): Added
included_files field.
(seen_include): Added. Checks whether or not a file has already
been included by any parent BusConfigParser.
(bus_config_parser_new): Copy the parent's included_files.
(include_file): Track which files have been included, and fail on
circular inclusions.
(process_test_valid_subdir): Changed printf to report if we are
testing valid or invalid conf files.
(all_are_equiv): Changed printf to be a bit clearer about
what we are actually doing.
(bus_config_parser_test): Test invalid configuration files.

2004-04-09 Jon Trowbridge <trow@ximian.com>

* bus/config-parser.c (bus_config_parser_new): Added a 'parent'

argument. If non-null, the newly-constructed BusConfigParser will be initialized with the parent's BusLimits instead of the default values.

(include_file): When including a config file, pass in the current parser as the parent and then copy the BusLimits from the included BusConfigParser pack to the current parser.
(process_test_valid_subdir): Renamed from process_test_subdir.
(process_test_equiv_subdir): Added. Walks through a directory, descending into each subdirectory and loading the config files it finds there. If any subdirectory contains two config files that don't produce identical BusConfigParser structs, fail. For now, the BusConfigParser's BusPolicies are not compared.
(bus_config_parser_test): Call both process_test_valid_subdir and process_test_equiv_subdir.

* bus/config-loader-libxml.c (bus_config_load): Take a parent argument and pass it along to the call to bus_config_parser_new. Also made a few small changes to allow this code to compile.

* bus/config-loader-expat.c (bus_config_load): Take a parent argument and pass it along to the call to bus_config_parser_new.

* bus/bus.c (bus_context_new): Load the config file with a NULL parent argument.

2004-03-29 Michael Meeks <michael@ximian.com>

* glib/dbus-gobject.c (introspect_properties): split out, fix mangled 'while' flow control.
(introspect_signals): implement.
(handle_introspect): update.

2004-03-29 Michael Meeks <michael@ximian.com>

* glib/dbus-gobject.c (set_object_property): split out / re-work, use the property type, and not the message type(!)
(get_object_property): ditto.

* glib/dbus-gvalue.c (dbus_gvalue_demarshal),
(dbus_gvalue_marshal): make this code re-usable, needed for signals too, also on both proxy and server side. Re-write for more efficiency / readability.

2004-03-29 Michael Meeks <michael@ximian.com>

* dbus/dbus-message.c
(dbus_message_new_error_printf): impl.

* dbus/dbus-connection.c
(dbus_connection_unregister_object_path): fix warning.

* configure.in: fix no-mono-installed situation.

2004-03-27 Havoc Pennington <hp@redhat.com>

Patch from Timo Teräs:

* tools/dbus-send.c (main): if --print-reply, assume type is method call; support boolean type args

* dbus/dbus-connection.c (dbus_connection_send_with_reply): fix a bunch of memleak and logic bugs

2004-03-23 Owen Fraser-Green <owen@discobabe.net>

* mono/Arguments.cs:
* mono/Introspector.cs:
* mono/Handler.cs:
* mono/InterfaceProxy.cs:
* mono/Message.cs
* mono/ProxyBuilder.cs:
* mono/Service.cs:

Added InterfaceProxy class to avoid building proxies for every object.

* dbus-message.h:
* dbus-message.c (dbus_message_append_args_valist)
(dbus_message_iter_get_object_path)
(dbus_message_iter_get_object_path_array)
(dbus_message_iter_append_object_path)
(dbus_message_iter_append_object_path_array):
Added object_path iter functions to handle OBJECT_PATH arguments

2004-03-23 Owen Fraser-Green <owen@discobabe.net>

First checkin of mono bindings.

* configure.in:

* Makefile.am:

Build stuff for the bindings

* dbus-sharp.pc.in: Added for pkgconfig

2004-03-21 Havoc Pennington <hp@redhat.com>

* test/test-service.c (main): remove debug spew

2004-03-21 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-marshal.c (_dbus_marshal_validate_arg): accept empty arrays

* dbus/dbus-message.h, bus/dbus-message.c
(dbus_message_iter_init)
(dbus_message_iter_init_array_iterator)

(dbus_message_iter_init_dict_iterator): return a dbus_bool_t to indicate whether the iterator is empty

* dbus/dbus-pending-call.c, dbus/dbus-server.c: silence compiler warnings

2004-03-19 Havoc Pennington <hp@redhat.com>

* NEWS: 0.21 updates

* configure.in: 0.21

* doc/Makefile.am: add all XMLTO usage to DBUS_XML_DOCS_ENABLED

* python/Makefile.am: change to avoid dist of dbus_bindings.c so you don't need pyrex to make dist

* qt/Makefile.am (libdbus_qt_1_la_SOURCES): add integrator.h to sources; run moc

2004-03-18 Richard Hult <richard@imendio.com>

* dbus/dbus-message.c (dbus_message_get_auto_activation), (dbus_message_set_auto_activation): Add doxygen docs.

2004-03-16 Richard Hult <richard@imendio.com>

* bus/activation.c: (bus_activation_service_created), (bus_activation_send_pending_auto_activation_messages), (bus_activation_activate_service):
* bus/activation.h:
* bus/dispatch.c: (bus_dispatch), (check_nonexistent_service_auto_activation), (check_service_auto_activated), (check_segfault_service_auto_activation), (check_existent_service_auto_activation), (bus_dispatch_test):
* bus/driver.c: (bus_driver_handle_activate_service):
* bus/services.c: (bus_registry_acquire_service):
* dbus/dbus-message.c: (dbus_message_set_auto_activation), (dbus_message_get_auto_activation):
* dbus/dbus-message.h:
* dbus/dbus-protocol.h: Implement auto-activation.

* doc/dbus-specification.xml: Add auto-activation to the spec.

2004-03-12 Olivier Andrieu <oliv__a@users.sourceforge.net>

* dbus/dbus-marshal.c (_dbus_marshal_get_arg_end_pos): fix a bug with CUSTOM types.

* dbus/dbus-message.c (message_iter_test, _dbus_message_test):

add

a unit test for this bug (used to fail).

2004-03-12 Mikael Hallendal <micke@imendio.com>

```
* bus/activation.c:
(babysitter_watch_callback): notify all pending activations
waiting for
    the same exec that the activation failed.
(bus_activation_activate_service): shortcut the activation if we
    already waiting for the same executable to start up.
```

2004-03-12 Mikael Hallendal <micke@imendio.com>

```
* bus/activation.c:
- Added service file reloading.
  Each service files directory is kept in an hash table in
  BusActivation and each BusActivationEntry knows what .service-
file it
  was read from. So when you try to activate a service the bus
will
  check if it's been updated, removed or if new .service-files
has
  been installed.
- Test code at the bottom for the service file reloading.
* bus/test-main.c: (main):
* bus/test.h:
- added service reloading test.
* dbus/dbus-sysdeps.c:
* dbus/dbus-sysdeps.h: (_dbus_delete_directory): Added.
```

2004-03-08 Michael Meeks <michael@ximian.com>

```
* dbus/dbus-connection.c (_dbus_connection_block_for_reply):
bail immediately if disconnected, to avoid busy loop.

* dbus/dbus-message.c (dbus_message_iter_get_args_valist):
cleanup cut/paste/inefficiency.
```

2004-03-01 David Zeuthen <david@fubar.dk>

```
* dbus/dbus-string.c (_dbus_string_append_printf_valist): Fix a
bug where args were used twice. This bug resulted in a segfault
on a Debian/PPC system when starting the messagebus daemon.
Include
  dbus-sysdeps.h for DBUS_VA_COPY

* dbus/dbus-sysdeps.h: Define DBUS_VA_COPY if neccessary. From
GLib

* configure.in: Check for va_copy; define DBUS_VA_COPY to the
appropriate va_copy implementation. From GLib
```

2004-02-24 Joe Shaw <joe@ximian.com>

* bus/services.c (bus_registry_acquire_service): We need to pass in the service name to dbus_set_error() to prevent a crash.

2003-12-26 Anders Carlsson <andersca@gnome.org>

* AUTHORS: Reveal my True identity.

2003-12-17 Mikael Hallendal <micke@imendio.com>

* dbus/dbus-message.c: (dbus_message_append_args_valist):
- Added case for DBUS_TYPE_BYTE, patch from Johan Hedberg.

2003-12-13 Mikael Hallendal <micke@imendio.com>

* doc/TODO: Added not about better error check of configuration files.

2003-12-02 Richard Hult <richard@imendio.com>

* Update AFL version to 2.0 throughout the source files to reflect the update that was done a while ago.

2003-12-02 Richard Hult <richard@imendio.com>

* dbus/dbus-message.c (dbus_message_iter_append_dict): Set wrote_dict_key to FALSE on the iter that the dict is appended to, just like when appending other types. Fixes a bug where a dict couldn't be put inside a dict.
(dbus_message_iter_append_dict_key): Fix typo in warning message.
(message_iter_test, _dbus_message_test): Add test case for dict inside dict.

2003-12-01 David Zeuthen <david@fubar.dk>

* python/dbus.py: Add the actual message when calling the reciever of a signal such that parameters can be inspected. Add the method remove_signal_receiver

2003-11-26 Mikael Hallendal <micke@imendio.com>

* bus/*.ch:
* dbus/*.ch:
* glib/*.ch: Made ref functions return the pointer

2003-11-25 Zack Rusin <zack@kde.org>

* qt/integrator.h, qt/integrator.cpp: Adding handling of DBusServer,

* qt/server.h, qt/server.cpp, qt/Makefile.am: Adding DBusServer wrappers,

* qt/connection.h, qt/connection.cpp: Adjusting to changes in the Integrator and to better fit with the server,

2003-11-24 Zack Rusin <zack@kde.org>

* qt/connection.h, qt/connection.cpp: removing initDbus method since the integrator handles it now

* qt/integrator.h, qt/integrator.cpp: reworking handling of timeouts, since QTimer wasn't really meant to be used the way DBusTimeout is

2003-11-24 Zack Rusin <zack@kde.org>

* qt/integrator.h, qt/integrator.cpp, Makefile.am: Adding Integrator class which integrates D-BUS with the Qt event loop,

* qt/connection.h, qt/connection.cpp: Move all the code which was dealing with D-BUS integration to the Integrator class, and start using Integrator,

2003-11-23 Zack Rusin <zack@kde.org>

* qt/connection.h, qt/connection.cpp: Adding the DBusConnection wrapper

* qt/message.h, qt/message.cpp: updating to the current D-BUS api, switching namespaces to DBusQt, reworking the class,

* Makefile.cvs: switching dependencies so that it matches KDE schematics,

* qt/Makefile.am: adding connection.{h,cpp} and message.{h,cpp} to the library

2003-11-19 Havoc Pennington <hp@redhat.com>

* NEWS: update

* configure.in: bump version to 0.20

* configure.in (have_qt): add yet another place to look for qt (someone hand trolltech a .pc file...)

2003-11-01 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: add state machine docs on the auth protocol; just a first draft, I'm sure it's wrong.

2003-10-28 David Zeuthen <david@fubar.dk>

* python/dbus_bindings.pyx.in: add get_dict to handle dictionaries
return types. Fixup TYPE_* to reflect changes in dbus/dbus-protocol.h

2003-10-28 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (get_next_field): delete unused function

2003-10-28 Havoc Pennington <hp@redhat.com>

* bus/expirelist.c (do_expiration_with_current_time): detect failure of the expire_func due to OOM

* bus/connection.c (bus_pending_reply_expired): return FALSE on OOM

* bus/dispatch.c (check_send_exit_to_service): fix to handle the NoReply error that's now created by the bus when the service exits

2003-10-28 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (_dbus_message_test): enable and fix the tests for set_path, set_interface, set_member, etc.

* dbus/dbus-string.c (_dbus_string_insert_bytes): allow 0 bytes

* dbus/dbus-message.c (set_string_field): always just delete and re-append the field; accept NULL for deletion
(re_align_fields_recurse): reimplement

2003-10-26 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c: fix docs to properly describe the disconnected message
(_dbus_connection_notify_disconnected): remove this function; we can't synchronously add the disconnected message, we have to do it after we've queued any remaining real messages
(_dbus_connection_get_dispatch_status_unlocked): queue the disconnect message only if the transport has finished queueing all
its real messages and is disconnected.
(dbus_connection_disconnect): update the dispatch status here

2003-10-22 Havoc Pennington <hp@redhat.com>

- * bus/bus.c (bus_context_check_security_policy): fix up assertion
- * bus/connection.c (bus_transaction_send_from_driver): set the destination to the connection's base service

2003-10-20 Havoc Pennington <hp@redhat.com>

hmm, make check is currently not passing.

- * doc/dbus-specification.xml: add requirement that custom type names follow the same rules as interface names.
- * dbus/dbus-protocol.h: change some of the byte codes, to avoid duplication and allow 'c' to be 'custom'; dict is now 'm' for 'map'
- * doc/dbus-specification.xml: update type codes to match dbus-protocol.h, using the ASCII byte values. Rename type NAMED to
CUSTOM. Add type OBJECT_PATH to the spec.

2003-10-17 Havoc Pennington <hp@redhat.com>

- * bus/driver.c (create_unique_client_name): use "." as separator in base service names instead of '-'
- * dbus/dbus-string.c (_dbus_string_get_byte): allow getting nul byte at the end of the string
- * dbus/dbus-internals.h (_DBUS_LIKELY, _DBUS_UNLIKELY): add optimization macros since string validation seems to be a slow point.
- * doc/dbus-specification.xml: restrict valid service/interface/member/error names. Add test suite code for the name validation.
- * dbus/dbus-string.c: limit service/interface/member/error names to [0-9][A-Z][a-z]_
- * dbus/dbus-connection.c (dbus_connection_dispatch): add missing format arg to verbose spew
- * glib/dbus-gproxy.c (dbus_gproxy_call_no_reply): if not out of memory, return instead of g_error
- * test/test-service.c (path_message_func): support emitting a signal on request
- * dbus/dbus-bus.c (init_connections_unlocked): only fill in

activation bus type if DBUS_BUS_ACTIVATION was set; default to assuming the activation bus was the session bus so that services started manually will still register.

(init_connections_unlocked): fix so that in OOM situation we get the same semantics when retrying the function

* test/test-service.c (main): change to use path registration, to test those codepaths; register with DBUS_BUS_ACTIVATION rather than DBUS_BUS_SESSION

2003-10-16 Havoc Pennington <hp@redhat.com>

* glib/dbus-gtest-main.c: bracket with #ifdef DBUS_BUILD_TESTS

* Makefile.am (GCOV_DIRS): remove "test", we don't care about test

coverage of the tests

(coverage-report.txt): don't move the .da and .bbg files around

2003-10-16 Havoc Pennington <hp@redhat.com>

* bus/bus.c (struct BusContext): remove struct field I didn't mean

to put there

2003-10-16 Havoc Pennington <hp@redhat.com>

* bus/connection.c (bus_pending_reply_expired): either cancel or execute, not both

(bus_connections_check_reply): use unlink, not remove_link, as we don't want to free the link; fixes double free mess

* dbus/dbus-pending-call.c (dbus_pending_call_block): fix in case where no reply was received

* dbus/dbus-connection.c

(_dbus_pending_call_complete_and_unlock):

fix a refcount leak

* bus/signals.c (match_rule_matches): add special cases for the bus driver, so you can match on sender/destination for it.

* dbus/dbus-sysdeps.c (_dbus_abort): print backtrace if DBUS_PRINT_BACKTRACE is set

* dbus/dbus-internals.c: add pid to assertion failure messages

* dbus/dbus-connection.c: add message type code to the debug spew

* glib/dbus-gproxy.c (gproxy_get_match_rule): match rules want sender=foo not service=foo

- * dbus/dbus-bus.c (dbus_bus_get): if the activation bus is the session bus but DBUS_SESSION_BUS_ADDRESS isn't set, use DBUS_ACTIVATION_ADDRESS instead
- * bus/activation.c: set DBUS_SESSION_BUS_ADDRESS, DBUS_SYSTEM_BUS_ADDRESS if appropriate
- * bus/bus.c (bus_context_new): handle OOM copying bus type into context struct
- * dbus/dbus-message.c (dbus_message_iter_get_object_path): new function
(dbus_message_iter_get_object_path_array): new function (half finished, disabled for the moment)
- * glib/dbus-gproxy.c (dbus_gproxy_end_call): properly handle DBUS_MESSAGE_TYPE_ERROR
- * tools/dbus-launch.c (babysit): support DBUS_DEBUG_OUTPUT to avoid redirecting stderr to /dev/null
(babysit): close stdin if not doing the "exit_with_session" thing
- * dbus/dbus-sysdeps.c (_dbus_become_daemon): delete some leftover debug code; change DBUS_DEBUG_OUTPUT to only enable stderr, not stdout/stdin, so things don't get confused
- * bus/system.conf.in: fix to allow replies, I modified .conf instead of .conf.in again.

2003-10-14 David Zeuthen <david@fubar.dk>

- * python/dbus_bindings.pyx.in (MessageIter.get): fixed typo in argtype to arg_type when raising unknown arg type exception. Changed type list to reflect the changes in dbus-protocol.h so the bindings actually work.

2003-10-14 Havoc Pennington <hp@redhat.com>

- * test/decode-gcov.c: support gcc 3.3 also, though gcc 3.3 seems to have a bug keeping it from outputting the .da files sometimes
(string_get_string): don't append garbage nul bytes to the string.

2003-10-15 Seth Nickell <seth@gnome.org>

- * python/Makefile.am:
Include dbus_h_wrapper.h in the dist tarball.

2003-10-14 Havoc Pennington <hp@redhat.com>

- * bus/bus.c (bus_context_check_security_policy): revamp this to

work more sanely with new policy-based requested reply setup

- * bus/connection.c (bus_transaction_send_from_driver): set bus driver messages as no reply

- * bus/policy.c (bus_client_policy_check_can_receive): handle a requested_reply attribute on allow/deny rules

- * bus/system.conf: add <allow requested_reply="true"/>

- * bus/driver.c (bus_driver_handle_message): fix check for replies sent to the bus driver, which was backward. How did this ever

work

at all though? I think I'm missing something.

- * dbus/dbus-message.c (decode_header_data): require error and method return messages to have a reply serial field to be valid
- (_dbus_message_loader_queue_messages): break up this function; validate that reply serial and plain serial are nonzero; clean up the OOM/error handling.

- (get_uint_field): don't return -1 from this

- (dbus_message_create_header): fix signed/unsigned bug

- * bus/connection.c (bus_connections_expect_reply): save serial of the incoming message, not reply serial

2003-10-14 Havoc Pennington <hp@redhat.com>

- * bus/connection.c: implement pending reply tracking using BusExpireList

- * bus/bus.c (bus_context_check_security_policy): verify that a reply is pending in order to allow a reply to be sent. Deny messages of unknown type.

- * bus/dbus-daemon-1.1.in: update to mention new resource limits

- * bus/bus.c (bus_context_get_max_replies_per_connection): new
- (bus_context_get_reply_timeout): new

2003-10-13 Seth Nickell <seth@gnome.org>

- * python/Makefile.am:

- Pass "make distcheck": remove a couple files from DIST_FILES that weren't included in the final version.

2003-10-12 Havoc Pennington <hp@pobox.com>

Added test code that 1) starts an actual bus daemon and 2) uses

DBusGProxy; fixed bugs that were revealed by the test. Lots

more testing possible, but this is the basic framework.

* glib/dbus-gproxy.c (dbus_gproxy_manager_unregister): remove empty proxy lists from the proxy list hash

* dbus/dbus-message.c (dbus_message_iter_get_args_valist): add a couple of return_if_fail checks

* dbus/dbus-pending-call.c (_dbus_pending_call_new): use dbus_new0 to allocate, so everything is cleared to NULL as it should be.

* glib/dbus-gmain.c (dbus_connection_setup_with_g_main): pass source as data to dbus_connection_set_timeout_functions() as the timeout functions expected

* test/glib/run-test.sh: add a little script to start up a message bus and run tests using it

* tools/dbus-launch.1: updates

* tools/dbus-launch.c (main): add --config-file option

* tools/dbus-launch.c (main): remove confusing else if (runprog) that could never be reached.

* dbus/dbus-message.c (dbus_message_new_method_return) (dbus_message_new_error, dbus_message_new_signal): set the no-reply-expected flag on all these. Redundant, but may as well be consistent.

2003-10-11 Havoc Pennington <hp@pobox.com>

* test/decode-gcov.c (function_solve_graph): make broken block graph a nonfatal error since it seems to be broken. Need to debug this.

* dbus/dbus-marshal.c (_dbus_type_is_valid): new function since we can't just check type > INVALID < LAST anymore

* dbus/dbus-message.c (dbus_message_get_signature): new function (dbus_message_has_signature): new function (struct DBusMessage): add signature field (right now it isn't sent over the wire, just generated on the fly) (dbus_message_copy): copy the signature, and init strings to proper length to avoid some reallocs (dbus_message_iter_init_array_iterator): return void, since it can't fail

(dbus_message_iter_init_dict_iterator): return void since it
can't fail
(_dbus_message_loader_queue_messages): add silly temporary hack
to
fill in message->signature on load

- * dbus/dbus-protocol.h: change DBUS_TYPE_* values to be ASCII characters, so they are relatively human-readable.

2003-10-11 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-message.c (_dbus_message_test): add more test coverage, but #if 0 for now since they uncover a bug not fixed yet; I think in re_align_field_recurse()
(re_align_field_recurse): add FIXME about broken assertion
- * dbus/dbus-sysdeps.c (_dbus_sysdeps_test): add more test coverage
- * bus/connection.c: share a couple code bits with expirelist.c
- * bus/expirelist.h, bus/expirelist.c: implement a generic expire-items-after-N-seconds facility, was going to share between expiring connections and replies, decided not to use for expiring connections for now.
- * COPYING: include AFL 2.0 (still need to change all the file headers)

2003-10-09 Havoc Pennington <hp@redhat.com>

- * configure.in: define DBUS_HAVE_GCC33_GCOV if we have gcc 3.3. Not that we do anything about it yet.
- * bus/signals.c (bus_match_rule_parse): impose max length on the match rule text
- * dbus/dbus-protocol.h: add DBUS_MAXIMUM_MATCH_RULE_LENGTH

2003-10-09 Havoc Pennington <hp@redhat.com>

Make matching rules theoretically work (add parser).

- * bus/bus.c (bus_context_check_security_policy): fix up to handle the case where destination is explicitly specified as bus driver and someone else is eavesdropping.
- * bus/policy.c (bus_client_policy_check_can_receive): fix up definition of eavesdropping and assertion
- * tools/dbus-send.c (main): use dbus_message_type_from_string

- * bus/signals.c (bus_match_rule_parse): implement
- * dbus/dbus-message.c (dbus_message_type_from_string): new
- * dbus/dbus-errors.h (DBUS_ERROR_MATCH_RULE_INVALID): add

2003-10-02 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gproxy.c (dbus_gproxy_call_no_reply): rename from dbus_gproxy_oneway_call
- * glib/dbus-gmain.c (dbus_connection_setup_with_g_main) (dbus_server_setup_with_g_main): fix to allow calling them more than once on the same args
- (dbus_bus_get_with_g_main): new function

2003-10-02 Havoc Pennington <hp@redhat.com>

- * doc/dbus-tutorial.xml: write some stuff

2003-09-29 Havoc Pennington <hp@pobox.com>

- * configure.in: split checks for Doxygen from XML docs, check for xmlto
- * doc/Makefile.am: XML-ify all the docs, and add a blank dbus-tutorial.xml

2003-09-29 Havoc Pennington <hp@pobox.com>

- * Merge dbus-object-names branch. To see the entire patch do `cvs diff -r DBUS_OBJECT_NAMES_BRANCHPOINT -r dbus-object-names`, it's huuuuge though.
- To revert, I tagged `DBUS_BEFORE_OBJECT_NAMES_MERGE`.

2003-09-28 Havoc Pennington <hp@pobox.com>

- * HACKING: update to reflect new server

2003-09-26 Seth Nickell <seth@gnome.org>

- * python/dbus.py:
- * python/examples/example-signals.py:

Start implementing some notions of signals. The API is really terrible, but they sort of work (with the exception of being able to filter by service, and to transmit signals *as* a particular service). Need to figure out how to make messages come from the service we registered :-(

* python/dbus_bindings.pyx.in:

Removed duplicate message_handler callbacks.

2003-09-25 Havoc Pennington <hp@redhat.com>

* bus/session.conf.in: fix my mess

2003-09-25 Havoc Pennington <hp@pobox.com>

* bus/session.conf.in: fix security policy, reported by Seth Nickell

2003-09-25 Seth Nickell <seth@gnome.org>

* python/examples/example-service.py:

Johan notices complete wrong code in example-service, but completely wrong in a way that works exactly the same (!). Johan is confused, how could this possibly work? Example code fails to serve purpose of making things clear. Seth fixes.

2003-09-25 Mark McLoughlin <mark@skynet.ie>

* doc/dbus-specification.sgml: don't require header fields to be 4-byte aligned and specify that fields should be distinguished from padding by the fact that zero is not a valid field name.

* doc/TODO: remove re-alignment item and add item to doc the OBJECT_PATH type.

* dbus/dbus-message.c:

(HeaderField): rename the original member to value_offset and introduce a name_offset member to keep track of where the field actually begins.
(adjust_field_offsets): remove.
(append_int_field), (append_uint_field),
(append_string_field): don't align the start of the header field to a 4-byte boundary.
(get_next_field): impl finding the next marshalled field after a given field.
(re_align_field_recurse): impl re-aligning a number of already marshalled fields.
(delete_field): impl deleting a field of any type and re-aligning any following fields.
(delete_int_or_uint_field), (delete_string_field): remove.
(set_int_field), (set_uint_field): no need to re-check that we have the correct type for the field.
(set_string_field): ditto and impl re-aligning any following fields.

(decode_header_data): update to take into account that the fields aren't 4-byte aligned any more and the new way to distinguish padding from header fields. Also, don't exit when there is too much header padding.
(process_test_subdir): print the directory.
(_dbus_message_test): add test to make sure a following field is re-aligned correctly after field deletion.

* dbus/dbus-string.[ch]:
(_dbus_string_insert_bytes): rename from insert_byte and allow the insert of multiple bytes.
(_dbus_string_test): test inserting multiple bytes.

* dbus/dbus-marshal.c: (_dbus_marshal_set_string): add warning note to docs about having to re-align any marshalled values following the string.

* dbus/dbus-message-builder.c:
(append_string_field), (_dbus_message_data_load): don't align the header field.

* dbus/dbus-auth.c: (process_test_subdir): print the directory.

* test/break-loader.c: (randomly_add_one_byte): upd. for insert_byte change.

* test/data/invalid-messages/bad-header-field-alignment.message: new test case.

* test/data/valid-messages/unknown-header-field.message: shove a dict in the unknown field.

2003-09-25 Seth Nickell <seth@gnome.org>

* python/dbus.py:
* python/dbus_bindings.pyx.in:

Handle return values.

* python/examples/example-client.py:
* python/examples/example-service.py:

Pass back return values from the service to the client.

2003-09-24 Seth Nickell <seth@gnome.org>

* python/dbus.py:

Connect Object methods (when you are sharing an object) up...

pass

in a list of methods to be shared. Sharing all the methods just

worked out too weird. You can now create nice Services over the DBus in Python. :-)

- * python/dbus_bindings.pyx.in:

Keep references to user_data tuples passed into C functions so Python doesn't garbage collect on us.

Implement MethodReturn and Error subclasses of Message for creating DBusMessage's of those types.

- * python/examples/example-client.py:
- * python/examples/example-service.py:

Simple example code showing both how create DBus services and objects, and how to use them.

2003-09-23 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gproxy.c (dbus_gproxy_manager_filter): implement

2003-09-23 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gproxy.c (dbus_gproxy_connect_signal): implement
- (dbus_gproxy_disconnect_signal): implement
- (dbus_gproxy_manager_remove_signal_match): implement
- (dbus_gproxy_manager_add_signal_match): implement
- (dbus_gproxy_oneway_call): implement

2003-09-23 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gproxy.c (struct DBusGProxy): convert to a GObject subclass. This means dropping the transparent thread safety of the proxy; you now need a separate proxy per-thread, or your own locking on the proxy. Probably right anyway.
- (dbus_gproxy_ref, dbus_gproxy_unref): nuke, just use g_object_ref

2003-09-22 Havoc Pennington <hp@redhat.com>

- * glib/dbus-gproxy.c (dbus_gproxy_manager_get): implement

2003-09-21 Seth Nickell <seth@gnome.org>

First checkin of the Python bindings.

- * python/.cvsignore:
- * python/Makefile.am:
- * python/dbus_bindings.pyx.in:
- * python/dbus_h_wrapper.h:

Pieces for Pyrex to operate on, building a `dbus_bindings.so` python module for low-level access to the Dbus APIs.

- * `python/dbus.py`:

High-level Python module for accessing Dbus objects.

- * `configure.in`:

- * `Makefile.am`:

Build stuff for the python bindings.

- * `acinclude.m4`:

Extra macro needed for finding the Python C header files.

2003-09-21 Havoc Pennington <hp@pobox.com>

- * `glib/dbus-gproxy.c` (`dbus_gproxy_manager_new`): start implementing the proxy manager, didn't get very far.

- * `dbus/dbus-bus.c` (`dbus_bus_add_match`): new
(`dbus_bus_remove_match`): new

- * `glib/dbus-gproxy.c` (`dbus_gproxy_new_for_service`): add a `path_name` argument; adjust the other not-yet-implemented gproxy constructors to be what I think they should be.

2003-09-21 Havoc Pennington <hp@pobox.com>

- * `dbus/dbus-bus.c` (`dbus_bus_get`): set `exit_on_disconnect` to TRUE by default for message bus connections.

- * `dbus/dbus-connection.c` (`dbus_connection_dispatch`): exit if `exit_on_disconnect` flag is set and we process the disconnected signal.
(`dbus_connection_set_exit_on_disconnect`): new function

2003-09-21 Havoc Pennington <hp@pobox.com>

Get matching rules mostly working in the bus; only actually parsing the rule text remains. However, the client side of "signal connections" hasn't been started, this patch is only the bus side.

- * `dbus/dispatch.c`: fix for the matching rules changes

- * `bus/driver.c` (`bus_driver_handle_remove_match`)
(`bus_driver_handle_add_match`): send an ack reply from these method calls

- * glib/dbus-gproxy.c (dbus_gproxy_begin_call): fix order of arguments, reported by Seth Nickell
- * bus/config-parser.c (append_rule_from_element): support eavesdrop=true|false attribute on policies so match rules can be prevented from snooping on the system bus.
- * bus/dbus-daemon-1.1.in: consistently use terminology "sender" and "destination" in attribute names; fix some docs bugs; add eavesdrop=true|false attribute
- * bus/driver.c (bus_driver_handle_add_match) (bus_driver_handle_remove_match): handle AddMatch, RemoveMatch messages
- * dbus/dbus-protocol.h (DBUS_SERVICE_ORG_FREEDESKTOP_BROADCAST):
get rid of broadcast service concept, signals are just always broadcast
- * bus/signals.c, bus/dispatch.c, bus/connection.c, bus/bus.c: mostly implement matching rules stuff (currently only exposed as signal connections)

2003-09-21 Mark McLoughlin <mark@skynet.ie>

- * doc/dbus-specification.sgml: Change the header field name to be an enum and update the rest of the spec to reference the fields using the conventional name.
- * dbus/dbus-protocol.h: update to reflect the spec.
- * doc/TODO: add item to remove the 4 byte alignment requirement.
- * dbus/dbus-message.c: Remove the code to generalise the header/body length and serial number header fields as named header fields so we can reference field names using the protocol values.
(append_int_field), (append_uint_field), (append_string_field): Append the field name as a byte rather than four chars.
(delete_int_or_uint_field), (delete_string_field): reflect the fact that the field name and typecode now occupy 4 bytes instead of 8.
(decode_string_field), (decode_header_data): update to reflect protocol changes and move the field specific encoding from decode_string_field() back into decode_header_data().
- * dbus/dbus-internals.[ch]: (_dbus_header_field_to_string): Add utility to aid debugging.
- * dbus/dbus-message-builder.c:

(append_string_field), (_dbus_message_data_load): Update to reflect protocol changes; Change the FIELD_NAME directive to HEADER_FIELD and allow it to take the field's conventional name rather than the actual value.

* test/data/*/*.message: Update to use HEADER_FIELD instead of FIELD_NAME; Always align the header on an 8 byte boundary *before* updating the header length.

2003-09-15 Havoc Pennington <hp@pobox.com>

* dbus/dbus-pending-call.c: add the get/set object data boilerplate as for DBusConnection, etc. Use generic object data for the notify callback.

* glib/dbus-gparser.c (parse_node): parse child nodes

* tools/dbus-viewer.c: more hacking on the dbus-viewer

* glib/dbus-gutils.c (_dbus_gutils_split_path): add a file to contain functions shared between the convenience lib and the installed lib

* glib/Makefile.am (libdbus_glib_1_la_LDFLAGS): add -export-symbols-regex to the GLib library

* dbus/dbus-object-tree.c
(_dbus_object_tree_dispatch_and_unlock):
fix the locking in here, and add a default handler for Introspect() that just returns sub-nodes.

2003-09-14 Havoc Pennington <hp@pobox.com>

* glib/dbus-gthread.c (dbus_g_thread_init): rename to make g_foo rather than gfoo consistent

* glib/dbus-gproxy.h: delete for now, move contents to dbus-glib.h, because the include files don't work right since we aren't in the dbus/ subdir.

* glib/dbus-gproxy.c (dbus_gproxy_send): finish implementing
(dbus_gproxy_end_call): finish
(dbus_gproxy_begin_call): finish

* glib/dbus-gmain.c (dbus_set_g_error): new

* glib/dbus-gobject.c (handle_introspect): include information about child nodes in the introspection

* dbus/dbus-connection.c (dbus_connection_list_registered): new function to help in implementation of introspection

* dbus/dbus-object-tree.c
(_dbus_object_tree_list_registered_and_unlock): new function

2003-09-12 Havoc Pennington <hp@pobox.com>

* glib/dbus-gidl.h: add common base class for all the foo_info types

* tools/dbus-viewer.c: add GTK-based introspection UI thingy similar to kdcop

* test/Makefile.am: try test srcdir -ef . in addition to test srcdir = ., one of them should work (yeah lame)

* glib/Makefile.am: build the "idl" parser stuff as a convenience library

* glib/dbus-gparser.h: make description_load routines return NodeInfo* not Parser*

* Makefile.am (SUBDIRS): build test dir after all library dirs

* configure.in: add GTK+ detection

2003-09-07 Havoc Pennington <hp@pobox.com>

* Make Doxygen contented.

2003-09-07 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: more updates

2003-09-06 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: partial updates

* bus/dbus-daemon-1.1.in: fix the config file docs for the zillionth time; hopefully I edited the right file this time.

* bus/config-parser.c (append_rule_from_element): support send_type, send_path, receive_type, receive_path

* bus/policy.c: add message type and path to the list of things that can be "firewalled"

2003-09-06 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_register_fallback): add this

(dbus_connection_register_object_path): make this not handle messages to paths below the given path

2003-09-03 Havoc Pennington <hp@pobox.com>

- * test/glib/Makefile.am: add this with random glib-linked test programs

- * glib/Makefile.am: remove the random test programs from here, leave only the unit tests

- * glib/dbus-gobject.c (_dbus_gobject_test): add test for uscore/javacaps conversion, and fix (get_object_property, set_object_property): change to .NET convention for mapping props to methods, set_FooBar/get_FooBar, since one language has such a convention we may as well copy it. Plus real methods in either getFooBar or get_foo_bar style won't collide with this convention.

2003-09-01 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gparser.c: implement

- * glib/dbus-gobject.c: start implementing skeletons support

- * configure.in: when disabling checks/assert, also define G_DISABLE_ASSERT and G_DISABLE_CHECKS

2003-09-01 Havoc Pennington <hp@pobox.com>

- * glib/Makefile.am: rearrange a bunch of files and get "make check" framework set up

2003-08-31 Havoc Pennington <hp@pobox.com>

- * fix build with --disable-tests

2003-08-30 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-connection.c: purge DBusMessageHandler

- * dbus/dbus-message-handler.c: remove DBusMessageHandler, just use callbacks everywhere

2003-08-30 Havoc Pennington <hp@pobox.com>

- * test/data/valid-config-files/system.d/test.conf: change to root for the user so warnings don't get printed

- * dbus/dbus-message.c: add dbus_message_get_path, dbus_message_set_path

- * dbus/dbus-object-tree.c (do_test_dispatch): add test of dispatching to a path

* dbus/dbus-string.c (_dbus_string_validate_path): add

* dbus/dbus-marshal.c (_dbus_demarshal_object_path): implement
(_dbus_marshal_object_path): implement

* dbus/dbus-protocol.h (DBUS_HEADER_FIELD_PATH): new header field
to contain the path to the target object
(DBUS_HEADER_FIELD_SENDER_SERVICE): rename
DBUS_HEADER_FIELD_SENDER to explicitly say it's the sender
service

2003-08-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-tree.c: write tests and fix the discovered
bugs

2003-08-29 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-tree.c: modify to allow overlapping paths to
be
registered
(struct DBusObjectSubtree): shrink this
a lot, since we may have a lot of them
(_dbus_object_tree_free_all_unlocked): implement
(_dbus_object_tree_dispatch_and_unlock): implement

2003-08-29 Havoc Pennington <hp@pobox.com>

* dbus/dbus-internals.h: fix _DBUS_N_GLOBAL_LOCKS

2003-08-28 Havoc Pennington <hp@pobox.com>

purge DBusObjectID

* dbus/dbus-connection.c: port to no ObjectID, create a
DBusObjectTree, rename ObjectTree to ObjectPath in public API

* dbus/dbus-connection.h (struct DBusObjectTreeVTable): delete
everything except UnregisterFunction and MessageFunction

* dbus/dbus-marshal.c: port away from DBusObjectID,
add DBUS_TYPE_OBJECT_PATH

* dbus/dbus-object-registry.[hc], dbus/dbus-object.[hc],
dbus/dbus-objectid.[hc]: remove these, we are moving to
path-based object IDs

2003-08-25 Havoc Pennington <hp@pobox.com>

Just noticed that dbus_message_test is hosed, I wonder when I
broke that. I thought make check was passing earlier...

- * dbus/dbus-object-tree.c: add new "object tree" to match DCOP container tree, will replace most of dbus-object-registry
- * dbus/dbus-string.c (_dbus_string_append_printf_valist): fix C99 screwup

2003-08-19 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-message.c (decode_string_field): support FIELD_SENDER (dbus_message_is_error): fix this function
- * bus/dbus-daemon-1.1: clarify logic on when <deny>/<allow> rules match
- * bus/policy.c (bus_client_policy_check_can_receive): fix code to reflect clarified man page (bus_client_policy_check_can_send): ditto
- * bus/session.conf.in: fixup
- * bus/system.conf.in: fixup

2003-08-18 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-hash.c (_dbus_hash_table_insert_two_strings): fix
- * dbus/dbus-message.c (_dbus_message_loader_queue_messages): fix dumb bug created earlier (wrong order of args to decode_header_data())
- * tools/dbus-send.c: port
- * tools/dbus-print-message.c (print_message): port
 - * test/data/*messages: port all messages over
 - * dbus/dbus-message-builder.c: support including message type
 - * bus/driver.c: port over
- * bus/dispatch.c: port over to new stuff
- * dbus/dbus-connection.c (_dbus_connection_new_for_transport): rename disconnect signal to "Disconnected"

2003-08-17 Havoc Pennington <hp@pobox.com>

This doesn't compile yet, but syncing up so I can hack on it from work. What are branches for if not broken code? ;-)

* dbus/dbus-protocol.h: remove DBUS_HEADER_FIELD_NAME, add DBUS_HEADER_FIELD_INTERFACE, DBUS_HEADER_FIELD_MEMBER, DBUS_HEADER_FIELD_ERROR_NAME

* dbus/dbus-hash.c: Introduce DBUS_HASH_TWO_STRINGS as hack to use

for the interface+member pairs
(string_hash): change to use g_str_hash algorithm
(find_direct_function, find_string_function): refactor these to share most code.

* dbus/dbus-message.c: port all of this over to support interface/member fields instead of name field

* dbus/dbus-object-registry.c: port over

* dbus/dbus-string.c (_dbus_string_validate_interface): rename from _dbus_string_validate_name

* bus/dbus-daemon-1.1: change file format for the <deny>/<allow> stuff to match new message naming scheme

* bus/policy.c: port over

* bus/config-parser.c: parse new format

2003-08-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-registry.c (add_and_remove_objects): remove broken assertion

* glib/dbus-gproxy.c: some hacking

2003-08-15 Havoc Pennington <hp@redhat.com>

* dbus/dbus-pending-call.c (dbus_pending_call_block): implement

* dbus/dbus-connection.c
(dbus_connection_send_with_reply_and_block): factor out
internals;
change to convert any error replies to DBusError instead of
returning them as a message

2003-08-15 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c,
dbus/dbus-pending-call.c: Finish the pending call stuff

2003-08-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-pending-call.c: start on new object that will replace
DBusMessageHandler and ReplyHandlerData for tracking outstanding

replies

- * dbus/dbus-gproxy.c: start on proxy object used to communicate with remote interfaces

- * dbus/dbus-gidl.c: do the boring boilerplate in here

2003-08-12 Havoc Pennington <hp@pobox.com>

- * bus/dispatch.c (bus_dispatch): make this return proper DBusHandlerResult to avoid DBUS_ERROR_UNKNOWN_METHOD

- * dbus/dbus-errors.c (dbus_set_error): use _dbus_string_append_printf_valist

- * dbus/dbus-string.c (_dbus_string_append_printf_valist) (_dbus_string_append_printf): new

- * dbus/dbus-errors.h (DBUS_ERROR_UNKNOWN_MESSAGE): change to UNKNOWN_METHOD

- * dbus/dbus-connection.c (dbus_connection_dispatch): handle DBUS_HANDLER_RESULT_NEED_MEMORY; send default error reply if a message is unhandled.

2003-08-11 Havoc Pennington <hp@pobox.com>

- * bus/test.c (client_disconnect_handler): change to return HANDLED (would have been REMOVE_MESSAGE)

- * dbus/dbus-object.h (enum DBusHandlerResult): rename to HANDLED/NOT_YET_HANDLED instead of REMOVE_MESSAGE/ALLOW_MORE_HANDLERS to make it clearer how it should be used.

2003-08-10 Havoc Pennington <hp@pobox.com>

- * tools/dbus-send.c (main): add --type argument, for now supporting only method_call and signal types.

- * tools/dbus-print-message.c: print message type

- * dbus/dbus-connection.c (_dbus_connection_new_for_transport): init connection->objects

- * doc/dbus-specification.sgml: fix sgml

- * bus/*.c: port over to object-instance API changes

- * test/test-service.c: ditto

- * dbus/dbus-message.c (dbus_message_create_header): allow #NULL

name, we will have to fix up the rest of the code to also handle this
(dbus_message_new): generic message-creation call
(set_string_field): allow appending name field

2003-08-06 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-object-registry.c: implement signal connection and dispatch
- * dbus/dbus-connection.c (_dbus_connection_unref_unlocked): new
- * dbus/dbus-internals.c (_dbus_memdup): new function

2003-08-02 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-message.c (dbus_message_get_no_reply) (dbus_message_set_no_reply): add these and remove set_is_error/get_is_error
- * dbus/dbus-protocol.h, doc/dbus-specification.sgml: remove the ERROR flag, since there's now an ERROR type

2003-08-01 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-object-registry.c (_dbus_object_registry_handle_and_unlock): implement
- * dbus/dbus-message.c (dbus_message_get_type): new function
- * doc/dbus-specification.sgml: add "type" byte to messages

2003-08-01 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-protocol.h (DBUS_MESSAGE_TYPE_*): introduce a message type enum to distinguish kinds of message (DBUS_HEADER_FLAG_NO_REPLY_EXPECTED): flag for a message that need not be replied to

2003-08-01 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-marshal.c: adapt to DBusObjectID changes (unpack_8_octets): fix no-64-bit-int bug
- * dbus/dbus-object-registry.c (validate_id): validate the connection ID bits, not just the instance ID.
- * dbus/dbus-connection.c (_dbus_connection_init_id): initialize the connection-global 33 bits of the object ID
- * dbus/dbus-object-registry.c (info_from_entry): fill in

object ID in the new way

* dbus/dbus-objectid.h: rather than high/low bits, specifically define server/client/instance bits.

2003-07-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_register_object): fix build

2003-07-13 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object.h (struct DBusObjectVTable): add padding fields to DBusObjectVTable and DBusObjectInfo

2003-07-12 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object-registry.c: implement unit test, fix bugs discovered in process

* dbus/dbus-connection.c: remove handler_table and register_handler(), add DBusObjectRegistry usage

* dbus/dbus-objectid.c (dbus_object_id_is_null) (dbus_object_id_set_null): new functions

2003-07-08 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object.c: implement some of this

* dbus/dbus-object-registry.c
(_dbus_object_registry_add_and_unlock): fill in the object_id out param
(_dbus_object_registry_new): handle OOM

2003-07-08 Havoc Pennington <hp@pobox.com>

* dbus/dbus-object.h: sketch out an API for registering objects with a connection, that allows us to use as little as 24 bytes per object and lets application code represent an object in any conceivable way.

* dbus/dbus-object-registry.c: implement the hard bits of the DBusConnection aspect of object API. Not yet wired up.

2003-07-06 Havoc Pennington <hp@pobox.com>

* dbus/dbus-marshal.c (_dbus_marshal_set_object_id): new function
(_dbus_marshal_object_id): new
(_dbus_demarshal_object_id): new
(_dbus_marshal_get_arg_end_pos): support object ID type, and consolidate identical switch cases. Don't conditionalize handling

of DBUS_TYPE_UINT64, need to handle the type always.
(_dbus_marshall_validate_arg): consolidate identical cases, and
handle DBUS_TYPE_OBJECT_ID

* dbus/dbus-objectid.c: new file with DBusObjectID data type.

* dbus/dbus-protocol.h: add DBUS_TYPE_OBJECT_ID

2003-09-28 Havoc Pennington <hp@pobox.com>

* real 0.13 release

2003-09-28 Havoc Pennington <hp@pobox.com>

* doc/Makefile.am (dbus-specification.html): testing a funky hack
to work with Debian db2html

2003-09-28 Havoc Pennington <hp@pobox.com>

* configure.in: 0.13

* doc/Makefile.am (dbus-test-plan.html): accept nonexistence of
stylesheet-images for benefit of Debian

Change back to using filesystem-linked sockets for the system
bus, so only root can create the default system bus address.

* bus/system.conf.in: change to use
DBUS_SYSTEM_BUS_DEFAULT_ADDRESS

* dbus/Makefile.am (INCLUDES): remove DBUS_SYSTEM_BUS_PATH define
from here.

* configure.in: define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
here, and AC_DEFINE DBUS_SYSTEM_PATH

2003-08-09 Anders Carlsson <andersca@codefactory.se>

* doc/TODO:
* doc/busconfig.dtd:
Add busconfig DTD.

2003-08-09 Anders Carlsson <andersca@codefactory.se>

* doc/dbus-specification.sgml:
Add activation reply values.

2003-08-05 Havoc Pennington <hp@redhat.com>

* configure.in: 0.12

2003-08-05 Anders Carlsson <andersca@codefactory.se>

* glib/dbus-gmain.c: (watch_fd_new), (watch_fd_ref),
(watch_fd_unref), (dbus_gsource_check), (dbus_gsource_dispatch),
(add_watch), (remove_watch), (create_source):
RefCount fds, fixes some reentrancy issues.

2003-07-30 Havoc Pennington <hp@redhat.com>

* dbus/dbus-bus.c (init_connections_unlocked): fix default system
bus address to be abstract if we have abstract sockets

* NEWS: update

2003-07-28 Havoc Pennington <hp@redhat.com>

* bus/messagebus.in: fix to avoid processname/servicename
confusion, from Michael Kearey
https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=100965

2003-07-23 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (dbus_message_iter_get_named):
fix from Andy Hanton to remove broken "+1"

2003-07-16 Havoc Pennington <hp@pobox.com>

* tools/dbus-launch.c (babysit): close stdout/stderr in the
babysitter process, as suggested by Thomas Leonard, so
an "eval `dbus-launch --exit-with-session`" will actually
return

2003-07-16 Havoc Pennington <hp@pobox.com>

* configure.in: print out EXPANDED_* variables in the summary at
the end; clean up the code that computes EXPANDED_ variables and
get the ones using exec_prefix right. Should make things work
when you build without --prefix

2003-06-29 Havoc Pennington <hp@pobox.com>

* mono/Test.cs (class Test): fire up a main loop and run it

* mono/DBus.cs (DBus): don't g_thread_init since it can only be
done once, the app has to do it

2003-06-26 Havoc Pennington <hp@pobox.com>

* mono/Connection.cs: set up connection with the glib main loop

2003-07-01 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.sgml: clarify the format of a type code,

change suggested by Jim Blandy

2003-06-29 Miloslav Trmac <mitr@volny.cz>

```
* doc/Makefile.am:
* tools/Makefile.am: Don't assume srcdir == builddir.

* dbus/dbus-memory.c (dbus_realloc): Don't check guards after
shrinking
the allocated block.
(_dbus_memory_test): New function.
* dbus/dbus-test.h: Add _dbus_memory_test ().
* dbus/dbus-test.c (dbus_internal_do_not_use_run_tests): Call it.

* dbus/dbus-message.c (decode_header_data): Use %.4s instead
of %c%c%c%c.
(dbus_message_new): Remove obsolete @todo.

* dbus/dbus-marshal.c (_dbus_marshal_set_int64)
(_dbus_marshal_set_uint64): Fix comment.

* dbus/dbus-message.c (append_int_field, append_uint_field):
Don't
hardcode FIELD_REPLY_SERIAL.

* dbus/dbus-mainloop.c (_dbus_loop_remove_watch)
(_dbus_loop_remove_timeout): Cast function pointers to (void *)
for %p

* configure.in: Add -D_POSIX_C_SOURCE=199309L -DBSD_SOURCE to
CFLAGS
and disable DBUS_USE_ATOMIC_INT_486 when --enable-ansi is used
```

2003-06-24 Havoc Pennington <hp@pobox.com>

```
* mono/*.cs: Use IntPtr.Zero instead of ((IntPtr) 0)
```

2003-06-23 Anders Carlsson <andersca@codefactory.se>

```
* configure.in:
* gcj/.cvsignore:
* gcj/Hello.java:
* gcj/Makefile.am:
* gcj/TestMessage.java: (TestMessage), (TestMessage.main):
* gcj/org/.cvsignore:
* gcj/org/Makefile.am:
* gcj/org/freedesktop/.cvsignore:
* gcj/org/freedesktop/Makefile.am:
* gcj/org/freedesktop/dbus/.cvsignore:
* gcj/org/freedesktop/dbus/Makefile.am:
* gcj/org/freedesktop/dbus/Message.java: (Message),
(Message.Message):
```

* gcj/org/freedesktop/dbus/natMessage.cc:
Fix the build system.

2003-06-22 Havoc Pennington <hp@pobox.com>

* mono/Connection.cs: add more bindings
* dbus/dbus-threads.c (dbus_threads_init): allow calling this more than once.

2003-06-22 Havoc Pennington <hp@pobox.com>

* mono/Connection.cs, mono/DBus.cs, mono/Error.cs:
Start wrapping more stuff.

2003-06-22 Havoc Pennington <hp@pobox.com>

* mono/Message.cs: implement Message.Wrap() that ensures we only have a single C# wrapper per DBusMessage, assuming it works which it probably doesn't.
* dbus/dbus-message.c (dbus_message_allocate_data_slot): new
(dbus_message_free_data_slot): new
(dbus_message_set_data): new
(dbus_message_get_data): new

2003-06-22 Havoc Pennington <hp@pobox.com>

* dbus/dbus-dataslot.c (_dbus_data_slot_allocator_unref)
(_dbus_data_slot_allocator_alloc): rework these to keep a reference count on each slot and automatically manage a global slot ID variable passed in by address
* bus/bus.c: convert to new dataslot API
* dbus/dbus-bus.c: convert to new dataslot API
* dbus/dbus-connection.c: convert to new dataslot API
* dbus/dbus-server.c: convert to new dataslot API
* glib/dbus-gmain.c: ditto
* bus/test.c: ditto
* bus/connection.c: ditto

2003-06-22 Anders Carlsson <andersca@codefactory.se>

* configure.in: Add AM_PROG_GCJ and move AM_PROG_LIBTOOL after the gcj checks so that the correct configuration tags will be added to libtool.

* dbus-glib-1.pc.in: No need to specify any includes since dbus-1.pc.in has those.

2003-06-22 Havoc Pennington <hp@pobox.com>

* mono/*, gcj/*, configure.in, Makefile.am:
Check in makefiles and subdirs for mono and gcj bindings.
Neither binding actually exists, just trying to get through
all the build and other boring bits.

2003-06-21 Philip Blundell <philb@gnu.org>

* tools/dbus-monitor.1: Updated.
* tools/dbus-send.1: Likewise.

2003-06-20 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-transport-unix.c (unix_handle_watch): Check
for hangup and error after checking read so we won't discard
pending data if both hangup and read are set.

2003-06-19 Philip Blundell <philb@gnu.org>

* tools/dbus-print-message.c (print_message): Handle BOOLEAN.
* tools/dbus-send.c: Accept both --system and --session.
* tools/dbus-monitor.c: Same here.

2003-06-19 Anders Carlsson <andersca@codefactory.se>

* glib/dbus-glib.h: Fix so that dbus-glib.h can be used
from C++ (Patch by Miloslav Trmac).

2003-06-15 Joe Shaw <joe@assbarn.com>

* configure.in: Check for socklen_t.
* dbus/dbus-sysdeps.c: Define socklen_t if it's not defined.
* test/test-segfault.c: Add #include <sys/time.h>
* tools/Makefile.am: Add DBUS_X_CFLAGS to the INCLUDES since
dbus-launch needs it.

2003-06-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_listen_unix_socket): don't use
SUN_LEN, it breaks abstract socket usage

* dbus/dbus-internals.c (_dbus_verbose_real): only print PID at starts of lines.

2003-06-04 Havoc Pennington <hp@pobox.com>

* dbus/dbus-server.c (dbus_server_listen): allow abstract sockets using unix:abstract=/foo, and when listening in a tmpdir i.e. unix:tmpdir=/tmp, always use abstract sockets if we can.

* dbus/dbus-transport.c (_dbus_transport_open): support unix:abstract=/foo

* dbus/dbus-server-unix.c (_dbus_server_new_for_domain_socket): support abstract sockets

* dbus/dbus-transport-unix.c (_dbus_transport_new_for_domain_socket): support abstract sockets

* dbus/dbus-sysdeps.c (_dbus_connect_unix_socket): add "abstract" toggle as an argument, implement abstract namespace support (_dbus_listen_unix_socket): ditto

* configure.in: add --enable-abstract-sockets and implement a configure check for autodetection of the right value.

2003-06-01 Havoc Pennington <hp@pobox.com>

* tools/dbus-cleanup-sockets.c: add utility to clean up sockets in /tmp (though on Linux this will end up being useless, when we add abstract namespace support)

* configure.in: define DBUS_SESSION_SOCKET_DIR in addition to subst'ing it

2003-05-28 Colin Walters <walters@verbum.org>

* tools/dbus-monitor.c (main): Fix silly typo (s/--session/--system/).

2003-05-18 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c (dbus_message_new): Remove @todo.

2003-05-17 Colin Walters <walters@gnu.org>

* tools/dbus-send.c: Don't exit with an error code if --help was passed. Default to using the session bus instead of the system one.

* tools/dbus-launch.c: Ditto.

* tools/dbus-monitor.c: Ditto.

- * tools/dbus-send.1: Update with new arguments.
- * tools/dbus-launch.c: Emit code to export variables. New arguments -s and -c to specify shell syntax, and a bit of code to autodetect syntax. Also, allow specifying a program to run.
- * tools/dbus-launch.1: Update with new arguments.
- * tools/dbus-send.1: Ditto.
- * tools/dbus-monitor.1: Ditto.

2003-05-17 Havoc Pennington <hp@pobox.com>

- * bus/config-parser.c (merge_included): merge in policies from child configuration file.
- * bus/policy.c (bus_policy_merge): function to merge two policies together

2003-05-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-connection.c: disable verbose lock spew
- * tools/dbus-send.c: add --print-reply command line option
- * tools/dbus-print-message.h (print_message): new util function shared by dbus-send and dbus-monitor
- * tools/dbus-monitor.c (handler_func): exit on disconnect
- * dbus/dbus-transport-unix.c (do_reading): if the transport is disconnected, don't try to use the read_watch
- * dbus/dbus-watch.c (dbus_watch_get_enabled): assert watch != NULL
so we can find this bug more easily

2003-05-16 Havoc Pennington <hp@redhat.com>

- * bus/policy.c (free_rule_list_func): avoid a crash when passed NULL as DBusHashTable is annoyingly likely to do.

2003-05-16 Colin Walters <walters@verbum.org>

- * tools/dbus-monitor.c: Add --session argument and usage() function.
- * tools/dbus-monitor.1: Update with new --session arg.
- * bus/Makefile.am (install-data-hook): Create

`$(libdir)/dbus-1.0/services` so that the session bus is happy.

2003-05-15 Havoc Pennington <hp@redhat.com>

* `dbus/dbus-sysdeps.c` (`_dbus_atomic_dec`, `_dbus_atomic_inc`): work on non-x86. `ifdef`'s are evil.

2003-05-15 Havoc Pennington <hp@redhat.com>

* `configure.in`: 0.11

* NEWS: update

* `bus/Makefile.am` (`initddir`): apparently we are supposed to put `init` scripts in `/etc/rc.d/init.d` not `/etc/init.d`

* `bus/Makefile.am`: remove the "you must `--enable-tests` to make check" as it broke `distcheck`

* `bus/Makefile.am` (`install-data-hook`): create `/etc/dbus-1/system.d`

2003-05-13 James Willcox <jwillcox@gnome.org>

* `configure.in`:

* `bus/activation.c`: (`bus_activation_service_created`), (`bus_activation_activate_service`):

* `bus/driver.c`: (`bus_driver_send_service_deleted`), (`bus_driver_send_service_created`), (`bus_driver_send_service_lost`),

(`bus_driver_send_service_acquired`),

(`bus_driver_send_welcome_message`),

(`bus_driver_handle_list_services`):

* `bus/session.conf.in`:

* `dbus/dbus-bus.c`: (`dbus_bus_acquire_service`),

(`dbus_bus_service_exists`), (`dbus_bus_activate_service`):

* `dbus/dbus-bus.h`:

Add some convenience API which lets you activate a service, and did a

bunch of `s/0/DBUS_TYPE_INVALID/` in calls to

`dbus_message_append_args()`

and `dbus_message_get_args()`

2003-05-11 Havoc Pennington <hp@pobox.com>

* `dbus/dbus-marshal.c` (`_dbus_marshal_validate_arg`): fix to avoid calling `_dbus_marshal_validate_arg()` for every byte in a byte array, etc.

* `dbus/dbus-message-handler.c`: use atomic reference counting to reduce number of locks slightly; the global lock in here sucks

```

* dbus/dbus-connection.c
  (_dbus_connection_update_dispatch_status_and_unlock): variant of
update_dispatch_status that can be called with lock held; then
use
in a couple places to reduce locking/unlocking
(dbus_connection_send): hold the lock over the whole function
instead of acquiring it twice.

* dbus/dbus-timeout.c (_dbus_timeout_new): handle OOM

* bus/connection.c (bus_connections_setup_connection): fix access
to already-freed memory.

* dbus/dbus-connection.c: keep a little cache of linked list
nodes, to avoid using the global linked list alloc lock in the
normal send-message case. Instead we just use the connection lock
that we already have to take.

* dbus/dbus-list.c (_dbus_list_find_last): new function

* dbus/dbus-sysdeps.c (_dbus_atomic_inc, _dbus_atomic_dec):
change to use a struct for the atomic type; fix docs,
they return value before increment, not after increment.

* dbus/dbus-string.c (_dbus_string_append_4_aligned)
(_dbus_string_append_8_aligned): new functions to try to
microoptimize this operation.
(reallocate_for_length): break this out of set_length(), to
improve profile info, and also so we can consider inlining the
set_length() part.

* dbus/dbus-message.c (dbus_message_new_empty_header): init data
strings with some preallocation, cuts down on our calls to
realloc
a fair bit. Though if we can get the "move entire string to empty
string" optimization below to kick in here, it would be better.

* dbus/dbus-string.c (_dbus_string_move): just call
_dbus_string_move_len
(_dbus_string_move_len): add a special case for moving
an entire string into an empty string; we can just
swap the string data instead of doing any reallocs.
(_dbus_string_init_preallocated): new function

```

2003-05-11 Havoc Pennington <hp@pobox.com>

Write a "test-profile" that does echo client-server with threads; profile reveals lock contention, memcpy/realloc of buffers, and UTF-8 validation as hot spots. 20% of lock contention eliminated with dbus_atomic_inc/dec implementation on x86. Much remaining contention is global mempool locks for GList and DBusList.

- * dbus/dbus-sysdeps.c (`_dbus_atomic_inc`, `_dbus_atomic_dec`): add x86 implementation
- * dbus/dbus-connection.c (struct `DBusConnection`): use `dbus_atomic_t` for the reference count
- * dbus/dbus-message.c (struct `DBusMessage`): declare `dbus_atomic_t` values as volatile
- * configure.in: code to detect ability to use atomic integer operations in assembly, from GLib patch
- * dbus/dbus-internals.c (`_dbus_verbose_real`): call `getpid` every time, tired of it being wrong in threads and forked processes
- * glib/test-profile.c: a little program to bounce messages back and forth between threads and eat CPU
- * dbus/dbus-connection.c: add debug spew macros for debugging thread locks; include `config.h` at top; fix deadlock in `dbus_connection_flush()`

2003-05-08 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-spawn.c: s/_exit/exit/ because it was keeping gcov data from getting written, and there wasn't a good reason to use `_exit` really.
- * test/decode-gcov.c (`mark_inside_dbus_build_tests`): don't count `dbus_verbose` lines in test coverage
(main): add list of functions sorted by # of untested blocks to the coverage report
- * dbus/dbus-mempool.c: put some test-only code in `DBUS_BUILD_TESTS`
- * dbus/dbus-marshal.c (`_dbus_marshall_test`): extend test coverage
- * dbus/dbus-message-handler.c (`_dbus_message_handler_test`): extend test coverage
- * test/data/auth/cancel.auth-script: test canceling an authentication
- * dbus/Makefile.am: remove `dbus-server-debug.[hc]` for now, as they aren't used. in CVS history if we end up needing them.

2003-05-04 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-message-handler.c (`_dbus_message_handler_test`): add

unit test

* dbus/dbus-marshal.c (dbus_demarshal_string_array): fix this function, which assumed length was in # of strings, not bytes

* dbus/dbus-message.c (dbus_message_test): add tests for some missing coverage

* dbus/dbus-connection.c (dbus_connection_queue_received_message): disable function for now, we are only using it in test mode

* dbus/dbus-message.c (dbus_message_loader_queue_messages): remove a mistaken FIXME

2003-05-04 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_preallocate_send): unlock mutex on successful return, patch from Anders Gustafsson

2003-05-04 Havoc Pennington <hp@pobox.com>

* dbus-glib-1.pc.in (Requires): fix dependencies, from Anders Gustafsson

2003-05-04 Havoc Pennington <hp@pobox.com>

* tools/dbus-launch.c: implement

* bus/main.c (main), bus/bus.c (bus_context_new): implement --print-pid and --fork

2003-05-03 Havoc Pennington <hp@redhat.com>

* dbus/dbus-address.c (dbus_parse_address): fix bug when a key in the address had no value, and add to test suite. Fix and regression test from Miloslav Trmac

2003-05-03 Havoc Pennington <hp@pobox.com>

* dbus/dbus-watch.c (dbus_watch_handle): warn and return if a watch is invalid when handled

* tools/Makefile.am, tools/dbus-launch.c, tools/dbus-launch.1:
add
dbus-launch utility to launch the bus from a shell script.
Didn't
actually implement dbus-launch yet, it's just a placeholder
still.

2003-05-03 Havoc Pennington <hp@pobox.com>

* bus/Makefile.am, bus/dbus-daemon-1.1.in: man page for the daemon; also documents daemon config file, so replaces doc/config-file.txt. Corrected some stuff from config-file.txt in the process of moving it.

2003-05-03 Havoc Pennington <hp@pobox.com>

* tools/Makefile.am, tools/dbus-send.1, tools/dbus-monitor.1: add some man pages

2003-05-03 Colin Walters <walters@verbum.org>

* dbus/dbus-sysdeps.c (fill_user_info): Test against DBUS_UID_UNSET to determine whether to do a uid lookup or not.

* Makefile.am: Update to use new .pc versioning scheme.

2003-05-02 Havoc Pennington <hp@redhat.com>

* bus/system.conf.in: allow send/receive to/from message bus service

2003-04-30 Havoc Pennington <hp@redhat.com>

* configure.in: print a note when building with unit tests and without assertions

2003-04-30 Havoc Pennington <hp@redhat.com>

* Makefile.am: add a check-local that complains if you didn't configure with --enable-tests

2003-04-29 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c: docs cleanups

* dbus/dbus-types.h: add docs on int64 types

* dbus/dbus-memory.c: fix docs to avoid putting private API in public API docs section

2003-04-29 Havoc Pennington <hp@redhat.com>

* dbus-1.pc.in, dbus-glib-1.pc.in: rename these from dbus-1.0.pc.in, dbus-glib-1.0.pc.in. As these change with the parallel install API version, not with the D-BUS package version.

* HACKING: move some of README over here

* README: updates, and document API/ABI policy

* configure.in: reindentation

2003-04-29 Havoc Pennington <hp@redhat.com>

- * dbus/dbus.h: add "you have to define DBUS_API_SUBJECT_TO_CHANGE to use this library" to be sure people have the right expectations.

2003-04-28 Havoc Pennington <hp@redhat.com>

- * configure.in: add --enable-docs which by default is auto yes if doxygen and db2html found, no otherwise; but can be forced on/off

- * doc/Makefile.am: conditionalize whether to build docs on --enable-docs

2003-04-28 Havoc Pennington <hp@redhat.com>

- * configure.in: 0.10

- * NEWS: update

- * bus/system.conf.in: add <includedir>system.d</includedir>

- * dbus/dbus-userdb.c (_dbus_user_database_lookup): fix bug when username was provided but not uid

- * bus/config-parser.c (struct BusConfigParser): keep track of whether the parser is toplevel or was included; change some of the error handling if it's included.

2003-04-27 Havoc Pennington <hp@pobox.com>

Unbreak my code...

- * dbus/dbus-transport.c (_dbus_transport_get_dispatch_status): report correct status if we finish processing authentication inside this function.

- * bus/activation.c (try_send_activation_failure): use bus_transaction_send_error_reply

- * bus/connection.c (bus_connection_get_groups): return an error explaining the problem

- * bus/bus.c (bus_context_check_security_policy): implement restriction here that inactive connections can only send the hello message. Also, allow bus driver to send anything to any recipient.

- * bus/connection.c (bus_connection_complete): create the BusClientPolicy here instead of on-demand.

- (bus_connection_get_policy): don't return an error

* dbus/dbus-message.c (dbus_message_new_error_reply): allow NULL sender field in message being replied to

* bus/bus.c (bus_context_check_security_policy): fix silly typo causing it to return FALSE always

* bus/policy.c (bus_client_policy_check_can_send): fix bug where we checked sender rather than destination

2003-04-25 Havoc Pennington <hp@redhat.com>

test suite is slightly hosed at the moment, will fix soon

* bus/connection.c (bus_connections_expire_incomplete): fix to properly disable the timeout when required
(bus_connection_set_name): check whether we can remove incomplete connections timeout after we complete each connection.

* dbus/dbus-mainloop.c (check_timeout): fix this up a bit, probably still broken.

* bus/services.c (bus_registry_acquire_service): implement max number of services owned, and honor allow/deny rules on which services a connection can own.

* bus/connection.c (bus_connection_get_policy): report errors here

* bus/activation.c: implement limit on number of pending activations

2003-04-25 Havoc Pennington <hp@redhat.com>

* dbus/dbus-transport.c (_dbus_transport_get_unix_user): fix bug where we used >= 0 instead of != DBUS_UID_UNSET.

2003-04-25 Havoc Pennington <hp@redhat.com>

* glib/dbus-gmain.c (remove_watch): fix for a crash when watches were toggled without add/remove, fix from Anders Gustafsson

2003-04-24 Havoc Pennington <hp@redhat.com>

* test/data/valid-config-files/basic.conf: add <limit> tags to this test

* bus/config-parser.h, bus/config-parser.c, bus/bus.c: Implement <limit> tag in configuration file.

2003-04-24 Havoc Pennington <hp@redhat.com>

* bus/dispatch.c: somehow missed some name_is

* dbus/dbus-timeout.c (_dbus_timeout_set_enabled)
(_dbus_timeout_set_interval): new

* bus/connection.c (bus_connections_setup_connection): record
time
when each connection is first set up, and expire them after the
auth timeout passes.

2003-04-24 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_name_is): rename
(dbus_message_service_is): rename
(dbus_message_sender_is): rename
(dbus_message_get_service): rename

2003-04-24 Havoc Pennington <hp@redhat.com>

* configure.in: add --enable-checks

* dbus/dbus-message.c (dbus_message_new): reverse name/service
arguments

* dbus/dbus-connection.c (dbus_connection_preallocate_send): fix
to use thread locks.
(_dbus_connection_handler_destroyed_locked): move some private
functions into proper docs group

* dbus/dbus-internals.h: add _dbus_return_if_fail,
_dbus_return_val_if_fail

Throughout: use dbus_return_if_fail

2003-04-23 James Willcox <jwillcox@gnome.org>

* glib/dbus-glib.h:
* glib/dbus-gmain.c: (add_timeout), (wakeup_main),
(create_source),
(dbus_connection_setup_with_g_main),
(dbus_server_setup_with_g_main):
* glib/test-dbus-glib.c: (main):
* glib/test-thread-client.c: (main):
* glib/test-thread-server.c: (new_connection_callback), (main):
* tools/dbus-monitor.c: (main):

Added a GMainContext argument to
dbus_connection_setup_with_g_main()
and dbus_server_setup_with_g_main().

2003-04-20 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: document the restrictions on message and service names

2003-04-22 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c, dbus/dbus-marshal.c: add 64-bit integer support, and do some code cleanups to share more code and speed up array marshal/demarshal.

* dbus-1.0.pc.in (Cflags): put libdir include file in cflags

* configure.in: generate dbus-arch-deps.h

* dbus/dbus-protocol.h (DBUS_TYPE_INT64, DBUS_TYPE_UINT64): add 64-bit typecodes

2003-04-22 Havoc Pennington <hp@redhat.com>

* test/data/valid-messages/opposite-endian.message: fix test to use proper type for rply field

* test/data/invalid-messages: add tests for below validation

* dbus/dbus-message.c (decode_header_data): validate field types, and validate that named fields are valid names (decode_name_field): consider messages in the org.freedesktop.Local. namespace to be invalid.

* dbus/dbus-string.c (_dbus_string_validate_name): new

2003-04-19 Havoc Pennington <hp@pobox.com>

* bus/driver.c (bus_driver_handle_hello): check limits and return an error if they are exceeded.

* bus/connection.c: maintain separate lists of active and inactive connections, and a count of each. Maintain count of completed connections per user. Implement code to check connection limits.

* dbus/dbus-list.c (_dbus_list_unlink): export

* bus/bus.c (bus_context_check_security_policy): enforce a maximum number of bytes in the message queue for a connection

2003-04-18 Havoc Pennington <hp@pobox.com>

* dbus/dbus-auth.c (record_mechanisms): memleak fixes

* dbus/dbus-sysdeps.c (_dbus_string_save_to_file): fix some memleaks

- * dbus/dbus-keyring.c (add_new_key): fix a memleak, and on realloc be sure to update the pointer in the keyring
- * dbus/dbus-string.c (_dbus_string_zero): compensate for align offset to avoid writing to unallocated memory
- * dbus/dbus-auth.c (process_rejected): return FALSE if we fail to try the next mechanism, so we properly handle OOM
- * dbus/dbus-keyring.c (_dbus_keyring_new_homedir): fix double-free on OOM.
- (_dbus_keyring_new): fix OOM bug
- (_dbus_keyring_new_homedir): always set error; impose a maximum number of keys we'll load from the file, mostly to speed up the test suite and make its OOM checks more useful, but also for general sanity.
- * dbus/dbus-auth.c (process_error_server): reject authentication if we get an error from the client
- (process_cancel): on cancel, send REJECTED, per the spec
- (process_error_client): send CANCEL if we get an error from the server.

2003-04-18 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-mainloop.c (_dbus_loop_iterate): fix UMR in verbose debug spew
- * dbus/dbus-auth.c (handle_client_data_cookie_shal_mech): fix OOM handling problem
- * dbus/dbus-keyring.c (_dbus_keyring_new_homedir): only whine about DBUS_TEST_HOMEDIR once
- * bus/Makefile.am (TESTS_ENVIRONMENT): put DBUS_TEST_HOMEDIR in the environment
- * bus/dispatch.c (bus_dispatch_shal_test): actually load shal config file so we test the right thing

Throughout: assorted docs improvements

2003-04-18 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gmain.c: adapt to watch changes
- * bus/bus.c, bus/activation.c, etc.: adjust to watch changes
- * dbus/dbus-server.h: remove dbus_server_handle_watch

- * dbus/dbus-connection.h: remove dbus_connection_handle_watch
- * dbus/dbus-watch.c (dbus_watch_handle): change DBusWatch to work like DBusTimeout, so we don't need dbus_connection_handle_watch etc.

2003-04-17 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-userdb.c, dbus/dbus-sysdeps.c: redo all the passwd database usage so it all goes via the DBusUserDatabase cache.

2003-04-17 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-mainloop.c (_dbus_loop_iterate): fix logic so that if there was an OOM watch we skipped, we always return TRUE so we iterate again to have a look at it again. Fixes test suite hang. Code rearrangement also lets us lose some memset and only iterate over callbacks once.

- * bus/driver.c (bus_driver_handle_message): sense of test for reply was backward

2003-04-16 Havoc Pennington <hp@pobox.com>

- * doc/dbus-specification.sgml: make spec say serials are unsigned
- * dbus/dbus-message.h: change message serials to unsigned
- * dbus/dbus-connection.c: adapt to message serials being unsigned

2003-04-15 Havoc Pennington <hp@pobox.com>

- * bus/bus.c: create and keep around a shared DBusUserDatabase object.

- * bus/connection.c (bus_connection_get_groups): don't cache groups for user in the connection object, since user database object now does that.

2003-04-16 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message.c (_dbus_message_add_size_counter): keep a list of size counters
- (_dbus_message_loader_putback_message_link): put back a popped link

- * dbus/dbus-connection.c (dbus_connection_set_max_live_messages_size): rename max_received_size
- (dbus_connection_get_outgoing_size): get size of outgoing queue
- (_dbus_connection_set_connection_counter): remove this cruft

2003-04-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-userdb.c: user database abstraction, mostly to get caching, but at some point we might want to be able to use a different database.

* bus/dispatch.c (bus_dispatch_shal_test): add a test that uses SHA1 conf file to test the shal auth mechanism, since the regular test always uses EXTERNAL when available.

* configure.in,
test/data/valid-config-files/debug-allow-all-shal.conf.in:
add conf file that requires use of shal auth

2003-04-13 Havoc Pennington <hp@pobox.com>

* tools/dbus-send.c, tools/dbus-monitor.c: two utility programs
from Philip Blundell to send messages and monitor them.

2003-04-13 Havoc Pennington <hp@pobox.com>

* dbus/dbus-mainloop.c: fix some reentrancy issues by recounting callbacks

* test/data/valid-config-files/debug-allow-all.conf.in: allow all users

* dbus/dbus-transport.c (_dbus_transport_get_dispatch_status):
fix to only recover unused bytes if we're already authenticated
(_dbus_transport_get_is_authenticated): fix to still mark us
authenticated if there are unused bytes.

* bus/dispatch.c: implement security policy checking

* bus/connection.c (bus_transaction_send_from_driver): new

* bus/bus.c (bus_context_check_security_policy): new

* bus/dispatch.c (send_service_nonexistent_error): delete this,
now we just set the DBusError and it gets converted to an error
reply.

* bus/connection.c (allow_user_function): enable code using
actual data from the config file

* bus/policy.c (list_allows_user): handle wildcard rules for
user/group connection perms

2003-04-13 Havoc Pennington <hp@pobox.com>

- * bus/config-parser.c: Load up the BusPolicy and BusPolicyRules
- * dbus/dbus-sysdeps.c (_dbus_get_user_id): new function
- * bus/policy.c (bus_policy_append_mandatory_rule)
(bus_policy_append_default_rule, bus_policy_append_user_rule)
(bus_policy_append_group_rule): new functions

2003-04-12 Havoc Pennington <hp@pobox.com>

- * bus/config-parser.c (bus_config_parser_new): fix a memleak
- * dbus/dbus-sysdeps.c: change DBusCredentials to use longs for the pid/gid/uid, just for paranoia.
- * test/break-loader.c (randomly_do_n_things): find a byte containing a type code, and randomly change it to a different type code.

2003-04-12 Havoc Pennington <hp@pobox.com>

- * bus/policy.h: change BusPolicy to be the thing from the config file, and rename old BusPolicy to BusClientPolicy
- * bus/bus.c, bus/connection.c, bus/config-parser.c: change to match change in how policy works
- * dbus/dbus-internals.h: mark assert_not_reached as __attribute__((noreturn))__

2003-04-11 Havoc Pennington <hp@redhat.com>

- * doc/dbus-specification.sgml: fix a spot with the wrong name for the broadcast service. Use boolean return for ServiceExists.

2003-04-11 Havoc Pennington <hp@redhat.com>

- * configure.in: add another directory to look for qt in.

2003-04-11 Havoc Pennington <hp@redhat.com>

- * AUTHORS: add Colin Walters

2003-04-11 Havoc Pennington <hp@redhat.com>

- * NEWS: update
- * configure.in: 0.9

2003-04-11 Havoc Pennington <hp@redhat.com>

* bus/messagebus.in: remove pid file when stopping the message bus, since the bus won't have privileges to remove it itself.

2003-04-11 Havoc Pennington <hp@redhat.com>

* bus/bus.c (bus_context_new): move credentials change after creating pidfile

2003-04-11 Havoc Pennington <hp@pobox.com>

* test/decode-gcov.c: add "below average functions" to the coverage report, and change how some of the code works.

* bus/test-main.c: bracket some stuff in DBUS_BUILD_TESTS so it's not in the coverage stats.

* test/test-service.c (main): use _dbus_verbose not fprintf in a couple places so running the test suite doesn't result in megaspan.

2003-04-11 Havoc Pennington <hp@pobox.com>

* bus/dispatch.c (check_existent_service_activation): accept a no memory error in a place we didn't before

* bus/test.c (bus_test_run_everything): remove hacky "do it twice in case the first one failed," since the test suite is less broken now.

2003-04-10 Havoc Pennington <hp@pobox.com>

* bus/dispatch.c (check_segfault_service_activation): add test for launching an executable that just crashes.

* test/test-segfault.c (main): try setting coredumpsize to 0 so we don't leave a million cores. We'll see how portable this is.

2003-04-10 Havoc Pennington <hp@pobox.com>

* dbus/dbus-spawn.c (_dbus_spawn_async_with_babysitter): move all the possible parent failures before we fork, so that we don't fail to create a babysitter after creating the child.

* bus/activation.c (bus_activation_activate_service): kill child if we don't successfully complete the activation.

2003-04-10 Havoc Pennington <hp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_flush): don't spin on the connection if it's disconnected

```
* bus/activation.c (bus_activation_service_created): use new
transaction features to roll back removal of pending activation
if
we don't successfully create the service after all. Don't remove
pending activation if the function fails.

* dbus/dbus-list.c (_dbus_list_insert_before_link)
(_dbus_list_insert_after_link): new code to facilitate
services.c fixes

* dbus/dbus-hash.c (_dbus_hash_table_insert_string_preallocated):
new functionality, so we can preallocate the ability to insert
into a hash table.

* bus/connection.c (bus_transaction_add_cancel_hook): new
function
allowing us to put custom hooks in a transaction to be used for
cancelling said transaction

* doc/dbus-specification.sgml: add some discussion of secondary
service owners, and disallow zero-length service names

* bus/services.c (bus_registry_acquire_service): new function,
splits out part of bus_driver_handle_acquire_service() and fixes
a bug where we didn't remove the service doing the acquiring
from the secondary queue if we failed to remove the current owner
from the front of the queue.
```

2003-04-10 Alexander Larsson <alexl@redhat.com>

```
* doc/dbus-specification.sgml:
s/org.freedesktop.Broadcast/org.freedesktop.DBus.Broadcast/
```

2003-04-10 Alexander Larsson <alexl@redhat.com>

```
* bus/.cvsignore:
* glib/.cvsignore:
* test/.cvsignore:
Added files to cvsignore

* dbus/dbus-message.h:
* dbus/dbus-message.c: (dbus_message_iter_get_named):
Make get_named() take two out argument and return a boolean.
(dbus_message_iter_get_args_valist):
Update usage of get_named().
(dbus_message_iter_append_byte):
Fix typo
(dbus_message_iter_append_named)
Fix typo
(message_iter_test), (check_message_handling_type),
(_dbus_message_test):
```


More tests.

2003-04-10 Alexander Larsson <alexl@redhat.com>

- * dbus/dbus-marshal.[ch]:
Add array_type_pos argument to _dbus_marshal_validate_arg.
Let you pass a NULL end_pos to _dbus_marshal_validate_type.
- * dbus/dbus-message.[ch]:
Multi-dimensional arrays have full type specification in the
outermost array. Iter code re-arranged to handle this.
Added some more iter tests.
- * doc/dbus-specification.sgml:
Add me to authors.
Remove old FIXME.
Update new array encoding description.
Correct DBUS_SERVICE_FLAGS_REPLACE_EXISTING description.
- * test/data/invalid-messages/array-with-mixed-types.message:
* test/data/valid-messages/array-of-array-of-uint32.message:
Change to the new array format.
- * test/data/invalid-messages/too-short-dict.message:
Fix bug in test.
- * test/data/valid-messages/recursive-types.message:
Fix up and extend test.

2003-04-10 Havoc Pennington <hp@pobox.com>

- * bus/dispatch.c: lots of fixes
- * dbus/dbus-mainloop.c (_dbus_loop_dispatch): export
(_dbus_loop_iterate): remove old "quit if no callbacks" code,
that was crack, broke the test service.
- * dbus/dbus-transport.c (_dbus_transport_open): fix error
handling to avoid piling up errors if we get a failure on the
first address.
- * dbus/dbus-internals.c (_dbus_real_assert_not_reached): include
pid in assertion failures.
- * dbus/dbus-mainloop.c (_dbus_loop_iterate): use static arrays up
to some fixed size of file descriptor array. Don't return TRUE
anytime a timeout exists, that led to lots of busy loop silliness
in the tests.

2003-04-09 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-mainloop.c (check_timeout): fix timeouts, I thought

I'd checked this in earlier but hadn't.

2003-04-09 Havoc Pennington <hp@redhat.com>

- * bus/dispatch.c (bus_dispatch_test): get a bit further through the activation test (man this is getting old!)

2003-04-09 Havoc Pennington <hp@redhat.com>

- * test/test-utils.c: use dispatch status function to fix this up
- * bus/connection.c (connection_watch_callback): don't dispatch from here
(connection_timeout_callback): don't dispatch from here
(bus_connections_setup_connection): set the dispatch status function
(bus_connection_disconnected): unset it
- * dbus/dbus-mainloop.c (_dbus_loop_queue_dispatch): new function used to add a connection to be dispatched
(_dbus_loop_iterate): do the dispatching at the end of each iteration
- * dbus/dbus-connection.c
(dbus_connection_set_dispatch_status_function): new function allowing us to fix up main loop usage
(_dbus_connection_last_unref): free all the various function user data
(dbus_connection_dispatch): call the DispatchStatusFunction whenever this function returns
(dbus_connection_handle_watch): call DispatchStatusFunction
(dbus_connection_send_with_reply_and_block): call DispatchStatusFunction
(reply_handler_timeout): call DispatchStatusFunction
(dbus_connection_flush): call DispatchStatusFunction

2003-04-09 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-bus.c (dbus_bus_register): fix up error handling and a memory leak
- * bus/dispatch.c (check_service_activated): fix bug in test
- * dbus/dbus-mainloop.c (check_timeout): fix this up
- * dbus/dbus-internals.c (_dbus_verbose_real): include PID in verbose output so we can sort out output from different processes,
e.g. in the activation case.

2003-04-08 Colin Walters <walters@gnu.org>

- * bus/bus.c (struct BusContext) [pidfile]: New member, to store the pid file.
- (bus_context_new): Set it.
- (bus_context_unref): Use it to delete the pid file.

2003-04-08 Havoc Pennington <hp@redhat.com>

- * test/data/invalid-messages/array-with-mixed-types.message: regression test that fails for the moment
- * dbus/dbus-test.c (dbus_internal_do_not_use_run_tests): reorder tests for convenience
- * dbus/dbus-marshal.c (_dbus_marshal_validate_arg): don't allow array of nil, it broke things.
- * test/data/invalid-messages/array-of-nil.message: regression test
- * test/data/valid-messages/array-of-array-of-uint32.message: happened to write this so added it to suite

2003-04-08 Havoc Pennington <hp@redhat.com>

- * bus/driver.c (bus_driver_handle_acquire_service): init retval/reply before checking name
- * dbus/dbus-marshal.c (_dbus_marshal_validate_arg): add a recursion depth argument
- * dbus/dbus-message.h (struct DBusMessageIter): put some padding in the public struct for future extension
- * dbus/dbus-message-builder.c (_dbus_message_data_load): fix typo
- * dbus/dbus-marshal.c (_dbus_marshal_validate_arg): fix a verbose message
- * doc/dbus-specification.sgml: fix typo

2003-04-08 Alexander Larsson <alexl@redhat.com>

Implemented recursive types, named types and new-style iters

- * bus/driver.c:
- * glib/test-thread-client.c: (thread_func):
- * glib/test-thread-server.c: (handle_test_message):
- * test/test-service.c: (handle_echo):

Update to new api

- * dbus/Makefile.am:

- * dbus/dbus-dict.c:
- * dbus/dbus-dict.h:
- * dbus/dbus.h
- Remove DBusDict

- * dbus/dbus-internals.c: (_dbus_type_to_string):
- Update for new types.

- * dbus/dbus-marshal.[ch]:
- Implement recursive types and the new marshalling format.
- Remove hardcoded dict marshalling.
- Marshal named types.

- * dbus/dbus-message-builder.c:
- Add BYTE_ARRAY.
- Remove references to old types

- * dbus/dbus-message.[ch]:
- New non-refcounted iter API that supports recursive iters.
- Use iters for appending, including support for recursive iters.
- Add byte and named type support.
- Update everything to new marshalling formats.
- Add tests for new API.

- * dbus/dbus-protocol.h:
- Remove old array types.
- Add types: BYTE, ARRAY, DICT, NAMED

- * dbus/dbus-string.c:
- * dbus/dbus-sysdeps.c:
- Make parse_double locale safe.

- * dbus/dbus-test-main.c:
- Call setlocale.

- * dbus/dbus-test.c:
- Kill dict test

- * doc/dbus-specification.sgml:
- Update spec

- * test/data/incomplete-messages/missing-body.message:
- * test/data/invalid-messages/bad-boolean.message:
- * test/data/invalid-messages/bad-boolean-array.message:
- * test/data/invalid-messages/boolean-array-length-too-long.message-raw:
- * test/data/invalid-messages/boolean-has-no-value.message-raw:
- * test/data/invalid-messages/too-short-dict.message:
- * test/data/valid-messages/dict-simple.message:
- * test/data/valid-messages/dict.message:
- * test/data/valid-messages/emptiness.message:

- * test/data/valid-messages/lots-of-arguments.message:
- * test/data/valid-messages/no-padding.message:
- * test/data/valid-messages/recursive-types.message:

Add missing NAME fields
Fix up dicts & arrays

- * test/data/invalid-messages/dict-with-nil-value.message:

Removed, this is not invalid anymore.

- * test/data/valid-messages/recursive-types.message:

Add new test for deeply recursive types.

2003-04-07 Havoc Pennington <hp@pobox.com>

- * bus/driver.c (bus_driver_handle_acquire_service): return an error if you try to acquire a service that starts with ':'

2003-04-07 Havoc Pennington <hp@redhat.com>

- * doc/dbus-specification.sgml: require that base service names start with ':' and that the base service is created/deleted as first and last things a connection does on the bus

- * bus/dispatch.c (check_existent_service_activation): lots more work on the activation test; it doesn't fully pass yet...

- * test/test-service.c (main): fix so we don't memleak the connection to the message bus
(filter_func): accept a message asking us to exit

2003-04-06 Havoc Pennington <hp@pobox.com>

- * qt/Makefile.am (dbusinclude_HEADERS): install dbus-qt.h, from Colin Walters

- * configure.in: fixes to Qt detection from Colin Walters

- * doc/Makefile.am: Only remove generated docbook dirs if they exist, from Colin Walters

- * dbus/dbus-bus.c: change how we set well-known connections to NULL, so that it works if a single connection is stored in two well-known array slots.

- * test/Makefile.am: remove a lot of stuff that isn't immediately useful, it's in CVS history if we want it.

- * test/test-service.c: use dbus-mainloop instead of that watch.[hc] crack

2003-04-06 Havoc Pennington <hp@pobox.com>

in

- * dbus/Makefile.am: split lists of sources into stuff that goes in the library, util functions that go in the lib and are also used elsewhere, and util functions that are used in tests/daemon but don't go in the lib.

- * dbus/dbus-mainloop.h, dbus/dbus-mainloop.c: move bus/loop.[hc] here so it can be used in test binaries also

2003-04-06 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-sysdeps.c (_dbus_become_daemon): write the pidfile here in the parent process, so we can return an error if it fails. Also, move some of the code into the child so the parent is less hosed if we fail midway through.

- * bus/bus.c (bus_context_new): move pidfile detection further up in the function, before we start overwriting sockets and such.

- * bus/messagebus.in: adjust this a bit, not sure if it will work.

- * configure.in: add --with-system-pid-file and --with-system-socket

2003-04-06 Colin Walters <walters@verbum.org>

- * configure.in (DBUS_SYSTEM_PID_FILE): New variable.

- * bus/system.conf.in: Declare a pidfile.

- * bus/bus.c (bus_context_new): Test for an existing pid file, and create one (if appropriate).

- * bus/config-parser.c (enum ElementType) [ELEMENT_PIDFILE]: New.
(struct BusConfigParser) [pidfile]: New.
(element_type_to_name, merge_included, start_busconfig_child)
(bus_config_parser_end_element, bus_config_parser_content):

Handle it.

- (bus_config_parser_unref): Free it.

- (bus_config_parser_get_pidfile): New function.

- * bus/config-parser.h (_dbus_write_pid_file): Prototype.

- * dbus/dbus-errors.h (DBUS_ERROR_PIDFILE_EXISTS): New error.

- * dbus/dbus-sysdeps.c (_dbus_write_pid_file): New function.

- * dbus/dbus-sysdeps.h: Prototype it.

2003-04-06 Havoc Pennington <hp@pobox.com>

- * bus/bus.c (bus_context_new): print the address in here, rather

than in main(), because we need to do it before forking the daemon

- * bus/dispatch.c (send_service_nonexistent_error): set the sender on the service nonexistent error

- * bus/driver.c (bus_driver_handle_acquire_service): set the sender on the AcquireService reply

- * test/data/valid-config-files/debug-allow-all.conf.in: Make test server also listen on a UNIX socket so services can connect to it.

2003-04-06 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-threads.c: Redo how the fake debug mutexes are done so it detects deadlocks and also we actually init threads when debugging.

2003-04-06 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-server-unix.c (_dbus_server_new_for_domain_socket): save the domain socket name, and unlink it when we disconnect the server. Means that at least when we exit normally, we won't leave a bunch of junk in /tmp

- * dbus/dbus-transport-unix.c (_dbus_transport_new_for_domain_socket): code cleanup (nicer memory management). (I was making a real change here but then didn't)

2003-04-06 Havoc Pennington <hp@pobox.com>

- * bus/bus.c (bus_context_new): fix wrong handling of server_data_slot_unref() in the error case.

- * dbus/dbus-internals.h (_dbus_assert): change so it passes "(condition) != 0" to _dbus_real_assert so that "_dbus_assert (pointer)" doesn't cause a warning

- * bus/main.c (main): accept --print-address option to print out the message bus address

- * dbus/dbus-sysdeps.c (_dbus_generate_random_ascii): export this

- * dbus/dbus-transport.c (_dbus_transport_open): special error for "tmpdir" option to unix: address on client side

- * dbus/dbus-server.c (dbus_server_listen): handle "tmpdir" option to unix: address

- * configure.in (TEST_SOCKET_DIR): locate a temporary directory

we can use to create sockets in the test suite.

* bus/main.c (signal_handler): on SIGTERM, exit the daemon cleanly. To be used for testing.

* dbus/dbus-spawn.c (babysit): use _dbus_set_signal_handler()

* dbus/dbus-sysdeps.c (_dbus_set_signal_handler): new

* dbus/dbus-server-debug-pipe.c (_dbus_transport_debug_pipe_new): handle trying to call this when there's no servers active

2003-04-05 Havoc Pennington <hp@pobox.com>

* NEWS: update

* configure.in: 0.8

2003-04-05 Havoc Pennington <hp@pobox.com>

* bus/bus.c (setup_server): fix this so dbus-daemon-1 doesn't crash on startup. Need to get "try starting the daemon" in the test suite I guess. ;-)

* dbus/dbus-server.h, dbus/dbus-server.c: remove the stuff that tracked the number of open connections; it's better done in application-specific code as you want it to span all servers etc.

2003-04-05 Havoc Pennington <hp@pobox.com>

* bus/Makefile.am (install-data-hook): add missing DESTDIR, patch from Colin Walters

2003-04-05 Havoc Pennington <hp@pobox.com>

* doc/config-file.txt (Elements): fix docs of <auth> to reflect reality; in fact multiple mechanisms are allowed.

* dbus/dbus-internals.c (_dbus_real_assert) (_dbus_real_assert_not_reached): move guts of _dbus_assert() and _dbus_assert_not_reached() into functions, so that they don't

show

up in basic block counts for test coverage, and don't use up as much disk space. Does mean slower execution speed though, so assumes --disable-asserts is the normal production case.

2003-04-05 Havoc Pennington <hp@pobox.com>

* test/Makefile.am (dist-hook): also dist *.in files

* NEWS: update

* configure.in: 0.7

2003-04-05 Havoc Pennington <hp@pobox.com>

* dbus/dbus-string.c: docs warning

* dbus/dbus-spawn.c: missing docs

* dbus/dbus-memory.c (struct ShutdownClosure): missing docs

2003-04-05 Havoc Pennington <hp@pobox.com>

* bus/loop.c (bus_loop_iterate): fix the timeout code, using magic from GLib

* dbus/dbus-spawn.c (_dbus_babysitter_unref): set sitter_pid to -1 once we've reaped the babysitter
(_dbus_babysitter_handle_watch): do as much work as we can, not just one go of it

* bus/activation.c: add code using DBusBabysitter so that we handle it when a service fails to start up properly.
(bus_activation_service_created): don't remove the activation entries as we go, just let them get removed when we free the pending activation. Unref reply messages after sending them.

2003-04-05 Havoc Pennington <hp@pobox.com>

* test/decode-gcov.c (main): print per-directory stats in the report

* Makefile.am (coverage-report.txt): don't include test/* in gcov stats

2003-04-05 Havoc Pennington <hp@pobox.com>

* Makefile.am (coverage-report.txt): add target "coverage-report.txt"

* test/decode-gcov.c: hack up a little program to suck data out of gcov files. Yes this is sort of silly.

* configure.in: define something in config.h and do an AM_CONDITIONAL when gcov is enabled

2003-04-04 Havoc Pennington <hp@redhat.com>

* dbus/dbus-spawn.c, dbus/dbus-spawn.h: Change dbus_spawn to return a "babysitter" object that is used to monitor the status of the spawned process and reap it when required.

* test/test-segfault.c, test/test-exit.c,
test/test-sleep-forever.c: binaries that do various lame things,
used in the test suite.

* dbus/dbus-sysdeps.c: kill _dbus_errno_to_string()

2003-04-03 Havoc Pennington <hp@pobox.com>

* dbus/dbus-spawn.c: Move dbus-spawn into a separate file
in preparation for modifying it, dbus-sysdeps is getting
a bit unmanageable.

2003-04-03 Havoc Pennington <hp@redhat.com>

* bus/loop.h, bus/loop.c: make the mainloop an object so we can
have multiple ones

* bus/*. [hc]: adapt to mainloop change

2003-04-03 Havoc Pennington <hp@redhat.com>

* bus/activation.c (load_directory): fix up memleaks
(bus_activation_entry_free): free the entry

* dbus/dbus-bus.c (dbus_bus_acquire_service): return an error if
we get one from the message bus; fix memleaks.

* dbus/dbus-message.c (dbus_set_error_from_message): new function

2003-04-03 Havoc Pennington <hp@pobox.com>

* bus/config-parser.c (bus_config_parser_unref): free
list of mechanisms, bug discovered by test suite enhancements
(putting system.conf and session.conf into suite)

* test/Makefile.am, test/test-service.c: add placeholder for a
test service that we'll activate as part of test suite. Doesn't
do anything yet.

* dbus/dbus-sysdeps.c (_dbus_setenv): support unsetenv by
setting NULL value, and use system malloc not dbus_malloc()
when we have unavoidable memleakage.

* dbus/dbus-bus.c (dbus_bus_get): fix bug where bus type of 0
didn't work, and support DBUS_BUS_ACTIVATION.

* bus/activation.c (child_setup): pass our well-known bus type to
the child

* bus/config-parser.c: support <type> to specify well-known type

* doc/dbus-specification.sgml: document the env variables to locate well-known buses and find service activator

2003-04-02 Havoc Pennington <hp@redhat.com>

* test/Makefile.am (all-local): add a rule to copy tests to builddir, so we can have generated tests. Use this to remove the silly hack for testing system.conf and session.conf. Will use this shortly to generate .service files pointing to test binaries.

2003-04-02 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (set_length): fix a bug - we allocated max of current alloc and needed new length, not max of the doubled allocation and needed new length. Also, when building tests, don't do the double-allocation stuff, just realloc every time.

2003-04-02 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_file_get_contents): include filenames in error messages
(_dbus_string_get_dirname): new
(_dbus_sysdeps_test): new
(_dbus_directory_open): include dirnames in error messages

* bus/config-parser.c: interpret <include> and <includedir> and <servicedir> relative to config file location if the given filename is not absolute.

* dbus/dbus-string.c (_dbus_string_find_byte_backward): new

2003-04-02 Havoc Pennington <hp@redhat.com>

* bus/connection.c (bus_transaction_send_error_reply): set sender service for the error, and unref the reply on success

* bus/activation.c: convert to use BusTransaction so OOM can be handled correctly
(bus_activation_service_created): set sender of the message

2003-04-01 Havoc Pennington <hp@redhat.com>

* bus/config-parser.c, bus/bus.c: implement <servicedir> and <includedir> (at least mostly)

* dbus/dbus-sysdeps.c (_dbus_change_identity): set the group ID first, then the user ID

2003-04-01 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-server.c (dbus_server_set_auth_mechanisms): new function
- * dbus/dbus-auth.c (_dbus_auth_set_mechanisms): new
- * dbus/dbus-internals.c (_dbus_dup_string_array): new function
- * dbus/dbus-sysdeps.c (_dbus_listen_unix_socket): chmod the socket 0777, and unlink any existing socket.
- * bus/bus.c (bus_context_new): change our UID/GID and fork if the configuration file so specifies; set up auth mechanism restrictions
- * bus/config-parser.c (bus_config_parser_content): add support for <fork> option and fill in code for <auth>
- * bus/system.conf.in: add <fork/> to default configuration, and limit auth mechanisms to EXTERNAL
- * doc/config-file.txt (Elements): add <fork>
- * dbus/dbus-sysdeps.c (_dbus_become_daemon): new function
(_dbus_change_identity): new function

2003-03-31 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps.c (_dbus_connect_unix_socket)
(_dbus_listen_unix_socket): fix off-by-one error in null termination spotted by Nalin

2003-03-31 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-keyring.c (_dbus_keyring_new_homedir): allow setting DBUS_TEST_HOMEDIR when tests are enabled, so we can test without having a real home directory available.

2003-03-31 Havoc Pennington <hp@redhat.com>

- * bus/Makefile.am (install-data-hook): create /var/run/dbus
- * bus/messagebus.in: add init script for Red Hat /etc/init.d
- * configure.in: add support for specifying a style of init script to install

2003-03-31 Havoc Pennington <hp@redhat.com>

- Fix some annoying DbusString API and fix all affected code.
- * dbus/dbus-string.c (_dbus_string_init): get rid of annoying

max_length argument
(_dbus_string_get_data): change to return string instead of using
an out param
(_dbus_string_get_const_data): ditto
(_dbus_string_get_data_len): ditto
(_dbus_string_get_const_data_len): ditto

2003-03-31 Havoc Pennington <hp@redhat.com>

* bus/main.c (main): fix up the command line arguments to be
nicer

2003-03-31 Havoc Pennington <hp@redhat.com>

* dbus/Makefile.am (INCLUDES): use EXPANDED_LOCALSTATEDIR to
define DBUS_SYSTEM_BUS_PATH as we want to compile in the same
final location that lands in the config file

* bus/config-loader-expat.c (bus_config_load): fix type of
XML_Parser variable

* doc/TODO: remove TODO item for dbus_bus_get()

* dbus/dbus-bus.c (bus_data_free): add missing lock/unlock

2003-03-31 Havoc Pennington <hp@pobox.com>

* dbus/dbus-transport-unix.c
(_dbus_transport_new_for_domain_socket)
(_dbus_transport_new_for_tcp_socket): these didn't need the
"server"
argument since they are always client side

* dbus/dbus-server.c (dbus_server_get_address): new function

* bus/main.c (main): take the configuration file as an argument.

* test/data/valid-config-files/debug-allow-all.conf: new file to
use with dispatch.c tests for example

* bus/test-main.c (main): require test data dir

* bus/bus.c (bus_context_new): change this to take a
configuration file name as argument

* doc/config-file.txt (Elements): add <servicedir>

* bus/system.conf, bus/session.conf: new files

* dbus/dbus-bus.c (dbus_bus_get): look for system bus on
well-known socket if none set

* configure.in: create system.conf and session.conf

2003-03-30 Havoc Pennington <hp@pobox.com>

* bus/config-parser.c: hacking

* dbus/dbus-memory.c: don't use DBusList for the list of stuff to shut down, since it could cause weirdness with the DBusList lock

* dbus/dbus-list.c (_dbus_list_test): add tests for the link-oriented stack routines
(alloc_link): free the mempool if the first alloc from it fails

* dbus/dbus-mempool.c (struct DBusMemBlock): fix alignment issue

* dbus/dbus-string.c (UNICODE_VALID): sync new version of this from GLib
(_dbus_string_skip_white): new

* doc/config-file.txt (Elements): add <includedir>

2003-03-28 Havoc Pennington <hp@pobox.com>

* dbus/dbus-string.c (_dbus_string_copy_data_len)
(_dbus_string_copy_data): new functions

2003-03-28 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-bus.c: (bus_data_free), (dbus_bus_get):
* dbus/dbus-bus.h:
Add dbus_bus_get.

* dbus/dbus-memory.c:
Fix a doc comment.

2003-03-28 Havoc Pennington <hp@pobox.com>

* bus/test.c (bus_test_flush_bus): remove the sleep from here, I think it may have just been superstition. Not sure.

* dbus/dbus-string.c (_dbus_string_base64_decode): catch some OOM failures that were not being handled.

* dbus/dbus-auth.c (process_auth): fix a memleak in OOM handling

* dbus/dbus-memory.c: add ability to set number of mallocs in a row that will fail on out-of-memory.

* dbus/dbus-internals.c (_dbus_test_oom_handling): convenience function for testing out-of-memory handling.

* bus/config-loader-expat.c (memsuite): don't wrap the dbus allocation functions, they do map exactly to the expat ones.

2003-03-27 Havoc Pennington <hp@redhat.com>

* bus/config-loader-libxml.c (bus_config_load): add another error check

2003-03-26 Anders Carlsson <andersca@codefactory.se>

* doc/TODO:
Add note about automatic service activation.

* doc/dbus-specification.sgml:
Rename the specification and clarify a few things.

2003-03-26 Anders Carlsson <andersca@codefactory.se>

* Doxyfile.in:
* dbus/dbus-address.c:
* dbus/dbus-dict.c:
* dbus/dbus-marshal.c:
* dbus/dbus-server-debug-pipe.c:
* dbus/dbus-transport-unix.c:
Fix documentation warnings.

2003-03-26 Havoc Pennington <hp@pobox.com>

* bus/test-main.c, dbus/dbus-test.c (main): check memleaks after every test so it's quick and easy to see which leaked, and so we test multiple dbus_shutdown() calls

* configure.in: change configure.in XML stuff to also support expat

* config-loader-libxml.c: some hacking

* config-loader-expat.c: some hacking

* config-parser.c: some hacking, plus tests

2003-03-25 Havoc Pennington <hp@redhat.com>

* throughout - add more _DBUS_ASSERT_ERROR_IS_CLEAR

* configure.in: add --with-xml option to specify XML library, right now only libxml is supported.

* bus/config-loader-libxml.c, config-parser.c: sync some minor nonworking code between home and work, still just stubs

2003-03-24 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps.c (_dbus_set_fd_nonblocking): move to this file

* dbus/dbus-errors.c (dbus_set_error, dbus_set_error_const):
allow
NULL argument for "message" if the error is a well-known one,
fill in a generic message in this case.

* dbus/dbus-errors.h (DBusResultCode): Kill DBusResultCode in
favor of DBusError

* bus/test.c (bus_test_flush_bus): add

* bus/policy.c (bus_policy_test): test code stub

2003-03-24 Havoc Pennington <hp@pobox.com>

* bus/connection.c (bus_connections_setup_connection): set up
the "can this user connect" function, but it always returns
TRUE until we have a config file parser so we can have a config
file that allows connections.

2003-03-23 Havoc Pennington <hp@pobox.com>

* dbus/dbus-threads.c (dbus_mutex_new, dbus_condvar_new): with
DBUS_BUILD_TESTS, actually alloc/free a block of memory for
the mutex, so we can check for proper memory management
and OOM handling.

* dbus/dbus-dataslot.c: remove the mutex from
DBusDataSlotAllocator and lock it manually when using it,
to simplify fitting it into the global slots framework.

* dbus/dbus-threads.c (init_static_locks): rework how we're
handling global locks so they are easily shut down.

* bus/policy.c (bus_policy_append_rule): fix

* bus/test-main.c (main): check for memleaks

* dbus/dbus-test.c (dbus_internal_do_not_use_run_tests): make
test suite check for memleaks

* dbus/dbus-memory.c: add support in test mode for tracking
number of outstanding blocks

2003-03-23 Havoc Pennington <hp@pobox.com>

* bus/policy.c, bus/bus.c, bus/connection.c: implement allow/deny
policies code

- * dbus/dbus-hash.h: add ULONG hash keys
- * dbus/dbus-sysdeps.c (_dbus_get_groups): new
(_dbus_get_group_id): new function

2003-03-20 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-connection.c
(dbus_connection_set_unix_user_function):
new function
(dbus_connection_get_unix_user): new function

2003-03-20 Havoc Pennington <hp@pobox.com>

- * bus/connection.c (bus_connection_send_oom_error): assert that
message has a sender
(connection_execute_transaction): ditto
(bus_connection_preallocate_oom_error): fix to set the sender,
and
set recipient to the destination service, not the bus driver
- * bus/policy.c: hacking
- * dbus/dbus-message.c (dbus_message_service_is): new function
(dbus_message_sender_is): new

2003-03-19 Havoc Pennington <hp@redhat.com>

- * bus/policy.c: start sketching code for policy restrictions on
what connections can do.

2003-03-18 Havoc Pennington <hp@redhat.com>

- * doc/TODO: some notes on high-level todo items. Little nitpick
stuff is all in @todo, so no need to add it here.
- * doc/config-file.txt: some notes on how config file might look

2003-03-18 Anders Carlsson <andersca@codefactory.se>

- * configure.in: 0.6
- * NEWS: Update.

2003-03-17 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-internals.h: add gcc attributes so that
our printf-style functions warn on bad arguments to
format
- * dbus/dbus-sysdeps.c (_dbus_connect_tcp_socket): fix printf
format bug

* dbus/dbus-message.c (_dbus_message_loader_queue_messages): fix printf format bug

2003-03-17 Havoc Pennington <hp@redhat.com>

* bus/test-main.c (main): make it print something as it runs so make check doesn't look stuck

* doc/negotiation.txt, doc/dbus-sasl-profile.txt: remove from CVS, now obsolete

2003-03-17 Anders Carlsson <andersca@codefactory.se>

* bus/dispatch.c: (bus_dispatch):
Refetch the service name since it may have been reallocated when dbus_message_set_sender was called.

* dbus/dbus-sysdeps.c: (_dbus_accept):
Add address and address length variables and use them to stop valgrind from complaining.

2003-03-17 Havoc Pennington <hp@pobox.com>

All tests pass, no memleaks, no valgrind complaints.

* bus/test.c: refcount handler_slot

* bus/connection.c (bus_connections_new): refcount connection_data_slot

* dbus/dbus-auth-script.c (_dbus_auth_script_run): delete unused bytes so that auth scripts pass.

* bus/dispatch.c: init message_handler_slot so it gets allocated properly

* bus/dispatch.c (message_handler_slot_ref): fix memleak

* dbus/dbus-server-debug-pipe.c (_dbus_server_debug_pipe_new): dealloc server_pipe_hash when no longer used for benefit of leak checking

* dbus/dbus-auth.c (process_command): memleak fix

* bus/dispatch.c (check_hello_message): memleak fix

2003-03-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-bus.c (ensure_bus_data): fix double-unref of the data slot

2003-03-17 Anders Carlsson <andersca@codefactory.se>

- * bus/activation.c (bus_activation_activate_service): Append the pending activation entry to the list of pending activations.

2003-03-16 Havoc Pennington <hp@pobox.com>

- * bus/dispatch.c (bus_dispatch_test): remove double-unrefs of connections

- * dbus/dbus-address.c (create_entry): fix OOM handling when failing to alloc entry->method

2003-03-16 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-watch.c (_dbus_watch_new): handle failure to malloc the watch

- * dbus/dbus-server-debug-pipe.c (_dbus_transport_debug_pipe_new): add some missing dbus_set_result

- * bus/dispatch.c (bus_dispatch_add_connection): handle failure to alloc the DBusMessageHandler

- * dbus/dbus-transport.c (_dbus_transport_disconnect): don't ref the transport here, since we call this from the finalizer; it resulted in a double-finalize.

- * dbus/dbus-transport.c (_dbus_transport_disconnect): fix a bug where we tried to use transport->connection that was NULL, happened when transport was disconnected early on due to OOM

- * bus/*.c: adapt to handle OOM for watches/timeouts

- * dbus/dbus-transport-unix.c: port to handle OOM during watch handling

- * dbus/dbus-auth.c (_dbus_auth_get_unused_bytes): return a reference to unused bytes instead of a copy

- * dbus/dbus-server.c (dbus_server_handle_watch): return FALSE for out of memory

- * dbus/dbus-connection.c (dbus_connection_handle_watch): return FALSE on OOM

- * dbus/dbus-timeout.c (dbus_timeout_handle): return FALSE for out of memory

2003-03-16 Anders Carlsson <andersca@codefactory.se>

- * doc/dbus-specification.sgml:

Document reply message for ActivateService.

2003-03-16 Anders Carlsson <andersca@codefactory.se>

- * bus/activation.c: (bus_pending_activation_entry_free), (bus_pending_activation_free), (bus_activation_new), (bus_activation_unref), (bus_activation_service_created), (bus_activation_activate_service):
- * bus/activation.h:
- * bus/bus.c: (bus_context_new):
- * bus/desktop-file.c: (new_section):
- * bus/driver.c: (bus_driver_send_service_deleted), (bus_driver_handle_activate_service):
- * bus/services.c: (bus_registry_new), (bus_registry_ensure):
- * bus/services.h:
- * dbus/dbus-connection.c: (dbus_connection_send_with_reply_and_block):
- * dbus/dbus-message.c: (dbus_message_append_args_valist):
- * dbus/dbus-protocol.h:

Make activation work better. Now pending activations will be queued

and the daemon won't try to activate services that are already registered.

2003-03-16 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-bus.c (ensure_bus_data): handle failure to set connection data
- * dbus/dbus-memory.c (_dbus_initialize_malloc_debug): support DBUS_MALLOC_BACKTRACES to print trace when failing an alloc

2003-03-16 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-string.c (_dbus_string_validate_utf8): oops, unbreak this. always run the test suite before commit...

- * bus/*: adapt to DBusConnection API changes

- * glib/dbus-gmain.c: adapt to DBusConnection API changes, requires renaming stuff to avoid dbus_connection_dispatch name conflict.

- * dbus/dbus-transport.c (_dbus_transport_queue_messages): new function

- * dbus/dbus-message.c (_dbus_message_loader_queue_messages): separate from _dbus_message_loader_return_buffer()

- * dbus/dbus-connection.c (dbus_connection_get_n_messages): remove this, because it's now always broken to use; the number of messages in queue vs. the number still buffered by the message

loader is undefined/meaningless. Should use
dbus_connection_get_dispatch_state().
(dbus_connection_dispatch): rename from
dbus_connection_dispatch_message

2003-03-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-string.c (_dbus_string_validate_utf8): copy in a real
implementation

2003-03-16 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-connection.c:
(dbus_connection_send_with_reply_and_block):
Decrease connection->n_incoming when removing an entry
from the list.
* dbus/dbus-dict.c: (dbus_dict_entry_free),
(dbus_dict_set_boolean_array), (dbus_dict_set_int32_array),
(dbus_dict_set_uint32_array), (dbus_dict_set_double_array),
(dbus_dict_set_byte_array), (dbus_dict_set_string_array),
(dbus_dict_get_boolean_array), (dbus_dict_get_double_array),
(dbus_dict_get_byte_array):
Handle NULL arrays and strings. Also add support for byte arrays.

* dbus/dbus-marshal.c: (_dbus_marshall_byte_array),
(_dbus_marshall_dict), (_dbus_demarshal_byte_array),
(_dbus_demarshal_int32_array), (_dbus_demarshal_uint32_array),
(_dbus_demarshal_double_array), (_dbus_demarshal_string_array),
(_dbus_demarshal_dict), (demarshal_and_validate_len),
(_dbus_marshall_validate_arg), (_dbus_marshall_test):
* dbus/dbus-marshal.h:
Add support for marshalling and demarshalling empty arrays and
strings.

* dbus/dbus-message.c: (dbus_message_append_args_valist),
(dbus_message_append_string_array),
(dbus_message_iter_get_boolean),
(dbus_message_iter_get_boolean_array),
(dbus_message_iter_get_int32_array),
(dbus_message_iter_get_uint32_array),
(dbus_message_iter_get_double_array),
(dbus_message_iter_get_byte_array),
(dbus_message_iter_get_string_array),
(dbus_message_iter_get_dict),
(check_message_handling):
Add support for getting empty arrays and dicts.

* dbus/dbus-string.c: (_dbus_string_validate_utf8):
Don't do any validation at all for now, that's better than just
checking
for ASCII.

* test/data/valid-messages/emptiness.message:
New test message with lots of empty arrays.

2003-03-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c
(_dbus_connection_queue_received_message_link): new function that
can't fail due to OOM

* dbus/dbus-message.c (_dbus_message_loader_pop_message_link):
new function pops a message together with a list link
containing it.

* dbus/dbus-transport-unix.c (queue_messages): use new link-based
message queuing functions to avoid needing to alloc memory

2003-03-16 Havoc Pennington <hp@pobox.com>

Oops - test code was only testing failure of around 30 of the
mallocs in the test path, but it turns out there are 500+
mallocs. I believe this was due to misguided linking setup such
that there was one copy of dbus_malloc etc. in the daemon and one
in the shared lib, and only daemon mallocs were tested. In any
case, the test case now tests all 500+ mallocs, and doesn't pass
yet, though there are lots of fixes in this patch.

* dbus/dbus-connection.c (dbus_connection_dispatch_message): fix
this so that it doesn't need to allocate memory, since it
has no way of indicating failure due to OOM (and would be
annoying if it did).

* dbus/dbus-list.c (_dbus_list_pop_first_link): new function

* bus/Makefile.am: rearrange to create two self-contained
libraries, to avoid having libraries with overlapping symbols.
that was resulting in weirdness, e.g. I'm pretty sure there
were two copies of global static variables.

* dbus/dbus-internals.c: move the malloc debug stuff to
dbus-memory.c

* dbus/dbus-list.c (free_link): free list mempool if it becomes
empty.

* dbus/dbus-memory.c (_dbus_disable_mem_pools): new function

* dbus/dbus-address.c (dbus_parse_address): free list nodes
on failure.

* bus/dispatch.c (bus_dispatch_add_connection): free
message_handler_slot when no longer using it, so
memory leak checkers are happy for the test suite.

* dbus/dbus-server-debug-pipe.c (debug_finalize): free server name

* bus/bus.c (new_connection_callback): disconnect in here if bus_connections_setup_connection fails.

* bus/connection.c (bus_connections_unref): fix to free the connections
(bus_connections_setup_connection): if this fails, don't disconnect the connection, just be sure there are no side effects.

* dbus/dbus-string.c (undo_alignment): unbreak this

* dbus/dbus-auth.c (_dbus_auth_unref): free some stuff we were leaking
(_dbus_auth_new): fix the order in which we free strings on OOM failure

* bus/connection.c (bus_connection_disconnected): fix to not send ServiceDeleted multiple times in case of memory allocation failure

* dbus/dbus-bus.c (dbus_bus_get_base_service): new function to get the base service name
(dbus_bus_register_client): don't return base service name, instead store it on the DBusConnection and have an accessor function for it.
(dbus_bus_register_client): rename dbus_bus_register()

* bus/dispatch.c (check_hello_message): verify that other connections on the bus also got the correct results, not just the one sending hello

2003-03-15 Havoc Pennington <hp@pobox.com>

Make it pass the Hello handling test including all OOM codepaths. Now to do other messages...

* bus/services.c (bus_service_remove_owner): fix crash when removing owner from an empty list of owners
(bus_registry_ensure): don't leave service in the list of a connection's owned services if we fail to put the service in the hash table.

* bus/connection.c (bus_connection_preallocate_oom_error): set error flag on the OOM error.

* dbus/dbus-connection.c (_dbus_connection_new_for_transport): handle _dbus_transport_set_connection failure

* dbus/dbus-transport-unix.c (`_dbus_transport_new_for_fd`): modify to create watches up front and simply enable/disable them as needed.

(`unix_connection_set`): this can now fail on OOM

* dbus/dbus-timeout.c, dbus/dbus-watch.c: add concept of enabling/disabling a watch or timeout.

* bus/loop.c (`bus_loop_iterate`): don't touch disabled watches/timeouts

* glib/dbus-gmain.c: adapt to enable/disable watches and timeouts

2003-03-15 Havoc Pennington <hp@pobox.com>

* bus/dispatch.c (`bus_dispatch_test`): OK, now finally actually write useful test code, after all that futzing around ;-)

Test does not yet pass because we can't handle OOM in `_dbus_transport_messages_pending` (basically, `dbus_connection_preallocate_send()` does not prealloc the write watch). To fix this, I think we need to add new stuff to `set_watch_functions`, namely a `SetEnabled` function so we can alloc the watch earlier, then enable it later.

* dbus/Makefile.am (`libdbus_convenience_la_SOURCES`): move `dbus-memory.c` to the convenience lib

* bus/test.c: rename some static functions to keep them clearly distinct from stuff in `connection.c`. Handle client disconnection.

2003-03-14 Havoc Pennington <hp@pobox.com>

* bus/dispatch.c (`bus_dispatch_test`): do test using debug-pipe transport, tests more of the real codepath. Set up clients with `bus_setup_debug_client`.

* bus/test.c (`bus_setup_debug_client`): function to set up debug "clients" on the main loop

* dbus/dbus-transport.c (`_dbus_transport_open`): add debug-pipe support

* dbus/dbus-server.c (`dbus_server_listen`): add debug-pipe server type

* dbus/dbus-server-debug.c: support a debug server based on pipes

* dbus/dbus-sysdeps.c (`_dbus_full_duplex_pipe`): new function
(`_dbus_close`): new function

* `configure.in`: check for `socketpair`

2003-03-14 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-memory.c: add a "detect buffer overwrites on free" cheesy hack
- * dbus/dbus-transport-debug.c: rework this a good bit to be less complicated. hopefully still works.
- * dbus/dbus-server-debug.c (handle_new_client): remove timeout manually
- * glib/dbus-gmain.c (timeout_handler): don't remove timeout after running it
- * dbus/dbus-message.c (dbus_message_copy): rename from dbus_message_new_from_message, fix it up to copy all the message fields, add test case
- * bus/dispatch.c (bus_dispatch_test): add some more test code, not quite passing yet

2003-03-14 Havoc Pennington <hp@pobox.com>

- * bus/loop.c (bus_loop_iterate): add this so we can "run loop until no work remains" in test code. (the large diff here is just code movement, no actual changes)
- * dbus/dbus-server-debug.c (DEFAULT_INTERVAL): change interval to 1, no point waiting around for test code. (_dbus_server_debug_accept_transport): unref the timeout after adding it (right?)
- * dbus/dbus-transport-debug.c (DEFAULT_INTERVAL): ditto

2003-03-13 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-timeout.c (_dbus_timeout_list_set_functions): handle out of memory
- * dbus/dbus-watch.c (_dbus_watch_list_set_functions): handle out of memory
- * dbus/dbus-connection.h: Make AddWatchFunction and AddTimeoutFunction return a bool so they can fail on out-of-memory
- * bus/bus.c (bus_context_new): set up timeout handlers
- * bus/connection.c (bus_connections_setup_connection): set up timeout handlers

- * glib/dbus-gmain.c: adapt to the fact that set_functions stuff can fail
- * bus/bus.c (bus_context_new): adapt to changes
- * bus/connection.c: adapt to changes
- * test/watch.c: adapt to DbusWatch changes
- * bus/dispatch.c (bus_dispatch_test): started adding this but didn't finish

2003-03-14 Anders Carlsson <andersca@codefactory.se>

- * bus/dispatch.c (send_service_nonexistent_error): Fix typo.

2003-03-13 Havoc Pennington <hp@pobox.com>

- * bus/test.c, bus/test.h, bus/Makefile.am, bus/test-main.c: set up a test framework as for the library

2003-03-12 Havoc Pennington <hp@pobox.com>

Throughout: purge global variables, introduce BusActivation, BusConnections, BusRegistry, etc. objects instead.

- * bus/bus.h, bus/bus.c: introduce BusContext as a global message bus object
- * test/Makefile.am (TEST_BINARIES): disable bus-test for now, going to redo this a bit differently I think

2003-03-12 Havoc Pennington <hp@redhat.com>

Mega-patch that gets the message bus daemon initially handling out-of-memory. Work still needed. Also lots of random moving stuff to DBusError instead of ResultCode.

- * dbus/dbus-list.c (_dbus_list_length_is_one): new function
 - * dbus/dbus-connection.c (dbus_connection_send_with_reply_and_block): use DBusError
 - * dbus/dbus-bus.c: adapt to API changes, make it use DBusError
- not
DBusResultCode
- * dbus/dbus-connection.c (dbus_connection_send): drop the result code here, as the only failure possible is OOM.
 - * bus/connection.c (bus_connection_disconnect): rename bus_connection_disconnected as it's a notification only

```

* bus/driver.c (bus_driver_handle_acquire_service): don't free
"name" on get_args failure, should be done by get_args;
don't disconnect client for bad args, just return an error.
(bus_driver_handle_service_exists): ditto

* bus/services.c (bus_services_list): NULL-terminate returned
array

* bus/driver.c (bus_driver_send_service_lost)
(bus_driver_send_service_acquired): send messages from driver to
a
specific client to the client's unique name, not to the broadcast
service.

* dbus/dbus-message.c (decode_header_data): reject messages that
contain no name field
(_dbus_message_get_client_serial): rename to
dbus_message_get_serial and make public
(_dbus_message_set_serial): rename from set_client_serial
(_dbus_message_set_reply_serial): make public
(_dbus_message_get_reply_serial): make public

* bus/connection.c (bus_connection_foreach): allow stopping
iteration by returning FALSE from foreach function.

* dbus/dbus-connection.c (dbus_connection_send_preallocated)
(dbus_connection_free_preallocated_send)
(dbus_connection_preallocate_send): new API for sending a message
without possibility of malloc failure.
(dbus_connection_send_message): rename to just
dbus_connection_send (and same for whole function family)

* dbus/dbus-errors.c (dbus_error_free): make this reinit the
error

* dbus/dbus-sysdeps.c (_dbus_exit): new function

* bus/activation.c: handle/return errors

* dbus/dbus-errors.h: add more DBUS_ERROR #define

* dbus/dbus-sysdeps.c (_dbus_directory_open)
(_dbus_file_get_contents)
(_dbus_directory_get_next_file): use DBusError instead of
DBusResultCode
(_dbus_result_from_errno): move to this file

```

2003-03-10 Anders Carlsson <andersca@codefactory.se>

```

* dbus/dbus-marshal.c:
(_dbus_marshal_set_string):

```

Take a length argument so we can marshal the correct string length.

```
(_dbus_marshal_dict), (_dbus_demarshal_dict),
(_dbus_marshal_get_arg_end_pos), (_dbus_marshal_validate_arg),
(_dbus_marshal_test):
* dbus/dbus-marshal.h:
Add support for marshalling and demarshalling dicts.

* dbus/dbus-message-builder.c: (_dbus_message_data_load):
Add support for TYPE_DICT.

* dbus/dbus-message.c: (set_string_field):
Adjust header padding.

(dbus_message_append_args_valist), (dbus_message_append_dict),
(dbus_message_get_args_valist), (dbus_message_iter_get_arg_type),
(dbus_message_iter_get_dict),
(_dbus_message_loader_return_buffer),
(check_message_handling), (check_have_valid_message):
* dbus/dbus-message.h:
Add functions for setting and getting dicts.

* dbus/dbus-protocol.h:
Add DBUS_TYPE_DICT.

* dbus/dbus.h:
Add dbus-dict.h

* doc/dbus-specification.sgml:
Add information about how dicts are marshalled.

* test/data/invalid-messages/dict-with-nil-value.message:
* test/data/invalid-messages/too-short-dict.message:
* test/data/valid-messages/dict-simple.message:
* test/data/valid-messages/dict.message:
Add sample messages containing dicts.
```

2003-03-08 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-dict.h: Add DBUS_END_DECLS.
```

2003-03-07 Anders Carlsson <andersca@codefactory.se>

```
* dbus/Makefile.am:
* dbus/dbus-dict.c: (dbus_dict_entry_free), (dbus_dict_new),
(dbus_dict_get_keys), (insert_entry), (dbus_dict_set_boolean),
(dbus_dict_set_int32), (dbus_dict_set_uint32),
(dbus_dict_set_double), (dbus_dict_set_string),
(dbus_dict_set_boolean_array), (dbus_dict_set_int32_array),
(dbus_dict_set_uint32_array), (dbus_dict_set_double_array),
(dbus_dict_set_string_array), (_dbus_dict_test):
```

* dbus/dbus-dict.h:
Fix according to comments from Havoc.

2003-03-06 Michael Meeks <michael@server.home>

* configure.in: if we don't have kde-config, disable have_qt.

2003-03-07 Anders Carlsson <andersca@codefactory.se>

* dbus/Makefile.am:
Add dbus-dict.[ch]

* dbus/dbus-dict.c: (dbus_dict_entry_free), (dbus_dict_new),
(dbus_dict_ref), (dbus_dict_unref), (dbus_dict_contains),
(dbus_dict_remove), (dbus_dict_get_value_type),
(dbus_dict_get_keys), (dbus_dict_put_boolean),
(dbus_dict_put_int32), (dbus_dict_put_uint32),
(dbus_dict_put_double), (dbus_dict_put_string),
(dbus_dict_put_boolean_array), (dbus_dict_put_int32_array),
(dbus_dict_put_uint32_array), (dbus_dict_put_double_array),
(dbus_dict_put_string_array), (dbus_dict_get_boolean),
(dbus_dict_get_int32), (dbus_dict_get_uint32),
(dbus_dict_get_double), (dbus_dict_get_string),
(dbus_dict_get_boolean_array), (dbus_dict_get_int32_array),
(dbus_dict_get_uint32_array), (dbus_dict_get_double_array),
(dbus_dict_get_string_array), (_dbus_dict_test):

* dbus/dbus-dict.h:
Add DbusDict implementation

* dbus/dbus-test.c: (dbus_internal_do_not_use_run_tests):
* dbus/dbus-test.h:
Add _dbus_dict_test

2003-03-04 Havoc Pennington <hp@pobox.com>

* test/data/auth/*: adapt to changes

* dbus/dbus-auth-script.c (_dbus_auth_script_run): add
USERID_BASE64 and change USERNAME_BASE64 to put in username not
userid

* dbus/dbus-keyring.c (_dbus_keyring_validate_context): prevent
more stuff from being in a context name, to make the protocol
simpler to deal with

* dbus/dbus-errors.c (dbus_error_has_name): new function
(dbus_error_is_set): new function

* dbus/dbus-auth.c: replace DBUS_STUPID_TEST_MECH auth
with DBUS_COOKIE_SHA1, implement DBUS_COOKIE_SHA1

* dbus/dbus-connection.c (dbus_connection_flush): also read

messages during a flush operation

* dbus/Makefile.am: remove dbus-md5 since it isn't currently used.

2003-03-05 Anders Carlsson <andersca@codefactory.se>

* configure.in: Check for gethostbyname on Solaris.

* dbus/dbus-transport.c: (_dbus_transport_open):
Remove duplicate "tcp" entry.

* doc/dbus-specification.sgml:
Clarify some things.

2003-03-05 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-auth.c: (send_rejected), (process_test_subdir):

* dbus/dbus-keyring.c: (_dbus_keyring_new_homedir),
(_dbus_keyring_test):

* dbus/dbus-md5.c: (_dbus_md5_compute):

* dbus/dbus-sha.c: (_dbus_sha_compute):

Plug memory leaks.

2003-03-05 Anders Carlsson <andersca@codefactory.se>

* README: Add some things.

2003-03-04 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c (dbus_message_append_args_valist): Add a
break;

after case DBUS_TYPE_BOOLEAN.

2003-03-02 Havoc Pennington <hp@pobox.com>

* test/break-loader.c (randomly_set_extreme_ints): add test that
sets really huge and small integers

* dbus/dbus-marshal.c (_dbus_marshal_validate_arg): add check
that length of boolean array fits in the string, and that
string has room for boolean value in single-bool case.

* dbus/dbus-message-builder.c (_dbus_message_data_load): add
optional value to "ALIGN" command which is what to fill the
alignment with.

* test/data/valid-messages/no-padding.message: add regression
test for the message padding problem

2003-03-02 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (decode_header_data): fix to always init message_padding, from Benjamin Dauvergne

2003-03-02 Havoc Pennington <hp@pobox.com>

* configure.in: 0.5

* NEWS: Update.

2003-03-01 Joe Shaw <joe@assbarn.com>

* configure.in: Check for "struct cmsgcred" and try to access its members for BSD-like unices.

* dbus/dbus-sysdeps.c (read_credentials_byte): Fold this back into

_dbus_read_credentials_unix_socket().

(_dbus_read_credentials_unix_socket): Use recvmsg() instead of read() for reading the credential byte off the unix socket. Use struct cmsgcred on systems that support it.

2003-02-27 Alexander Larsson <alexl@redhat.com>

* glib/Makefile.am:

* configure.in:

Make gthreads-2.0 dependency optional. Don't build thread test if its not found.

2003-02-27 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c

(dbus_connection_send_message_with_reply_and_block): fix doh! doh! doh! bug that resulted in never removing a reply from the queue, no wonder we called get_reply_serial so much ;-)

* dbus/dbus-message.c (struct DBusMessage): cache reply serial and client serial instead of demarshaling them every time

2003-02-27 Havoc Pennington <hp@pobox.com>

* dbus/dbus-marshal.c (_dbus_demarshal_int32): rewrite to be much more inlined, using dbus-string-private.h, speeds things up substantially

* dbus/dbus-string.c (_dbus_string_free): apply align offset when freeing the string

(_dbus_string_steal_data): fix for align offset

(undo_alignment): new function

2003-02-26 Havoc Pennington <hp@redhat.com>

All kinds of audit fixes from Owen, plus initial attempt to

handle unaligned memory returned from malloc.

- * dbus/dbus-string.c (`_dbus_string_init`): clamp max length to leave room for `align_offset` and nul byte
- (`fixup_alignment`): function to track an `align_offset` and ensure `real->str` is aligned
- (`DBUS_GENERIC_STRING_PREAMBLE`): len must be less than allocated, to allow a nul byte plus align offset
- (`_dbus_string_lock`): fix overflow issue
- (`_dbus_string_init_const_len`): add assertions on sanity of len, assign allocated to be `ALLOCATION_PADDING` larger than len
- (`set_length`): fixup the overflow handling
- (`_dbus_string_get_data_len`): fix overflow in assertion
- (`open_gap`): detect overflow in size of gap to be opened
- (`_dbus_string_lengthen`): add overflow check
- (`_dbus_string_align_length`): fix overflow with `_DBUS_ALIGN_VALUE`
- (`_dbus_string_append`): add overflow check
- (`_dbus_string_append_unichar`): overflow
- (`_dbus_string_delete`): fix overflow in assertion
- (`_dbus_string_copy_len`): overflow in assertion
- (`_dbus_string_replace_len`): overflows in assertions
- (`_dbus_string_find`): change to implement in terms of `_dbus_string_find_to`
- (`_dbus_string_find_to`): assorted fixage
- (`_dbus_string_equal_c_str`): assert `c_str != NULL`, fix logic so the function works
- (`_dbus_string_ends_with_c_str`): fix overflow thingy
- (`_dbus_string_base64_encode`): overflow fix
- (`_dbus_string_validate_ascii`): overflow
- (`_dbus_string_validate_nul`): overflow

2003-02-26 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-marshal.c (`_dbus_marshall_test`): fix to work with `DISABLE_ASSERTS`

2003-02-26 Alexander Larsson <alexl@redhat.com>

- * `configure.in`:
Set `DBUS_GLIB_THREADS_LIBS` for apps using `gthread-2.0`
- * `dbus/dbus-connection.c`:
* `dbus/dbus-connection.h`:
Fix `_dbus_connection_acquire_io_path` and `_dbus_connection_acquire_dispatch`.
Add `_dbus_connection_set_wakeup_main_function` and use it when queueing incoming and outgoing messages.
- * `dbus/dbus-dataslot.c`:
Threadsafe usage of `DBusDataSlotAllocator`

- * dbus/dbus-message.c: (dbus_message_get_args_iter):
dbus_new can fail.
- * dbus/dbus-server-unix.c:
Add todo comment
- * glib/dbus-gmain.c:
Implement the new wakeup functions for glib.
- * glib/Makefile.am:
- * glib/test-thread-client.c:
- * glib/test-thread-server.c:
- * glib/test-thread.h:
Initial cut at some thread test code. Not really done yet.

2003-02-26 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-connection.c
(dbus_connection_send_message_with_reply_and_block): fix crash
where we ref'd the outgoing message instead of the returned reply
- * dbus/dbus-transport-unix.c (do_authentication): check read
watch
at the end of this function, so if we didn't need to read for
authentication, we reinstall it for receiving messages
- * dbus/dbus-message.c (dbus_message_new_reply): allow replies to
a NULL sender for peer-to-peer case
- * dbus/dbus-transport-unix.c (check_read_watch): handle
!authenticated case correctly
- * glib/dbus-gmain.c: add support for DBusServer
- * dbus/dbus-server.c: add data slot support
- * glib/dbus-gmain.c (dbus_connection_setup_with_g_main): check
return values and handle errors
- * dbus/dbus-dataslot.c: factor out the data slot stuff from
DBusConnection
- * Doxyfile.in (INPUT): add glib subdir
- * glib/dbus-gmain.c (dbus_connection_setup_with_g_main): rename
setup_with_g_main instead of hookup_with_g_main; write docs

2003-02-24 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-marshal.c: (_dbus_marshal_validate_arg):
- * dbus/dbus-message-builder.c: (_dbus_message_data_load):

- * dbus/dbus-message.c: (dbus_message_append_boolean),
(dbus_message_append_boolean_array),
(dbus_message_get_args_valist), (_dbus_message_test):
- * dbus/dbus-message.h:
- * doc/dbus-specification.sgml:
Various fixes as pointed out by Havoc.

- * test/data/invalid-messages/bad-boolean-array.message:
- * test/data/invalid-messages/bad-boolean.message:
Add invalid boolean value test cases.

2003-02-24 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-internals.c: (_dbus_type_to_string):
- * dbus/dbus-marshal.c: (_dbus_marshall_get_arg_end_pos),
(_dbus_marshall_validate_arg):
- * dbus/dbus-message-builder.c: (_dbus_message_data_load):
- * dbus/dbus-message.c: (dbus_message_append_args_valist),
(dbus_message_append_boolean),
(dbus_message_append_boolean_array),
(dbus_message_get_args_valist), (dbus_message_iter_get_boolean),
(dbus_message_iter_get_int32), (dbus_message_iter_get_uint32),
(dbus_message_iter_get_double),
(dbus_message_iter_get_boolean_array), (message_iter_test):
- * dbus/dbus-message.h:
- * dbus/dbus-protocol.h:
- * doc/dbus-specification.sgml:
- * test/data/valid-messages/lots-of-arguments.message:
Add support for boolean and boolean array types.

2003-02-23 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-keyring.c: finish most of this implementation and
simple unit test

- * dbus/dbus-errors.c (dbus_set_error_const, dbus_set_error): make
these barf if the error isn't cleared to NULL

- * dbus/dbus-sysdeps.c (_dbus_delete_file): set error on failure
(_dbus_create_directory): new function

- * dbus/dbus-errors.c (dbus_set_error): fix warning

- * dbus/dbus-string.c (_dbus_string_hex_encode): new function
(_dbus_string_hex_decode): new function
(test_hex_roundtrip): test code

- * dbus/dbus-sha.c (_dbus_sha_compute): use dbus_string_hex_encode

- * dbus/dbus-md5.c (_dbus_md5_compute): use dbus_string_hex_encode

- * dbus/dbus-sysdeps.c (_dbus_string_save_to_file): make this use

the save-to-temp/rename trick to atomically write the new file
(_dbus_string_parse_uint): new function

2003-02-22 Havoc Pennington <hp@pobox.com>

* test/Makefile.am (dist-hook): fix dist for test/data/sha-1

2003-02-22 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (dbus_message_iter_get_string_array):
(dbus_message_iter_get_byte_array): Fix up doxygen warnings

* dbus/dbus-sha.c: add implementation of SHA-1 algorithm

* dbus/test/data/sha-1: add US government test suite for SHA-1

2003-02-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-marshal.c: (_dbus_demarshal_string_array):
Make string arrays NULL-terminated.

* dbus/dbus-memory.c: (dbus_free_string_array):

* dbus/dbus-memory.h:

New function for freeing NULL-terminated string arrays.

* dbus/dbus-message-builder.c: (append_quoted_string),
(_dbus_message_data_load):
Add support for array types.

* dbus/dbus-message.c: (check_message_handling):
Add more types as test cases.

* dbus/dbus-sysdeps.c: (_dbus_string_parse_int),
(_dbus_string_parse_double):
Add the start offset to the end offset.

* test/data/valid-messages/lots-of-arguments.message:
New test message with lots of arguments.

2003-02-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c: (dbus_message_append_nil),
(dbus_message_append_int32), (dbus_message_append_uint32),
(dbus_message_append_double), (dbus_message_append_string),
(dbus_message_append_int32_array),
(dbus_message_append_uint32_array),
(dbus_message_append_double_array),
(dbus_message_append_byte_array),
(dbus_message_append_string_array):
Fix all out-of-memory handling in these functions.

2003-02-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c: (dbus_message_append_nil):
Fix a silly.

2003-02-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c: (dbus_message_append_args_valist),
(dbus_message_append_nil), (dbus_message_append_int32_array),
(dbus_message_append_uint32_array),
(dbus_message_append_double_array),
(dbus_message_append_byte_array),
(dbus_message_append_string_array),
(dbus_message_get_args_valist),
(dbus_message_iter_get_int32_array),
(dbus_message_iter_get_uint32_array),
(dbus_message_iter_get_double_array),
(dbus_message_iter_get_byte_array),
(dbus_message_iter_get_string_array):

* dbus/dbus-message.h:
Add functions for appending and getting arrays.

2003-02-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-mempool.c (_dbus_mem_pool_new): Make the
element size at least 8 bytes, fixes mempool tests on
64-bit machines.

2003-02-20 Alexander Larsson <alexl@redhat.com>

* dbus/dbus-transport-unix.c (unix_do_iteration):
Unlock the connection mutex during a blocking select call.
Add todo about how we need a way to wake up the select.

* dbus/dbus-connection-internal.h:
* dbus/dbus-connection.c:
Add _dbus_connection_lock and _dbus_connection_unlock.

2003-02-19 Havoc Pennington <hp@pobox.com>

* Doxyfile.in (PREDEFINED): put DOXYGEN_SHOULD_SKIP_THIS in
Doxyfile.in, not Doxyfile

* dbus/dbus-keyring.c: do some hacking on this

* dbus/dbus-sysdeps.c (_dbus_delete_file): new

* dbus/dbus-errors.c (dbus_set_error_const): do not call
dbus_error_init
(dbus_set_error): remove dbus_error_init, check for message ==
NULL *before* we sprintf into it, and add @todo about including
system headers in this file

- * dbus/dbus-sysdeps.c (`_dbus_create_file_exclusively`): new
- * dbus/dbus-errors.h (`DBUS_ERROR_FAILED`): add
- * dbus/dbus-sysdeps.c (`get_user_info`): break this function out to get various bits of user information based on either username or user ID
- (`_dbus_homedir_from_username`): new function

2003-02-19 Anders Carlsson <andersca@codefactory.se>

- * `configure.in`:
Add check for nonposix `getpwnam_r`
- * dbus/dbus-mempool.c: (`_dbus_mem_pool_new`):
Align the pool element size to a `sizeof (void *)` boundary.
- * dbus/dbus-sysdeps.c: (`_dbus_setenv`),
(`_dbus_connect_unix_socket`),
(`_dbus_listen_unix_socket`), (`_dbus_credentials_from_username`):
General Solaris fixes.
- * `test/data/valid-messages/simplest-manual.message`:
Explicitly state that we want little-endian packing.

2003-02-19 Mikael Hallendal <micke@codefactory.se>

- * dbus/dbus-server.c (`dbus_server_listen`): Support tcp: addresses.
- * dbus/dbus-transport-unix.c
(`_dbus_transport_new_for_tcp_socket`):
Added to create a transport connecting using a tcp/ip socket.
- * dbus/dbus-sysdeps.c (`_dbus_connect_tcp_socket`): Added to connect
to a tcp socket at given host and port.
(`_dbus_listen_tcp_socket`): added to listen on tcp socket for
given
hostname and port.
- * dbus/dbus-server.c (`dbus_server_listen`): Support tcp: addresses.
- * dbus/dbus-server-unix.c (`_dbus_server_new_for_tcp_socket`):
Added to create a server listening on a TCP/IP socket.

2003-02-19 Havoc Pennington <hp@pobox.com>

Throughout: mop up all the Doxygen warnings and undocumented stuff.

* dbus/dbus-sysdeps.c (do_exec): do not use execvp, we don't want to search any paths.

* dbus/dbus-threads.c: move global mutex initializers to dbus-internals.h, multiple prototypes was confusing doxygen besides being kind of ugly

* Doxyfile (PREDEFINED): have Doxygen define DOXYGEN_SHOULD_SKIP_THIS so we can exclude things from docs with #ifndef DOXYGEN_SHOULD_SKIP_THIS (do not abuse the feature! it's for stuff like the autogenerated macros in dbus-md5.c, not just for things you don't feel like documenting...)

2003-02-18 Havoc Pennington <hp@pobox.com>

* dbus/dbus-string.c (_dbus_string_zero): new function

* dbus/dbus-md5.c: include MD5 implementation by L. Peter Deutsch, wrap it in some dbus-friendly API

* dbus/dbus-types.h: add 16-bit types

2003-02-18 Joe Shaw <joe@assbarn.com>

* dbus/dbus-auth.c (handle_server_data_stupid_test_mech): Just get credentials from our currently running process. (get_word): Fix a buglet where we were copying the entire length instead of relative to our position.

* dbus/dbus-hash.c (_dbus_hash_test): Don't try to allocate the keys on the stack... it's 640k of data.

* dbus/dbus-sysdeps.c (_dbus_read_credentials_unix_socket): Always read the credentials byte off the socket, even if we don't have SO_PEERCREC. (_dbus_poll): Implement poll() using select() for systems which don't have it.

* glib/test-dbus-glib.c (main): Print out an error if no parameters are given.

* test/data/auth/fallback.auth-script: Added. Tests that a client can fallback to a secondary auth mechanism if the first fails.

2003-02-18 Havoc Pennington <hp@pobox.com>

* AUTHORS: add Alex

2003-02-17 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: lots of cosmetic cleanups/rearrangement, add assorted FIXME, change DBUS_ADDRESS env variable to DBUS_BUS_ADDRESS, s/client/application/, s/server/bus/ (except in authentication section). Add a section "Message Bus Message Routing"

2003-02-17 Anders Carlsson <andersca@codefactory.se.>

Release 0.4

* NEWS: Update

2003-02-17 Anders Carlsson <andersca@codefactory.se>

* doc/dbus-specification.sgml:
Specification updates.

2003-02-17 Anders Carlsson <andersca@codefactory.se>

* bus/activation.c: (bus_activation_init), (child_setup),
(bus_activation_activate_service):

* bus/activation.h:

* bus/main.c: (main):

Set DBUS_ADDRESS environment variable.

* dbus/dbus-errors.c: (dbus_set_error):

Don't use va_copy since that's a C99 feature.

* dbus/dbus-sysdeps.c: (_dbus_setenv), (do_exec),
(_dbus_spawn_async):

* dbus/dbus-sysdeps.h:

Add child_setup_func to _dbus_spawn_async.

* doc/dbus-specification.sgml:

Update specification.

* test/spawn-test.c: (setup_func), (main):

Fix test.

2003-02-17 Alexander Larsson <alexl@redhat.com>

* dbus/dbus-connection.c

(_dbus_connection_handler_destroyed_locked):

Added todo.

2003-02-17 Anders Carlsson <andersca@codefactory.se>

* doc/.cvsignore:

* doc/Makefile.am:
* doc/dbus-test-plan.sgml:
Add test plan document.

* test/Makefile.am:
Fix distcheck.

2003-02-17 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-message.c: (decode_header_data),
(_dbus_message_loader_return_buffer):
Set the header padding amount when loading a message.

2003-02-16 Anders Carlsson <andersca@codefactory.se>

* bus/dispatch.c: (send_one_message):
Only send broadcast messages to registered connections.

* dbus/dbus-message.c: (dbus_message_name_is):
* dbus/dbus-message.h:
New convenience function.

* dbus/dbus-transport-debug.c: (do_reading):
Only dispatch one message per run.

* test/Makefile.am:
* test/bus-test.c: (new_connection_callback), (die),
(test_hello_client1_handler), (test_hello_client2_handler),
(test_hello_replies), (main):

* test/bus-test-loop.[ch]:
Add these.

2003-02-16 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_dispatch_message): fix
backward conditional

2003-02-16 Alexander Larsson <alex1@redhat.com>

* dbus/dbus-connection.c:
Implement sent_message_with_reply. (with_reply_and_block is still
busted).
Made dispatch_message not lose message if OOM.

* dbus/dbus-errors.h:
Add NoReply error (for reply timeouts).

2003-02-16 Alexander Larsson <alex1@redhat.com>

* dbus/dbus-hash.c (_dbus_hash_table_unref):
Actually free keys and values when destroying hashtable.

2003-02-16 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-auth.c: (client_try_next_mechanism):
Plug a leak.

* dbus/dbus-threads.c: (dbus_condvar_wait_timeout):
Return TRUE if there's no thread implementation around.

* glib/dbus-gmain.c: (free_source),
(dbus_connection_hookup_with_g_main):
Make sure to remove the GSource when the connection is finalized.

2003-02-16 Anders Carlsson <andersca@codefactory.se>

* bus/dispatch.c: (bus_dispatch_message_handler):
* dbus/dbus-errors.h:
Return an error if someone tries to send a message to a service
that doesn't exist.

2003-02-16 Anders Carlsson <andersca@codefactory.se>

* bus/activation.c: (load_directory), (bus_activation_init),
(bus_activation_activate_service):
* bus/activation.h:
* bus/driver.c:
(bus_driver_handle_activate_service),
(bus_driver_handle_message):
More work on the activation handling.

* dbus/dbus-errors.h:
Add some error messages

* dbus/dbus-message.c: (dbus_message_new_error_reply):
* dbus/dbus-message.h:
New function that creates an error message.

* dbus/dbus-protocol.h:
Add ACTIVATE_SERVER message.

* dbus/dbus-server-unix.c: (unix_handle_watch),
(_dbus_server_new_for_domain_socket):
Call _dbus_fd_set_close_on_exec.

* dbus/dbus-sysdeps.c: (make_pipe), (do_exec),
(_dbus_spawn_async), (_dbus_disable_sigpipe),
(_dbus_fd_set_close_on_exec):
* dbus/dbus-sysdeps.h:
Add _dbus_fd_set_close_on_exec function. Also add function that
checks
that all open fds are set to close-on-exec and warns otherwise.

```
* dbus/dbus-transport-unix.c:
(_dbus_transport_new_for_domain_socket):
Call _dbus_fd_set_close_on_exec.
```

2003-02-16 Havoc Pennington <hp@pobox.com>

```
* dbus/dbus-connection.c (dbus_connection_set_change_sigpipe):
allow people to avoid setting SIGPIPE to SIG_IGN
(_dbus_connection_new_for_transport): disable SIGPIPE unless
we've been asked not to
```

2003-02-15 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-list.c: (_dbus_list_append_link),
(_dbus_list_prepend_link):
* dbus/dbus-memory.c: (dbus_malloc), (dbus_malloc0),
(dbus_realloc):
Warning fixes.
```

2003-02-15 Anders Carlsson <andersca@codefactory.se>

```
* bus/Makefile.am:
* bus/activation.c: (bus_activation_entry_free),
(add_desktop_file_entry), (load_directory),
(bus_activation_init):
* bus/activation.h:
* bus/main.c: (main):
Add simple activation support, doesn't work yet though.
```

2003-02-15 Zack Rusin <zack@kde.org>

```
* qt/dbus-qthread.cpp: small casting fix
```

2003-02-15 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-errors.c: (dbus_set_error):
* dbus/dbus-errors.h:
Add a few errors and make dbus_set_error void.

* dbus/dbus-sysdeps.c:
(_dbus_errno_to_string), (close_and_invalidate), (make_pipe),
(write_err_and_exit), (read_ints), (do_exec),
(_dbus_spawn_async):
* dbus/dbus-sysdeps.h:
Add _dbus_spawn_async.

* test/spawn-test.c: (main):
Test for _dbus_spawn_async.
```

2003-02-15 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-internals.h:
```

Fix build without tests.

* dbus/dbus-list.c: (alloc_link):
Fix a segfault when a malloc fails.

* dbus/dbus-memory.c: (initialize_malloc_debug), (dbus_malloc),
(dbus_malloc0), (dbus_realloc):
Add support for malloc debugging.

2003-02-15 Alexander Larsson <alexl@redhat.com>

* dbus/dbus-threads.c:
* dbus/dbus-threads.h:
Add condvars. Remove static mutext from API.
Implement static mutexes by initializing them from threads_init.

* glib/dbus-gthread.c:
* qt/dbus-qthread.cpp:
Update with the thread api changes.

* dbus/dbus-list.c:
* dbus/dbus-list.h:
Turn StaticMutex into normal mutex + init function.
Export new functions _dbus_list_alloc_link, _dbus_list_free_link,
_dbus_list_append_link, _dbus_list_prepend_link

* dbus/dbus-sysdeps.c:
* dbus/dbus-sysdeps.h:
New type dbus_atomic_t, and new functions _dbus_atomic_inc,
_dbus_atomic_dec. Only slow fallback implementation at the
moment.

* dbus/dbus-protocol.h:
Add DBUS_MESSAGE_LOCAL_DISCONNECT define

* dbus/dbus-message.c:
Make ref/unref atomic.
Fix some docs.

* dbus/dbus-connection-internal.h:
* dbus/dbus-connection.c:
* dbus/dbus-connection.h:
Make threadsafe.
Change _peek to _borrow, _return & _steal_borrowed.
Change disconnect callback to event.
Make dbus_connection_dispatch_messages reentrant.

* dbus/dbus-transport.c:
Don't ref the connection on calls to the transport
implementation.

```
* dbus/dbus-message-handler.c:
Make threadsafe.

* glib/dbus-gmain.c:
Don't use peek_message anymore

* test/Makefile.am:
* test/debug-thread.c:
* test/debug-thread.h:
Simple thread implementation that asserts() on deadlocks in
single-threaded code.

* test/bus-test.c:
(main) Call debug_threads_init.

* test/watch.c:
Use disconnect message instead of disconnect callback.

* bus/connection.c:
* bus/connection.h:
Don't call dbus_connection_set_disconnect_function. Instead
export
bus_connection_disconnect.

* bus/dispatch.c:
Call bus_connection_disconnect when we get a disconnected
message.

2003-02-15 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (dbus_message_new): fool around with the
docs

2003-02-14 Havoc Pennington <hp@pobox.com>

* dbus/dbus-mempool.c: fail if the debug functions so indicate

* dbus/dbus-memory.c: fail if the debug functions indicate we
should

* dbus/dbus-internals.c (_dbus_set_fail_alloc_counter)
(_dbus_decrement_fail_alloc_counter): debug functions to
simulate memory allocation failures

2003-02-14 Havoc Pennington <hp@pobox.com>

* dbus/dbus-errors.h (struct DBusError): add a word of padding
to DBusError

2003-02-13 Anders Carlsson <andersca@codefactory.se>
```

```
* bus/driver.c: (bus_driver_handle_hello):
* bus/driver.h:
* bus/services.c: (bus_service_lookup):
Reorder message sending so we get a more sane order.

* test/bus-test.c: (message_handler):
Fix tyop.
```

2003-02-13 Anders Carlsson <andersca@codefactory.se>

```
* bus/driver.c: (bus_driver_send_service_deleted),
  (bus_driver_send_service_created),
(bus_driver_send_service_lost),
  (bus_driver_send_service_acquired), (bus_driver_handle_hello),
  (bus_driver_send_welcome_message),
  (bus_driver_handle_list_services),
  (bus_driver_handle_acquire_service),
  (bus_driver_handle_service_exists):
* dbus/dbus-bus.c: (dbus_bus_register_client),
  (dbus_bus_acquire_service), (dbus_bus_service_exists):
* dbus/dbus-errors.c: (dbus_result_to_string):
* dbus/dbus-errors.h:
* dbus/dbus-message.c: (dbus_message_append_args),
  (dbus_message_append_args_valist), (dbus_message_get_args),
  (dbus_message_get_args_valist), (dbus_message_get_args_iter),
  (dbus_message_iter_get_arg_type), (dbus_message_iter_get_string),
  (dbus_message_iter_get_byte_array),
  (dbus_message_iter_get_string_array), (message_iter_test),
  (check_message_handling), (_dbus_message_test):
* dbus/dbus-message.h:
* test/bus-test.c: (main):
Change fields to arguments in messages, so that they won't be
confused with header fields.

* glib/test-dbus-glib.c: (main):
Remove append_fields from hello message.
```

2003-02-13 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-errors.c:
* dbus/dbus-message.c:
* dbus/dbus-string.c:
Documentation fixes.
```

2003-02-13 Anders Carlsson <andersca@codefactory.se>

```
* glib/dbus-gmain.c: (timeout_handler), (add_timeout),
  (remove_timeout):
Implement support for timeouts in dbus-glib.
```

2003-02-13 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-message-builder.c: (_dbus_message_data_load):
- * dbus/dbus-message.c: (process_test_subdir):
- * test/break-loader.c: (find_breaks_based_on):

Plug some memory leaks.

2003-02-13 Richard Hult <rhult@codefactory.se>

- * bus/main.c: Fix build.
- * dbus/dbus-errors.h:
- * dbus/dbus-errors.c: Fix copyright for Anders.

2003-02-13 Anders Carlsson <andersca@codefactory.se>

- * bus/Makefile.am:
Add utils.[ch]
- * bus/connection.c: (bus_connection_foreach):
Fix a warning.
- * bus/desktop-file.c: (grow_lines_in_section), (grow_sections),
(unescape_string), (new_section), (parse_section_start),
(parse_key_value), (report_error), (bus_desktop_file_load),
(bus_desktop_file_get_string):
- * bus/desktop-file.h:
Use DBusError for error reporting.
- * bus/dispatch.c: (send_one_message),
(bus_dispatch_message_handler):
- * bus/driver.c: (bus_driver_send_service_deleted),
(bus_driver_send_service_created),
(bus_driver_send_service_lost),
(bus_driver_send_service_acquired), (bus_driver_handle_hello),
(bus_driver_send_welcome_message),
(bus_driver_handle_list_services),
(bus_driver_handle_acquire_service),
(bus_driver_handle_service_exists):
- * bus/loop.c: (bus_loop_run):
- * bus/main.c:
Use BUS_HANDLE_OOM instead of _DBUS_HANDLE_OOM.
- * bus/utils.c: (bus_wait_for_memory):
- * bus/utils.h:
New files with general utility functions.
- * dbus/dbus-internals.h:
Remove _DBUS_HANDLE_OOM.

2003-02-13 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-errors.c: (dbus_result_to_string), (dbus_error_init),
(dbus_error_free), (dbus_set_error_const), (dbus_set_error):

* dbus/dbus-errors.h:
Add DBusError structure.

2003-02-13 Anders Carlsson <andersca@codefactory.se>

* test/data/valid-messages/standard-acquire-service.message:
* test/data/valid-messages/standard-hello.message:
* test/data/valid-messages/standard-list-services.message:
* test/data/valid-messages/standard-service-exists.message:
Add some standard messages.

2003-02-13 Anders Carlsson <andersca@codefactory.se>

* bus/driver.c: (bus_driver_send_welcome_message),
(bus_driver_handle_list_services),
(bus_driver_handle_acquire_service),
(bus_driver_handle_service_exists), (bus_driver_handle_message):
Update for API changes in libdbus.

* dbus/dbus-message.c: (dbus_message_new_reply):
* dbus/dbus-message.h:
Remove the name argument. The spec states that replies shouldn't
have a name.

2003-02-13 Anders Carlsson <andersca@codefactory.se>

* bus/desktop-file.c: (parse_section_start), (parse_key_value),
(report_error), (bus_desktop_file_load), (lookup_section),
(lookup_line), (bus_desktop_file_get_raw),
(bus_desktop_file_get_string):
* bus/desktop-file.h:
Some fixes, and new functions for getting a key value from a
section.

2003-02-13 Havoc Pennington <hp@pobox.com>

* test/data/auth/fail-after-n-attempts.auth-script: new test

* dbus/dbus-auth.c (send_rejected): shutdown_mech() when we
reject the client.

2003-02-13 Havoc Pennington <hp@pobox.com>

* dbus/dbus-auth.c (handle_server_data_external_mech): args to
dbus_credentials_match were backward

* dbus/dbus-auth-script.c (_dbus_auth_script_run): support
NO_CREDENTIALS and ROOT_CREDENTIALS

* dbus/dbus-auth.c (_dbus_auth_do_work): move get_state() routine
into here. Never process more commands after we've reached an
end state; store further data as unused bytes.

- * test/data/auth/*: add more auth tests
- * dbus/dbus-auth-script.c (_dbus_auth_script_run): support EXPECT command to match exact string and EXPECT_UNUSED to match unused bytes
- * test/Makefile.am (dist-hook): fix to dist all the test stuff

2003-02-12 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-string.c (_dbus_string_pop_line): fix to also strip \r off of popped lines
- * dbus/dbus-auth.c (_dbus_auth_test): write code to run auth scripts
- * dbus/dbus-auth-script.c (_dbus_auth_script_run): when doing a SEND, append \r\n

2003-02-12 Havoc Pennington <hp@pobox.com>

- * dbus/Makefile.am: remove break-loader from the build, since it moved.
- * configure.in: add --enable-gcov to turn on coverage profiling flags and disable optimization

2003-02-10 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-auth-script.c, dbus/dbus-auth-script.h: sync initial cut at test framework for DBusAuth from laptop. Doesn't quite work yet but it compiles and I need to get it off the 266mhz laptop. ;-)
- * dbus/dbus-server-debug.c (_dbus_server_debug_accept_transport): fix a memleak in error case

2003-02-12 Anders Carlsson <andersca@codefactory.se>

- * bus/Makefile.am:
 - * bus/desktop-file.c:
 - * bus/desktop-file.h:
- Add a desktop file parser.

2003-02-11 Zack Rusin <zack@kde.org>

- * qt/message.[h|cpp]: sample implementation of the KDE wrapper for DBusMessage

2003-02-09 Zack Rusin <zack@kde.org>

- * test/bus-test.c: make_it_compile
- * doc/dbus-specification.sgml: minimal semantic fix

2003-02-06 Anders Carlsson <andersca@codefactory.se>

Release 0.3

- * NEWS: Update

2003-02-06 Anders Carlsson <andersca@codefactory.se>

- * dbus/Makefile.am:
 - * dbus/dbus-break-loader.c:
 - * test/Makefile.am:
 - * test/break-loader.c:
- Move dbus-break-loader to test/ and rename it to break-loader.

2003-02-02 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-keyring.c, dbus/dbus-keyring.h: template files for code to manage cookies in your home directory
- * dbus/dbus-sysdeps.c (_dbus_generate_random_bytes): new function
- * dbus/dbus-auth.c (get_state): impose a maximum number of tries to authenticate, then disconnect the client.

2003-02-03 Alexander Larsson <alexl@redhat.com>

- * dbus/dbus-message.c (dbus_message_append_fields): Correct docs.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * doc/dbus-specification.sgml: Update address format section.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * test/Makefile.am:
- * test/bus-test.c: (get_time), (add_timeout), (remove_timeout), (message_handler), (new_connection_callback), (loop_quit), (loop_run), (main): Add bus test.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * bus/driver.c: (bus_driver_handle_service_exists): Simplify the code a bit.
- * dbus/dbus-bus.c: (dbus_bus_service_exists): Fix a silly.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * bus/Makefile.am:
Add libdbus-daemon.la and link to it.

2003-02-01 James Willcox <jwillcox@gnome.org>

- * bus/driver.c: (bus_driver_handle_own_service):
Actually include the service reply code in the message.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * bus/driver.c: (bus_driver_handle_service_exists):
Don't unref the incoming message.

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus.h: Add dbus-address.h and dbus-bus.h

2003-02-02 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-server.c: (dbus_server_listen):
- * dbus/dbus-transport.c: (_dbus_transport_open):
ifdef out the calls to the debug transport and server.

2003-02-02 Alexander Larsson <alexl@redhat.com>

- * dbus/dbus-watch.c (dbus_watch_get_flags):
Add note in the docs that ERROR or HANGUP won't be returned
and are assumed always on.
- * glib/dbus-gmain.c (add_watch):
Always add IO_ERR | IO_HUP
- * dbus/dbus-message.h:
Add semicolon after dbus_message_iter_get_string_array().
Makes qt code build again

2003-02-01 Anders Carlsson <andersca@codefactory.se>

- * bus/driver.c: (create_unique_client_name),
(bus_driver_handle_hello):
Don't take a name, just use a numeric id to identify
each client.
- * dbus/Makefile.am:
- * dbus/dbus-bus.c: (dbus_bus_register_client),
(dbus_bus_acquire_service), (dbus_bus_service_exists):
- * dbus/dbus-bus.h:
Add new convenience functions for communicating with the bus.

* dbus/dbus-message.h:

* dbus/dbus-protocol.h:
Fix a typo.

2003-02-01 Alexander Larsson <alexl@redhat.com>

* dbus/dbus-message.c (dbus_message_append_fields):
Add some more doc comments.

2003-02-01 Havoc Pennington <hp@pobox.com>

* dbus/dbus-break-loader.c (randomly_modify_length): change
a 4-byte value in the message as if it were a length

* dbus/dbus-sysdeps.c (_dbus_string_save_to_file): don't set
execute bit on saved files

2003-02-01 Havoc Pennington <hp@pobox.com>

* dbus/dbus-break-loader.c (main): new program to find messages
that break the loader.

* dbus/dbus-sysdeps.c (_dbus_string_append_uint): new function

* dbus/dbus-sysdeps.c (_dbus_string_save_to_file): new function

* dbus/dbus-string.c (_dbus_string_set_byte): new

2003-01-31 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c: refactor the test code to be more general,
in preparation for writing a "randomly permute test cases to
try to break the loader" program.

2003-01-31 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: work on the specification

* dbus/dbus-message.c (_dbus_message_loader_return_buffer): check
the protocol version of the message.

* dbus/dbus-protocol.h: drop special _REPLY names, the spec
no longer specifies that.

(DBUS_SERVICE_REPLY_SERVICE_EXISTS): fix flags (1/2/4/8 not
1/2/3/4)

* dbus/dbus-marshal.c (_dbus_marshall_get_arg_end_pos): add
missing

"break" for DBUS_TYPE_NIL, remove @todo

2003-01-31 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (dbus_message_set_is_error_reply): rename just set_is_error/get_is_error as this is a commonly-used function, and write docs.

2003-01-31 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-address.c: (dbus_address_entry_free):
Free key and value lists.

* dbus/dbus-internals.c: (_dbus_type_to_string):
Add the types we didn't have.

* dbus/dbus-marshal.c: (_dbus_marshal_get_arg_end_pos),
(_dbus_marshal_validate_arg):
Add NIL types.

* dbus/dbus-message.c: (dbus_message_set_sender):
Remove todo about being able to set sender to NULL.

(dbus_message_set_is_error_reply),
(dbus_message_get_is_error_reply):

* dbus/dbus-message.h:
New functions.

* dbus/dbus-protocol.h:
Add error reply flag.

* test/data/valid-messages/opposite-endian.message:
Add NIL type to test.

2003-01-31 Havoc Pennington <hp@pobox.com>

* doc/dbus-specification.sgml: fully specify the header. Add flags and major protocol version, and change header/body len to unsigned.

* dbus/dbus-message-builder.c (append_saved_length): append
length
as uint32

* dbus/dbus-message.c (dbus_message_create_header): change header length and body length to unsigned. Add the new fields from the spec
(_dbus_message_loader_return_buffer): unsigned header/body len

2003-01-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-auth.c: rework to use only REJECTED, no MECHANISMS

* doc/dbus-sasl-profile.txt: drop MECHANISMS and just use REJECTED, suggested by Mark McLoughlin

2003-01-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-server.c (dbus_server_listen): @todo about how we need

a better way to report errors here. e.g. "unix address lacks path" or something. also "no such file" when the path doesn't exist, etc.

* dbus/dbus-address.c (dbus_address_entries_free): add @todo about

leaking list nodes

(dbus_parse_address): add @todo about documenting address format, and allowing , and ; to be escaped

2003-01-30 Anders Carlsson <andersca@codefactory.se>

* dbus/Makefile.am:
Add dbus-address.[ch]

* dbus/dbus-address.c: (dbus_address_entry_free),
(dbus_address_entries_free), (create_entry),
(dbus_address_entry_get_method), (dbus_address_entry_get_value),
(dbus_parse_address), (_dbus_address_test):

* dbus/dbus-address.h:
New files for dealing with address parsing.

* dbus/dbus-connection.c:
Document timeout functions.

* dbus/dbus-message.c:
Document dbus_message_new_from_message.

* dbus/dbus-server-debug.c:
Document.

* dbus/dbus-server.c: (dbus_server_listen):
Parse address and use correct server implementation.

* dbus/dbus-string.c: (_dbus_string_find_to),
(_dbus_string_test):

* dbus/dbus-string.h:
New function with test.

* dbus/dbus-test.c: (dbus_internal_symbol_do_not_use_run_tests):
* dbus/dbus-test.h:
Add address tests.

* dbus/dbus-transport-debug.c:
Document.

* dbus/dbus-transport.c: (_dbus_transport_open):

Parse address and use correct transport implementation.

2003-01-30 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c: use message->byte_order instead of DBUS_COMPILER_BYTE_ORDER throughout.

(dbus_message_create_header): pad header to align the start of the body of the message to 8-byte boundary

* dbus/dbus-marshal.h: make all the demarshalers take const DBusString arguments.

* dbus/dbus-message.c (_dbus_message_loader_return_buffer): validate message args here, so we don't have to do slow validation later, and so we catch bad messages as they are incoming. Also add

better checks on header_len and body_len. Also fill in message->byte_order

* dbus/dbus-string.c (_dbus_string_validate_utf8): new (not implemented properly)

(_dbus_string_validate_nul): new function to check all-nul

* dbus/dbus-marshal.c (_dbus_marshal_get_field_end_pos): rename get_arg_end_pos and remove all validation

(_dbus_marshal_validate_arg): actually do validation here.

2003-01-29 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (check_message_handling): fix assertion failure on set_client_serial

2003-01-28 Havoc Pennington <hp@pobox.com>

* dbus/dbus-server-debug.c: Add doc section comments

* dbus/dbus-transport-debug.c: add doc section comments

2003-01-28 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (_dbus_string_base64_decode): append bytes in

the reverse order from how I had it

(_dbus_string_base64_encode): reverse encoding order. I was basically byteswapping everything during encoding.

2003-01-28 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-connection-internal.h:

* dbus/dbus-connection.c: (_dbus_connection_add_timeout), (_dbus_connection_remove_timeout):

Add functions for adding and removing timeouts.

* dbus/dbus-message.c: (dbus_message_new_from_message):
Add new function that takes a message and creates an exact
copy of it, but with the refcount set to 1.

(check_message_handling):

Fix build error.

* dbus/dbus-server-protected.h:

* dbus/dbus-server.c: (_dbus_server_init_base),
(_dbus_server_finalize_base), (_dbus_server_add_timeout),
(dbus_server_set_timeout_functions):

(_dbus_server_remove_timeout):

New functions so that a server can add and remove timeouts.

(dbus_server_listen):

Add commented out call to dbus_server_debug_new.

* dbus/dbus-timeout.c: (_dbus_timeout_new):

Actually set the handler, doh.

* dbus/dbus-transport.c: (_dbus_transport_open):

Add commented out call to dbus_transport_debug_client_new.

* dbus/Makefile.am:

Add dbus-transport-debug.[ch] and dbus-server-debug.[ch]

2003-01-28 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (check_message_handling): function to check
on the loaded message, iterates over it etc.

2003-01-28 Havoc Pennington <hp@pobox.com>

* test/Makefile.am (dist-hook): fix make distdir

* dbus/Makefile.am (TESTS_ENVIRONMENT): fix make check

2003-01-27 Havoc Pennington <hp@pobox.com>

* dbus/dbus-mempool.c (time_for_size): replace printf with
_dbus_verbose

* dbus/dbus-message-builder.c (_dbus_message_data_load): allow
empty lines; fix the SAVE_LENGTH stuff to be
START_LENGTH/END_LENGTH so it actually works; couple other
bugfixes

* test/Makefile.am (dist-hook): add dist-hook for .message files

* dbus/dbus-string.c (DBUS_STRING_COPY_PREAMBLE): source of a

copy

can be constant or locked.
(_dbus_string_free): allow freeing a const string as documented/intended

* dbus/dbus-sysdeps.c (_dbus_concat_dir_and_file): utility

* dbus/dbus-test-main.c (main): take an argument which is the directory containing test data

* dbus/dbus-message.c (_dbus_message_test): pass a test_data_dir argument to this and load all the messages in test/data/ checking that they can be loaded or not loaded as appropriate.

2003-01-27 Anders Carlsson <andersca@codefactory.se>

* bus/dispatch.c: (bus_dispatch_message_handler): Dispatch messages sent to services.

* bus/driver.c: (bus_driver_send_service_deleted),
(bus_driver_send_service_created),
(bus_driver_send_service_lost),
(bus_driver_send_service_acquired):
Add helper functions for sending service related messages.

(bus_driver_send_welcome_message):
Send HELLO_REPLY instead of WELCOME.

(bus_driver_handle_list_services):
Send LIST_SERVICES_REPLY instead of SERVICES.

(bus_driver_handle_own_service),
(bus_driver_handle_service_exists):
New message handlers.

(bus_driver_handle_message):
Invoke new message handlers.

(bus_driver_remove_connection):
Don't remove any services here since that's done automatically by bus_service_remove_owner now.

* bus/driver.h:
New function signatures.

* bus/services.c: (bus_service_add_owner):
Send ServiceAcquired message if we're the only primary owner.

(bus_service_remove_owner):
Send ServiceAcquired/ServiceLost messages.

(bus_service_set_prohibit_replacement),
(bus_service_get_prohibit_replacement):

Functions for setting prohibit replacement.

(bus_service_has_owner):

New function that checks if a connection is in the owner queue of a certain service.

* bus/services.h:

Add new function signatures.

* dbus/dbus-list.c: (_dbus_list_test):

Add tests for _dbus_list_remove_last and traversing the list backwards.

* dbus/dbus-list.h:

Fix a typo in _dbus_list_get_prev_link, if we're at the first element we can't go further, so return NULL then.

* dbus/dbus-protocol.h:

Add new messages, service flags and service replies.

2003-01-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message-builder.c: implement, completely untested.

* test/data/*: add data to be used in testing.

".message" files are our simple loadable text format.

".message-raw" will be binary dumps of messages.

* dbus/dbus-string.c (_dbus_string_starts_with_c_str): new

2003-01-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-sysdeps.c (_dbus_file_get_contents): new function

* dbus/dbus-errors.c (dbus_result_to_string): add file errors

* dbus/dbus-message-builder.c: new file, will contain code to load up messages from files. Not implemented yet.

2003-01-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.c (dbus_message_set_sender): support deleting the sender by setting to NULL

2003-01-26 Havoc Pennington <hp@pobox.com>

The unit tests pass, but otherwise untested. If it breaks, the tests should have been better. ;-)

* bus/driver.c (bus_driver_handle_hello): return if we disconnect the connection.

* dbus/dbus-message.c: redo everything so we maintain message->header as the only copy of the various fields. This avoids the possibility of out-of-memory in some cases, for example dbus_message_lock() can't run out of memory anymore, and avoids extra copying. Figured I may as well go ahead and do this since it was busted for dbus_message_lock to not return failure on OOM, and dbus_message_write_header was totally unchecked for OOM. Also fixed some random other bugs.

* dbus/dbus-marshal.c (_dbus_marshal_get_field_end_pos): verify that strings are nul-terminated. Also, end_pos can be equal to string length just not greater than, I think.

(dbus_marshal_set_int32): new function

(dbus_marshal_set_uint32): new function

(dbus_marshal_set_string): new function

* dbus/dbus-connection.c (_dbus_connection_new_for_transport):

fix

a warning, init timeout_list to NULL

(dbus_connection_send_message): don't use uninitialized variable "serial"

* dbus/dbus-string.c (_dbus_string_replace_len): new function

2003-01-26 Anders Carlsson <andersca@codefactory.se>

* bus/driver.c: (bus_driver_handle_hello),

(bus_driver_send_welcome_message):

Plug leaks

2003-01-26 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-auth.c: (process_auth), (_dbus_auth_unref):

* dbus/dbus-connection.c: (_dbus_connection_new_for_transport),

(dbus_connection_unref):

* dbus/dbus-marshal.c: (_dbus_marshal_test):

* dbus/dbus-message.c: (dbus_message_unref),

Plug memory leaks.

(dbus_message_get_fields):

Remove debugging printout.

(dbus_message_loader_return_buffer):

Don't store the header string.

(dbus_message_test):

Plug leaks.

2003-01-26 Richard Hult <rhult@codefactory.se>

* glib/dbus-gmain.c (dbus_connection_dispatch): Traverse a copy of the file descriptor list, since it can change under us.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* glib/dbus-gmain.c: (dbus_connection_prepare), (dbus_connection_check), (dbus_connection_dispatch), (add_watch), (remove_watch), (dbus_connection_hookup_with_g_main): Rewrite the glib handling to use its own GSource instead of a GIOChannel so we can catch messages put in the queue while waiting for a reply.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* bus/Makefile.am:
* bus/connection.c: (connection_disconnect_handler), (connection_watch_callback), (bus_connection_setup):
* bus/dispatch.c: (send_one_message), (bus_dispatch_broadcast_message), (bus_dispatch_message_handler), (bus_dispatch_add_connection), (bus_dispatch_remove_connection):
* bus/dispatch.h:
* bus/driver.c: (bus_driver_send_service_deleted), (bus_driver_send_service_created), (bus_driver_handle_hello), (bus_driver_send_welcome_message), (bus_driver_handle_list_services), (bus_driver_remove_connection), (bus_driver_handle_message):
* bus/driver.h:
Refactor code, put the message dispatching in its own file. Use _DBUS_HANDLE_OOM. Also send ServiceDeleted messages when a client is disconnected.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-internals.h:
Add _DBUS_HANDLE_OOM macro, it doesn't do anything currently.

* dbus/dbus-message.c: (dbus_message_get_sender):
* dbus/dbus-message.h:
Implement dbus_message_get_sender.

* dbus/dbus-protocol.h:
Add message and service defines.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-connection.c: (dbus_connection_send_message):
* dbus/dbus-message-internal.h:

* dbus/dbus-message.c: (_dbus_message_get_client_serial),
(dbus_message_write_header):
Remove _dbus_message_unlock and don't set the client serial on a
message if one already exists.

2003-01-24 Havoc Pennington <hp@pobox.com>

* dbus/dbus-list.c (alloc_link): put a thread lock on the global
list_pool

* bus/driver.c (bus_driver_handle_list_services): fix a leak
on OOM

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-list.c: (alloc_link), (free_link):
Use a memory pool for the links.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* bus/connection.c: (bus_connection_foreach):
* bus/connection.h:
Add new bus_connection_foreach function.

* bus/driver.c: (send_one_message),
(bus_driver_broadcast_message):
Add function that broadcasts a message to all clients.

(bus_driver_send_service_created), (bus_driver_handle_hello),
(bus_driver_send_welcome_message),
(bus_driver_handle_list_services), (bus_driver_message_handler):
Implement functions that take care of listing services, and
notifying
clients when new services are created.

* bus/services.c: (bus_services_list):
* bus/services.h:
Add new function that returns an array of strings with the
currently
registered services.

* glib/dbus-glib.h:
* glib/dbus-gmain.c:
Update copyright year.

2003-01-25 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-connection.c: (dbus_connection_send_message):
Unlock the message in case it was sent earlier.

(dbus_connection_send_message_with_reply_and_block):
Remove the reply message from the list.

```
* dbus/dbus-marshal.c: (_dbus_demarshal_string_array):
Set array_len and new_pos correctly.

(_dbus_marshall_test):
Remove debug output.

* dbus/dbus-message-internal.h:
* dbus/dbus-message.c: (_dbus_message_get_reply_serial):
New function that returns the reply serial.

(_dbus_message_unlock):
New function that unlocks a message and resets its header.

(dbus_message_append_string_array),
(dbus_message_get_fields_valist),
(dbus_message_iter_get_field_type),
(dbus_message_iter_get_string_array),
(dbus_message_get_fields),
(dbus_message_append_fields_valist):
Handle string arrays.

(dbus_message_set_sender):
Make this function public since the bus daemon needs it.

(decode_header_data):
Set the reply serial to -1 initially.

* dbus/dbus-message.h:
Add dbus_message_set_sender.
```

2003-01-24 Havoc Pennington <hp@pobox.com>

```
* doc/dbus-specification.sgml: add some stuff
```

2003-01-22 Havoc Pennington <hp@pobox.com>

```
* doc/dbus-specification.sgml: Start to document the protocol.
```

2003-01-22 Havoc Pennington <hp@pobox.com>

```
* dbus/dbus-connection.c
(dbus_connection_send_message_with_reply_and_block): add some
@todo
```

```
* bus/driver.c (bus_driver_add_connection): add a FIXME about
memleak
```

2003-01-21 Havoc Pennington <hp@pobox.com>

```
(patch untested because can't compile)
```

* bus/driver.c (create_unique_client_name): make this function never recycle client names. Also, caller should initialize the DBusString.

* dbus/dbus-sysdeps.c (_dbus_get_current_time): new function

2003-01-21 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-marshal.c: (_dbus_marshal_double),
(_dbus_marshal_int32), (_dbus_marshal_uint32),
(_dbus_marshal_int32_array), (_dbus_marshal_uint32_array),
(_dbus_marshal_double_array), (_dbus_marshal_string_array),
(_dbus_demarshal_int32_array), (_dbus_demarshal_uint32_array),
(_dbus_demarshal_double_array), (_dbus_demarshal_string_array),
(_dbus_marshal_get_field_end_pos), (_dbus_marshal_test):
* dbus/dbus-marshal.h:
* dbus/dbus-protocol.h:
Add support for marshalling and demarshalling integer, double and string arrays.

2003-01-21 Anders Carlsson <andersca@codefactory.se>

* bus/Makefile.am:
Add driver.[ch]

* bus/connection.c: (connection_disconnect_handler):
Remove the connection from the bus driver's list.

(connection_watch_callback): Dispatch messages.

(free_connection_data): Free connection name.

(bus_connection_setup): Add connection to the bus driver's list.

(bus_connection_remove_owned_service):

(bus_connection_set_name), (bus_connection_get_name):

Add functions for setting and getting the connection's name.

* bus/connection.h:
Add function headers.

* bus/driver.c: (create_unique_client_name),
(bus_driver_handle_hello_message),
(bus_driver_send_welcome_message), (bus_driver_message_handler),
(bus_driver_add_connection), (bus_driver_remove_connection):

* bus/driver.h:

* bus/main.c:

* bus/services.c: (bus_service_free):

* bus/services.h:

New file that handles communication and registration with the

bus

itself.

2003-01-21 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-connection.c: (dbus_connection_send_message):
Add a new client_serial parameter.
- (dbus_connection_send_message_with_reply):
Remove a @todo since we've implemented the blocking function.
- (dbus_connection_send_message_with_reply_and_block):
New function that sends a message and waits for a reply and then returns the reply.
- * dbus/dbus-connection.h:
Add new functions.
- * dbus/dbus-errors.c: (dbus_result_to_string):
- * dbus/dbus-errors.h:
Add new DBUS_RESULT.
- * dbus/dbus-message-internal.h:
- * dbus/dbus-message.c: (_dbus_message_get_reply_serial),
(_dbus_message_set_sender), (dbus_message_write_header),
(dbus_message_new_reply), (decode_header_data),
(_dbus_message_loader_return_buffer), (_dbus_message_test):
- * dbus/dbus-message.h:
Add new functions that set the reply serial and sender.
Also marshal and demarshal them correctly and add test.
- * dbus/dbus-protocol.h:
Add new DBUS_MESSAGE_TYPE_SENDER.
- * glib/dbus-glib.h:
- * glib/dbus-gmain.c: (watch_callback), (free_callback_data),
(add_watch), (remove_watch), (add_timeout), (remove_timeout),
(dbus_connection_hookup_with_g_main):
- * glib/test-dbus-glib.c: (main):
Rewrite to use GIOChannel and remove the GSource crack.
- * test/echo-client.c: (main):
- * test/watch.c: (check_messages):
Update for changed APIs

2003-01-19 Anders Carlsson <andersca@codefactory.se>

- * dbus/Makefile.am: Add dbus-timeout.[cÄ\$]
- * dbus/dbus-connection.c: (_dbus_connection_new_for_transport):
Create a DBusTimeoutList.
(dbus_connection_set_timeout_functions): Add new function to set timeout callbacks
- * dbus/dbus-connection.h: Add public DBusTimeout API.

- * dbus/dbus-message.c: (dbus_message_get_service):
- * dbus/dbus-message.h: New function.
- * dbus/dbus-server.c: Fix small doc typo.
- * dbus/dbus-timeout.[ch]: New files for mainloop timeouts.

2003-01-19 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-string.c (_dbus_string_move_len): Don't delete all of the string, just as long as specified.

2003-01-19 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-connection.c (dbus_connection_get_is_authenticated): new function
- * dbus/dbus-server.c (dbus_server_set_max_connections) (dbus_server_get_max_connections, dbus_server_get_n_connections): keep track of current number of connections, and add API for setting a max (but haven't implemented enforcing the max yet)

2003-01-18 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-transport-unix.c (unix_do_iteration): only do the reading/writing if read_watch != NULL or write_watch != NULL.
- * dbus/dbus-message.c (_dbus_message_loader_return_buffer): fix the message loader code to actually load message->header and message->body into the newly-created message.
- * dbus/dbus-transport-unix.c (check_write_watch): fix a mem leak in OOM case
- * dbus/dbus-connection.c (dbus_connection_set_max_message_size) (dbus_connection_get_max_message_size) (dbus_connection_set_max_live_messages_size) (dbus_connection_get_max_live_messages_size): implement some resource limitation functions
- * dbus/dbus-resources.c: new file implementing some of the resource limits stuff
- * dbus/dbus-message.c (dbus_message_iter_get_byte_array): add missing docs, add @todo to handle OOM etc.
- * dbus/dbus-marshal.c (_dbus_demarshal_byte_array): add missing docs

2003-01-18 Havoc Pennington <hp@pobox.com>

* dbus/dbus-connection.c (dbus_connection_unref): disconnect the connection if it hasn't been already.

* dbus/dbus-connection.h: kill off the idea of an ErrorFunction, replace with DisconnectFunction.

2003-01-18 Havoc Pennington <hp@pobox.com>

Building --disable-verbose-mode --disable-asserts --disable-tests

cuts the library from 112K to 45K or so

* configure.in: check for varargs macro support, add --enable-verbose-mode, --enable-asserts.

* dbus/dbus-internals.h (_dbus_assert): support DBUS_DISABLE_ASSERT (_dbus_verbose): support DBUS_ENABLE_VERBOSE_MODE

2003-01-18 Havoc Pennington <hp@pobox.com>

* dbus/dbus-test.c: include config.h so that tests actually run

* dbus/dbus-string.c: add assertions that stuff is 8-byte aligned, so the failure mode when that assumption fails will be plenty obvious.

2003-01-18 Havoc Pennington <hp@pobox.com>

* configure.in: default --enable-tests to \$USE_MAINTAINER_MODE

* dbus/Makefile.am: fix it up so dbus-test-main.c is included in the distribution

* test/Makefile.am: don't use special variable "TESTS" for echo-* since we don't want to use those in make check

2003-01-15 Havoc Pennington <hp@redhat.com>

Release 0.2

* NEWS: update

2003-01-15 Havoc Pennington <hp@redhat.com>

* test/Makefile.am: fix so that test source code ends up in the distribution on make distcheck

2003-01-15 Havoc Pennington <hp@redhat.com>

Release 0.1.

* NEWS: update

2003-01-15 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-test.c (dbus_internal_symbol_do_not_use_run_tests):
fix build when --disable-tests
- * Makefile.am (EXTRA_DIST): put HACKING in here
- * HACKING: document procedure for making a tarball release.

2003-01-14 Anders Carlsson <andersca@codefactory.se>

- * bus/connection.c: (connection_error_handler),
(bus_connection_setup):
- * bus/main.c: (main):

Make sure that the DBusConnectionData struct is NULled out to prevent a segfault.

- * dbus/dbus-errors.c: (dbus_result_to_string):
- * dbus/dbus-errors.h:
- * dbus/dbus-message.c: (dbus_message_get_fields),
(dbus_message_get_fields_valist), (_dbus_message_test):
- * dbus/dbus-message.h:

Make dbus_message_get_fields return a result code so we can track invalid fields as well as oom.

2003-01-11 Havoc Pennington <hp@pobox.com>

- * configure.in: change --enable-test/--enable-ansi action-if-given
to enable_foo=\$enableval instead of enable_foo=yes

2003-01-08 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-string.c (_dbus_string_align_length): new function
- * dbus/dbus-test-main.c: move main() for test app here
- * dbus/dbus-test.c
(dbus_internal_symbol_do_not_use_run_tests): we have to export a symbol to run tests, because dbus-test isn't in the main library

Code review nitpicks.

- * dbus/dbus-message.c (dbus_message_write_header): add newlines for people with narrow emacs ;-). Assert client_serial was filled in. Assert message->name != NULL.
- (dbus_message_append_fields): have "first_field_type" arg separate from va list, needed for C++ binding that also uses varargs IIRC

and helps with type safety
(dbus_message_new): add @todo about using DBusString to store service/name internally
(dbus_message_new): don't leak ->service and ->name on OOM later in the function
(dbus_message_unref): free the service name
(dbus_message_get_fields): same change to varargs i.e. first_field_type
(_dbus_message_loader_return_buffer): assert that the message data is aligned (if not it's a bug in our code). Put in verbose griping about why we set corrupted = TRUE.
(decode_header_data): add FIXME that char* is evil. Was going to add FIXME about evil locale-specific string.h strncmp, but just switched to wacky string-as-uint32 optimization. Move check for "no room for field name" above get_const_data_len() to avoid assertion failure in get_const_data_len if we have trailing 2 bytes or the like. Check for service and name fields being provided twice. Don't leak service/name on error. Require field names to be aligned to 4 bytes.

* dbus/dbus-marshal.c: move byte swap stuff to header
(_dbus_pack_int32): uscore-prefix
(_dbus_unpack_int32): uscore-prefix
(_dbus_unpack_uint32): export
(_dbus_demarshal_string): add @todo complaining about use of memcpy()
(_dbus_marshall_get_field_end_pos): add @todo about bad error handling allowing corrupt data to go unchecked

2003-01-08 Havoc Pennington <hp@redhat.com>

* dbus/dbus-transport-unix.c (unix_do_iteration): add read/write to the select() as needed for authentication. (should be using _dbus_poll() not select, but for another day)

* dbus/dbus.h: include dbus/dbus-protocol.h

2003-01-08 Anders Carlsson <andersca@codefactory.se>

* dbus/Makefile.am (dbusinclude_HEADERS): Install dbus-connection.h

2003-01-08 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-internals.c: (_dbus_type_to_string):
New function that returns a string describing a type.

* dbus/dbus-marshal.c: (_dbus_demarshal_byte_array):
* dbus/dbus-marshal.h:
* dbus/dbus-message.c: (dbus_message_get_fields_valist),

```
(dbus_message_iter_get_field_type),
(dbus_message_iter_get_double),
(dbus_message_iter_get_byte_array):
* dbus/dbus-message.h:
Add new convenience functions for appending and getting message
fields.
```

Also add demarshalling routines for byte arrays.

2003-01-07 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-connection-internal.h:
* dbus/dbus-connection.c: (_dbus_connection_new_for_transport),
(_dbus_connection_get_next_client_serial),
(dbus_connection_send_message):
* dbus/dbus-internals.h:
* dbus/dbus-marshal.c: (unpack_uint32), (dbus_unpack_int32),
(dbus_pack_int32), (_dbus_marshall_double), (_dbus_marshall_int32),
(_dbus_marshall_uint32), (_dbus_demarshal_double),
(_dbus_demarshal_int32), (_dbus_demarshal_uint32),
(_dbus_demarshal_string), (_dbus_marshall_get_field_end_pos),
(_dbus_verbose_bytes), (_dbus_marshall_test):
* dbus/dbus-marshal.h:
* dbus/dbus-message-internal.h:
* dbus/dbus-message.c: (_dbus_message_set_client_serial),
(dbus_message_write_header), (_dbus_message_lock),
(dbus_message_new), (dbus_message_ref), (dbus_message_unref),
(dbus_message_get_name), (dbus_message_append_int32),
(dbus_message_append_uint32), (dbus_message_append_double),
(dbus_message_append_string), (dbus_message_append_byte_array),
(dbus_message_get_fields_iter), (dbus_message_iter_ref),
(dbus_message_iter_unref), (dbus_message_iter_has_next),
(dbus_message_iter_next), (dbus_message_iter_get_field_type),
(dbus_message_iter_get_string), (dbus_message_iter_get_int32),
(dbus_message_iter_get_uint32), (dbus_message_iter_get_double),
(decode_header_data), (_dbus_message_loader_return_buffer),
(message_iter_test), (_dbus_message_test):
* dbus/dbus-message.h:
* dbus/dbus-protocol.h:
* dbus/dbus-test.c: (main):
* dbus/dbus-test.h:
* glib/test-dbus-glib.c: (message_handler), (main):
* test/echo-client.c: (main):
* test/watch.c: (check_messages):
Make messages sendable and receivable for real.
```

2003-01-07 Anders Carlsson <andersca@codefactory.se>

```
* dbus/dbus-marshal.c: (_dbus_marshall_double),
(_dbus_marshall_string), (_dbus_marshall_byte_array):
* dbus/dbus-message.c: (dbus_message_append_int32),
(dbus_message_append_uint32), (dbus_message_append_double),
(dbus_message_append_string), (dbus_message_append_byte_array):
```

Handle OOM restoration.

2003-01-07 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-marshal.c: (`_dbus_marshall_string`), (`_dbus_demarshal_string`), (`_dbus_marshall_test`):
- * dbus/dbus-marshal.h:
- * dbus/dbus-message.c: (`dbus_message_get_name`), Document these functions.

- (`dbus_message_append_int32`), (`dbus_message_append_uint32`), (`dbus_message_append_double`), (`dbus_message_append_string`), (`dbus_message_append_byte_array`):

- * dbus/dbus-message.h: Add functions for adding message fields of different types.

- * dbus/dbus-protocol.h: Add the different types.

2003-01-05 Havoc Pennington <hp@pobox.com>

- * bus/connection.c: implement routines for handling connections, first thing is keeping a list of owned services on each connection and setting up watches etc.

- * bus/services.c: implement a mapping from service names to lists of connections

- * dbus/dbus-hash.c: add `DBUS_HASH_POINTER`

- * dbus/dbus-threads.c (`dbus_static_mutex_lock`): add functions to use static mutexes for global data

- * dbus/dbus-connection.c (`dbus_connection_set_data`): add new collection of functions to set/get application-specific data on the `DBusConnection`.

2003-01-04 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-sysdeps.c (`_dbus_sleep_milliseconds`): new function (`_dbus_poll`): new function

- * dbus/dbus-internals.h (`_DBUS_STRUCT_OFFSET`): new macro copied from GLib

- * bus/loop.c: initial code for the daemon main loop

2003-01-04 Havoc Pennington <hp@pobox.com>

- * test/watch.c (`error_handler`): make it safe if the error handler is called multiple times (if we s/error handler/disconnect

handler/ we should just guarantee it's called only once)

* dbus/dbus-transport.c (`_dbus_transport_disconnect`): call the error handler on disconnect (it's quite possible we should just change the error handler to a "disconnect handler," I'm not sure we have any other meaningful errors)

* configure.in: check for `getpwnam_r`

* dbus/dbus-transport.c, dbus/dbus-transport-unix.c, dbus/dbus-auth.c: add credentials support, add EXTERNAL auth mechanism as in SASL spec, using socket credentials

* dbus/dbus-sysdeps.c (`_dbus_read_credentials_unix_socket`): new function

(`_dbus_send_credentials_unix_socket`): new function

* dbus/dbus-sysdeps.c (`_dbus_accept_unix_socket`): rename just `dbus_accept()`

(`_dbus_write`): only check `errno` if `<0` returned

(`_dbus_write_two`): ditto

2003-01-02 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-marshal.c: (`_dbus_marshall_utf8_string`), (`_dbus_marshall_byte_array`), (`_dbus_demarshal_utf8_string`), (`_dbus_marshall_test`):

* dbus/dbus-marshal.h:

Add `_dbus_marshall_byte_array` and rename `_dbus_marshall_string` to `_dbus_marshall_utf8_string`. Also fix some tests.

2002-12-28 Harri Porten <porten@kde.org>

* configure.in: added check for C++ compiler and a very cheesy check for the Qt integration

* Makefile.am (SUBDIRS): compile qt subdir if support is enabled

* qt/Makefile.am: added

* qt/.cvsignore: added

* qt/dbus-qthread.cc, qt/dbus-qthread.cpp: renamed former to latter, added `#ifdef QT_THREAD_SUPPORT` guard.

* dbus/Makefile.am: added missing headers for make dist

2002-12-28 Kristian Rietveld <kris@gtk.org>

* dbus/Makefile.am: fixup `export-symbols-regex`.

2002-12-27 Anders Carlsson <andersca@codefactory.se>

* acinclude.m4: Add this file and put the
PKG_CHECK_MODULE macro in it.

2002-12-27 Anders Carlsson <andersca@codefactory.se>

* dbus/dbus-marshal.c: (_dbus_marshall_string),
(_dbus_demarshal_double), (_dbus_demarshal_int32),
(_dbus_demarshal_uint32), (_dbus_demarshal_string),
(_dbus_marshall_test):
Make the demarshalling routines align the pos argument.
Add string marshalling tests and fix the obvious bugs
discovered.

2002-12-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-auth.c: fixes fixes fixes

* dbus/dbus-transport-unix.c: wire up support for
encoding/decoding data on the wire

* dbus/dbus-auth.c (_dbus_auth_encode_data)
(_dbus_auth_decode_data): append to target string
instead of nuking it.

2002-12-26 Havoc Pennington <hp@pobox.com>

* dbus/dbus-marshal.h (DBUS_COMPILER_BYTE_ORDER): #ifdef
WORDS_BIGENDIAN then compiler byte order is DBUS_BIG_ENDIAN,
doh

* dbus/dbus-marshal.c: Add macros to do int swapping in-place and
avoid swap_bytes() overhead (ignoring possible assembly stuff for
now). Main point is because I wanted unpack_uint32 to implement
_dbus_verbose_bytes
(_dbus_verbose_bytes): new function

* dbus/dbus-string.c (_dbus_string_validate_ascii): new function

* dbus/dbus-message.c (_dbus_message_loader_get_is_corrupted):
add
mechanism to handle a corrupt message stream
(_dbus_message_loader_new): fix preallocation to only prealloc,
not prelengthen

* dbus/dbus-string.c (_dbus_string_skip_blank): fix this function
(_dbus_string_test): enhance tests for copy/move and fix the
functions

* dbus/dbus-transport-unix.c: Hold references in more places to
avoid reentrancy problems

- * dbus/dbus-transport.c: ditto
- * dbus/dbus-connection.c (dbus_connection_dispatch_message):
don't leak reference count in no-message case
- * test/watch.c (do_mainloop): handle adding/removing watches during iteration over the watches. Also, ref the connection/server stored on a watch, so we don't try to mangle a destroyed one.
- * dbus/dbus-transport-unix.c (do_authentication): perform authentication
- * dbus/dbus-auth.c (get_state): add a state AUTHENTICATED_WITH_UNUSED_BYTES and return it if required (_dbus_auth_get_unused_bytes): append the unused bytes to the passed in string, rather than prepend
- * dbus/dbus-transport.c (_dbus_transport_init_base): create the auth conversation DBusAuth
- * dbus/dbus-transport-unix.c (_dbus_transport_new_for_fd) (_dbus_transport_new_for_domain_socket): when creating a transport, pass in whether it's a client-side or server-side transport so we know which DBusAuth to create

2002-12-03 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-transport-unix.c (unix_finalize): finalize base _after_ finalizing the derived members (unix_connection_set): unref watch if we fail to add it
- * dbus/dbus-connection.c (dbus_connection_unref): delete the transport first, so that the connection owned by the transport will be valid as the transport finalizes.
- * dbus/dbus-transport-unix.c (unix_finalize): free the write_watch if necessary, and remove watches from the connection.
- * dbus/dbus-watch.c (_dbus_watch_list_free): improve a comment

2002-12-26 Anders Carlsson <andersca@codefactory.se>

- * dbus/dbus-marshal.c: (_dbus_marshal_string), (_dbus_demarshal_double), (_dbus_demarshal_int32), (_dbus_demarshal_uint32), (_dbus_demarshal_string), (_dbus_marshal_test):
- * dbus/dbus-marshal.h: Add string marshal functions and have the demarshal functions return the new position.

2002-12-25 Havoc Pennington <hp@pobox.com>

- * doc/dbus-sasl-profile.txt: docs on the authentication protocol, it is a simple protocol that just maps directly to SASL.

- * dbus/dbus-auth.h, dbus/dbus-auth.c: authentication protocol initial implementation, not actually used yet.

- * dbus/dbus-string.c (_dbus_string_find): new function
- (_dbus_string_equal): new function
- (_dbus_string_base64_encode): new function
- (_dbus_string_base64_decode): new function

2002-12-25 Anders Carlsson <andersca@codefactory.se>

- * dbus/Makefile.am:
- * dbus/dbus-marshal.c: (swap_bytes), (_dbus_marshall_double), (_dbus_marshall_int32), (_dbus_marshall_uint32), (_dbus_demarshall_double), (_dbus_demarshall_int32), (_dbus_demarshall_uint32), (_dbus_marshall_test):

- * dbus/dbus-marshal.h:

- * dbus/dbus-protocol.h:

- * dbus/dbus-test.c: (main):

- * dbus/dbus-test.h:

Add un-optimized marshalling/demarshalling routines.

2002-12-25 Harri Porten <porten@kde.org>

- * qt/dbus-qt.h: adjusted ctor and getter to KDE/Qt conventions

2002-12-24 Zack Rusin <zack@kde.org>

- * qt/dbus-qthread.cc: adding - integrates QMutex into Dbus

- * qt/dbus-qt.h: skeleton with two sample implemenatation of the main loop stuff

2002-12-24 Havoc Pennington <hp@pobox.com>

- * glib/dbus-gthread.c: fix include

- * glib/dbus-glib.h: rename DBusMessageHandler for now.

I think glib API needs to change, though, as you don't

want to use DBusMessageFunction, you want to use the

DBusMessageHandler object. Probably

dbus_connection_open_with_g_main_loop()

and dbus_connection_setup_g_main_loop() or something like that

(but think of better names...) that just create a connection

that has watch/timeout functions etc. already set up.

- * dbus/dbus-connection.c

- (dbus_connection_send_message_with_reply): new function just to

show how the message handler helps us deal with replies.

- * dbus/dbus-list.c (_dbus_list_remove_last): new function
- * dbus/dbus-string.c (_dbus_string_test): free a string that wasn't
- * dbus/dbus-hash.c: use memory pools for the hash entries (rebuild_table): be more paranoid about overflow, and shrink table when we can (_dbus_hash_test): reduce number of sprintfs and write valid C89. Add tests for case where we grow and then shrink the hash table.
- * dbus/dbus-mempool.h, dbus/dbus-mempool.c: memory pools
- * dbus/dbus-connection.c (dbus_connection_register_handler) (dbus_connection_unregister_handler): new functions
- * dbus/dbus-message.c (dbus_message_get_name): new
- * dbus/dbus-list.c: fix docs typo
- * dbus/dbus-message-handler.h, dbus/dbus-message-handler.c: an object representing a handler for messages.

2002-12-16 Anders Carlsson <andersca@codefactory.se>

- * glib/dbus-glib.h:
- * glib/dbus-gthread.c: (dbus_gthread_init):
Don't use the gdbus prefix for public functions.

2002-12-16 Anders Carlsson <andersca@codefactory.se>

- * Makefile.am:
- * configure.in:
Add GLib checks and fixup .pc files
- * glib/Makefile.am:
- * glib/dbus-glib.h:
- * glib/dbus-gmain.c: (gdbus_connection_prepare), (gdbus_connection_check), (gdbus_connection_dispatch), (gdbus_add_connection_watch), (gdbus_remove_connection_watch), (dbus_connection_gsource_new):
- * glib/dbus-gthread.c: (dbus_gmutex_new), (dbus_gmutex_free), (dbus_gmutex_lock), (dbus_gmutex_unlock), (dbus_gthread_init):
- * glib/test-dbus-glib.c: (message_handler), (main):
Add GLib support.

2002-12-15 Harri Porten <porten@kde.org>

- * autogen.sh: check for libtoolize before attempting to use it

- * dbus/dbus-transport-unix.c: include <sys/time.h> for timeval struct.
- * .cvsignore: ignore more stamp files
- * dbus/dbus-watch.c (_dbus_watch_list_new): fixed doc error
- * test/Makefile.am: added -I\$(top_srcdir) to be able to compile without make install.

2002-12-15 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-threads.c: add thread stubs that a higher library layer can fill in. e.g. the GLib wrapper might fill them in with GThread stuff. We still need to use this thread API to thread-safe-ize the library.

2002-12-12 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-transport-unix.c, dbus/dbus-server-unix.c: use the below new interfaces and include fewer system headers.
- * dbus/dbus-sysdeps.c (_dbus_read): new function
(_dbus_write): new function
(_dbus_write_two): new function
(_dbus_connect_unix_socket): new function
(_dbus_listen_unix_socket): new function
- * dbus/dbus-message-internal.h: change interfaces to use DBusString

2002-12-11 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-types.h: add dbus_unichar
- * dbus/dbus-internals.c (_dbus_verbose): use _dbus_getenv
- * dbus/dbus-connection.c (dbus_connection_send_message): return TRUE on success
- * dbus/dbus-transport.c: include dbus-watch.h
- * dbus/dbus-connection.c: include dbus-message-internal.h
- * HACKING: add file with coding guidelines stuff.
- * dbus/dbus-string.h, dbus/dbus-string.c: Encapsulate all string handling here, for security purposes (as in vsftpd). Not actually using this class yet.
- * dbus/dbus-sysdeps.h, dbus/dbus-sysdeps.c: Encapsulate all

system/libc usage here, as in vsftpd, for ease of auditing (and should also simplify portability). Haven't actually moved all the system/libc usage into here yet.

2002-11-25 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-internals.c (_dbus_verbose): fix to not always print the first verbose message.

2002-11-24 Havoc Pennington <hp@pobox.com>

- * test/echo-client.c, test/echo-server.c: cheesy test clients.

- * configure.in (AC_CHECK_FUNCS): check for writev

- * dbus/dbus-message.c (_dbus_message_get_network_data): new function

- * dbus/dbus-list.c (_dbus_list_foreach): new function

- * dbus/dbus-internals.c (_dbus_verbose): new function

- * dbus/dbus-server.c, dbus/dbus-server.h: public object representing a server that listens for connections.

- * dbus/.cvsignore: create

- * dbus/dbus-errors.h, dbus/dbus-errors.c: public API for reporting errors

- * dbus/dbus-connection.h, dbus/dbus-connection.c: public object representing a connection that sends/receives messages. (Same object used for both client and server.)

- * dbus/dbus-transport.h, dbus/dbus-transport.c: Basic abstraction for different kinds of stream that we might read/write messages from.

2002-11-23 Havoc Pennington <hp@pobox.com>

- * dbus/dbus-internals.h (_DBUS_INT_MAX): add _DBUS_INT_MIN _DBUS_INT_MAX

- * dbus/dbus-test.c (main): add list test, and include dbus-test.h as intended

- * dbus/dbus-hash.c (_dbus_hash_table_remove_string) (_dbus_hash_table_remove_int): return value indicates whether the entry existed to remove

* dbus/dbus-list.c: add linked list utility class,
with docs and tests

* dbus/dbus-hash.c: add TODO item about shrinking the hash bucket
array sometimes.

2002-11-23 Havoc Pennington <hp@pobox.com>

* Doxyfile.in (INCLUDE_FILE_PATTERNS): expand DBUS_BEGIN_DECLS/
DBUS_END_DECLS to nothing, that should fix this once and for all

* Doxyfile.in (JAVADOC_AUTOBRIEF): set to YES

* dbus/dbus-message.c, dbus/dbus-hash.c:
add some missing @brief

2002-11-23 Havoc Pennington <hp@pobox.com>

* dbus/dbus-message.h: put semicolons after DEBUG_BEGIN_DECLS
to avoid confusing Doxygen

* dbus/dbus-hash.c: @} not }@

* dbus/dbus-message.c (struct DbusMessage): split out
internals docs

2002-11-23 Havoc Pennington <hp@pobox.com>

* configure.in: pile on more warning flags if using gcc

* Doxyfile.in (EXTRACT_STATIC): set to NO, so we don't have
to document static functions

* configure.in: add summary to end of configure so it
looks nice and attractive

* dbus/dbus-hash.c: finish implementation and write unit
tests and docs

* configure.in: add --enable-tests to enable unit tests

* dbus/dbus-test.c: test program to run unit tests
for all files in dbus/*, initially runs a test for
dbus-hash.c

* dbus/dbus-internals.h: file to hold some internal utility stuff

2002-11-22 Havoc Pennington <hp@redhat.com>

* dbus/dbus-hash.c: copy in Tcl hash table, not yet
"ported" away from Tcl

* dbus/dbus-types.h: header for types such as dbus_bool_t

2002-11-22 Havoc Pennington <hp@redhat.com>

* dbus/dbus.h: fixups for doc warnings

* Doxyfile.in (FILE_PATTERNS): we need to scan .h to pick up macros
(QUIET): make it quiet so we can see warnings

* dbus/dbus-memory.c: teach D-BUS to allocate and free memory

2002-11-22 Havoc Pennington <hp@redhat.com>

* Makefile.am: include "Doxyfile" target in all-local

* configure.in: generate the Doxyfile

* Doxyfile.in: move Doxyfile here, so we can use configure to generate a Doxyfile with the right version number etc.

2002-11-22 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c: move inline docs into .c file

* Doxyfile (OUTPUT_DIRECTORY): move output to doc/api so all docs are under doc/
(MAN_EXTENSION): generate man pages. Use extension ".3dbus" which matches ".3qt" on my system, I guess this is OK, I don't know really.
(FILE_PATTERNS): look for .c files not .h, makes sense for plain C I think

2002-11-22 Havoc Pennington <hp@pobox.com>

* Makefile.am (SUBDIRS): rename subdir "server" to "bus" because any app can be a server, and any app can be a client, the bus is a special kind of server.

Thu Nov 21 23:35:31 2002 Zack Rusin <zack@kde.org>

* Doxyfile : adding. Still needs Makefile rules to be generated automatically (just run "doxygen" in the toplevel dir for now to generate docs)

* dbus/dbus-message.h : Adding sample docs (javadoc since resembles gtk-doc a little more)

* dbus/dbus.h : Adding sample docs

2002-11-21 Havoc Pennington <hp@redhat.com>

- * dbus/Makefile.am (INCLUDES): define DBUS_COMPILATION so we can allow ourselves to include files directly, instead of having to use dbus.h
- * dbus/dbus.h: fill in
- * dbus/dbus-message.h: sketch out a sample header file. Include griping if you include it directly instead of via dbus.h
- * dbus/dbus-macros.h: new file with macros for extern "C", TRUE/FALSE, NULL, etc.
- * doc/file-boilerplate.c: put include guards in here

2002-11-21 Havoc Pennington <hp@redhat.com>

- * doc/file-boilerplate.c: include both AFL and GPL boilerplate.
- * COPYING: include the GPL as well, and license code under both AFL and GPL.

2002-11-21 Havoc Pennington <hp@redhat.com>

- * acconfig.h: get rid of this
- * autogen.sh (run_configure): add --no-configure option
- * configure.in: remove AC_ARG_PROGRAM to make autoconf complain less. add AC_PREREQ. add AC_DEFINE third arg.

2002-11-21 Anders Carlsson <andersca@codefactory.se>

- * doc/Makefile.am: Fix references so we can distcheck.

2002-11-21 Havoc Pennington <hp@redhat.com>

- * Initial module creation

File = ChangeLog.pre-1-2

2008-04-04 John (J5) Palmieri <johnp@redhat.com>

- * Released 1.2.1

2008-04-03 John (J5) Palmieri <johnp@redhat.com>

Patch from Sumit <sumitskj_20@yahoo.com>, comments added

* dbus/dbus-transport.c(_dbus_transport_open): fix mem leak

2008-04-03 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-connection.c (dbus_connection_send): add documentation to describe when to call dbus_connection_flush and dbus_connection_unref after a call to dbus_connection_send is made
Initial wording by Stanislav Brabec <sbrabec at suse.cz> (fd.o bug#13558)

2008-04-03 John (J5) Palmieri <johnp@redhat.com>

Patch from Kimmo Hämmäläinen <kimmo.hamalainen at nokia.com>

* bus/expirelist.c (do_expiration_with_current_time): calculate correct min wait time and next interval (bus_expire_list_add, bus_expire_list_add_link): if the timeout is disabled when we add an item to the expire list, enable the timeout (do_expiration_with_current_time): only set timeout if there are items to expire

2008-04-01 Timo Hoenig <thoenig@suse.de>

Patch from Frederic Crozat <fcrozat@mandriva.com>

* bus/dir-watch-inotify.c (bus_watch_directory): Only monitor IN_CLOSE_WRITE, IN_DELETE, IN_MOVE_TO and IN_MOVE_FROM events. This way, only atomic changes to configuration file are monitored.
* bus/dir-watch-inotify.c (_handle_inotify_watch): Fix typo in _dbus_verbose function call
* bus/dir-watch-inotify.c (bus_drop_all_directory_watches): Use _dbus_strerror instead of perror

2008-03-04 Havoc Pennington <hp@redhat.com>

* bus/connection.c, bus/expirelist.c: Make the BusExpireList struct opaque, adding accessors for manipulating the list. In this commit there should be no change in functionality or behavior. The

purpose of this change is to improve encapsulation prior to fixing some bugs Kimmo Härmäläinen found where the timeout is not properly updated, since we need to e.g. take some action whenever adding and removing stuff from the expire list.

2008-03-31 Colin Walters <walters@verbum.org>

Patch from Owen Taylor <otaylor@redhat.com>

* tools/dbus-launch-x11.c: Check for X11 events before selecting (FDO bug #15293)

2008-03-31 Colin Walters <walters@verbum.org>

Patch from Owen Taylor <otaylor@redhat.com>

* tools/dbus-launch-x11.c: Make sure we call XFlush() on all code paths (FDO bug #15293)

2008-03-27 Havoc Pennington <hp@redhat.com>

* tools/dbus-send.c (append_dict): Do not provide a signature to dbus_message_iter_open_container() when opening a dict entry.

2008-03-26 Colin Walters <walters@verbum.org>

Patch from Scott James Remnant <scott@netsplit.com>

* dbus/dbus-bus.c: Set default exit_on_disconnect after registration with the bus, not before. This ensures that programs which wish to set exit_on_disconnect to FALSE will not be terminated if the bus exits during registration. (FDO Bug #15112)

2008-03-04 John (J5) Palmieri <johnp@redhat.com>

* fix broken poll on Mac OSX - build patch by Benjamin Reed
* configure.in: check for OSX's deadlocking poll
* dbus/dbus-sysdeps-unix.c (_dbus_poll): if we have a broken poll don't use poll

2008-03-04 John (J5) Palmieri <johnp@redhat.com>

* check if the linker supports a flag instead of just checking for GNU ld
* configure.in: move AM_PROG_LIBTOOL to the top

(ld_supports_flag): new function for checking if the linker supports a given flag

2008-03-04 John (J5) Palmieri <johnp@redhat.com>

* add a changelog for Benjamin Reed's git patch RANT: Change Logs are handled by git and having an external changelog just screws up merging. We should write down rules for doing git commit messages and leave it at that.

* configure.in: Platform build fixes for Mac OS X the Darwin linker does not understand the -z option; wrap it in a check for \$with_gnu_ld. environ is only available at runtime, so you need to make a reference to _NSGetEnviron instead for symbols to resolve properly.

2008-03-04 John (J5) Palmieri <johnp@redhat.com>

* configure.in: add \$THREAD_LIBS to DBUS_LAUNCHER_LIBS so we link correctly

2008-03-04 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-launch.c: wrap X'ism in #ifdef so we can compile without X

2008-02-28 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-sysdeps-unix.c: define _AI_ADDRCONFIG as 0 if not defined so that we can compile with an older glibc

2008-02-26 John (J5) Palmieri <johnp@redhat.com>

* Released 1.1.20

2008-02-26 John (J5) Palmieri <johnp@redhat.com>

* CVE-2008-0595 - security policy of the type <allow send_interface="some.interface.WithMethods"/> work as an implicit allow for messages sent without an interface bypassing the default deny rules and potentially allowing restricted methods exported on the bus to be

executed by unauthorized users. This patch fixes the issue.

* bus/policy.c (bus_client_policy_check_can_send, bus_client_policy_check_can_receive): skip messages without an interface when evaluating an allow rule, and thus pass it to the

default deny rules

2008-02-26 John (J5) Palmieri <johnp@redhat.com>

- * correctly unref connections without guids during shutdown
- * dbus/dbus-connection.c (close_connection_on_shutdown): new method
- split out from shared_connections_shutdown
- (shared_connections_shutdown): shutdown all shared connections without guids
- (_dbus_connection_ref_unlocked): handle OOM when prepending no guid
- connections to the shared_connections_no_guid list
- * Patch by Kimmo Hämmäläinen <kimmo dot hamalainen at nokia dot com>

2008-02-21 John (J5) Palmieri <johnp@redhat.com>

- * fix build against the latest gcc/glibc
- * dbus/dbus-sysdeps-unix.c: define _GNU_SOURCE
- * bus/selinux.c: include limits.h
- * Patch by Matthias Clasen <mclasen at redhat.com>

2008-02-21 John (J5) Palmieri <johnp@redhat.com>

- * fixes dbus-launch so the bus goes away when X does (Red Hat Bug #430412)
- * tools/dbus-launch.c (main): set xdisplay = NULL
- * Patch by Matthias Clasen <mclasen at redhat.com>

2008-01-17 John (J5) Palmieri <johnp@redhat.com>

- * Released 1.1.4

2008-01-17 Timo Hoenig <thoenig@suse.de>

- * fix inotify support
- * bus/dir-watch-inotify.c (_handle_inotify_watch): fix reading of the inotify events. Also, use ssize_t not size_t for 'ret'.
- * bus/dir-watch-inotify.c (bus_watch_directory): watch not only for IN_MODIFY but also for IN_CREATE and IN_DELETE
- * bus/dir-watch-inotify.c (bus_drop_all_directory_watches): drop the inotify watches more elegantly by closing inotify:_fd, set inotify_fd to -1 after dropping the watches

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

- * configure.in: post-release version bump

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* Released 1.1.3 (1.2.0RC1)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* fix hacking to say git instead of cvs

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by SÃ©bastien Couret <10function at gmail dot com>

* dbus/dbus-marshall-recursive.c (all_reader_classes[]): wrap in #ifndef DBUS_DISABLE_ASSERT since it is only used in asserts which are noop

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Magnus Henoch <henoch plus bfdo at dtek dot chalmers dot se>

* dbus/dbus-auth.c (handle_server_data_external_mech): handle SASL EXTERNAL's initial empty response (FDO Bug #9945)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* bus/messagebus.in: add lsb headers (FDO Bug #11491)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Peter O'Gorman <pogma at thewrittenword dot com>

* dbus/dbus-spawn.c (babysit_signal_handler): check write return value so we don't hang (FDO Bug #11665)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Peter O'Gorman <pogma at thewrittenword dot com>

* dbus/dbus-sysdeps.h: support for AIX poll implementation (FDO Bug #11666)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* tests/name-test/run-test.sh: make more portable (FDO Bug #11667)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-connection.c
(_dbus_connection_get_next_client_serial):
don't check for < 0 on an unsigned variable (FDO Bug #12924)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* bus/bus.c (setup_server): check failed allocation (FDO Bug #12920)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-spawn.c (_dbus_spawn_async_with_babysitter): the API contract says sitter_p can be NULL, so let's check it (FDO Bug #12919)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-spawn.c (read_ints, read_pid): use correct ssize_t type instead of size_t (FDO Bug #12862)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-errors.c (dbus_set_error): make sure to call va_end if we hit an OOM error inside va_start (FDO Bug #12846)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-connection.c (dbus_connection_send_with_reply): fix possible crash if pending_return is NULL (FDO Bug #12673)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* portions of patch submitted by Tim Mooney
<enchanter at users dot sourceforge dot net>

* configure.in: never auto-select libxml (FDO Bug #12479)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patches by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-sysdeps-unix (_dbus_get_autolaunch_address): handle OOM
(FDO Bug #12945)

* dbus/dbus-uuidgen.c (return_uuid): handle OOM (FDO Bug #12928)

* dbus/dbus-misc.c (dbus_get_local_machine_id): handle OOM, fix return value to return NULL not FALSE (FDO Bug #12946)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* bus/bus.c (bus_context_check_security_policy): rewrite selinux error handling to not abort due to a NULL read and to set the error only if it is not already set (Based off of FDO Bug #12430)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-internals.c (_dbus_read_uuid_file_without_creating, _dbus_create_uuid_file_exclusively): add OOM handling (FDO Bug #12952)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-spawn.c (babysit, babysitter_iteration): add error handling when polling (FDO Bug #12954)

2008-01-15 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* bus/config-parser.c (locate_attributes): remove dead code which

always evaluated to TRUE

* dbus/dbus-shell.c (_dbus_shell_quote): remove unused code

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* bus/connection.c (bus_connection_complete): plug a possible BusClientPolicy leak (FDO Bug #13242)

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* patch by Frederic Crozat <fcrozat at mandriva dot com> (FDO Bz# 13268)

* add inotify support

* bus/Makefile.am: add inotify module to the build

* bus/dir-watch-inotify.c: inotify module based off the dnotify and kqueue modules

* configure.in: add checks and switch for inotify also add a printout at the end of configure if inotify and kqueue support is being built in (dnotify already had this)

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* patch by Frederic Crozat <fcrozat at mandriva dot com>

* bus/dir-watch-dnotify.c (bus_watch_directory): watch for file creates also

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* patch by Kimmo Härmäläinen <kimmo dot hamalainen at nokia dot com>

* dbus/dbus-transport-socket.c(do_reading): return message loader buffer in case of OOM (FDO Bug#12666)

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* configure.in: add warning to output when libxml is selected since we don't have a libxml maintainer and expat works perfectly fine for what we need an xml parser for

2008-01-14 John (J5) Palmieri <johnp@redhat.com>

* Patch by Andrea Luzzardi <scox at sig11 dot org>: creates a `_dbus_geteuid` function to fix EXTERNAL authentication in setuid applications

* `dbus/dbus-sysdeps-unix.c` (`_dbus_geteuid`): used to get the effective uid of the running program
(`_dbus_credentials_add_from_current_process`): use `geteuid` instead of `getuid`
(`_dbus_append_user_from_current_process`): use `geteuid` instead of `getuid`

* `dbus/dbus-sysdeps-util-unix.c` (`_dbus_change_to_daemon_user`): use `geteuid` instead of `getuid`
(`_dbus_unix_user_is_at_console`): use `geteuid` instead of `getuid`

* `dbus/dbus-sysdeps-win.c` (`_dbus_geteuid`): add a windows equivalent that returns `DBUS_UID_UNSET`

2007-12-18 Havoc Pennington <hp@redhat.com>

* `dbus/dbus-connection.c` (`_dbus_connection_block_pending_call`): fix location of curly braces

2007-11-23 Sjoerd Simons <sjoerd@luon.net>

* `tools/dbus-launch.c`: let both a normal `dbus-launch` and an autolaunched bus save their parameters in X11 if possible. This makes the autolaunch and non-autolaunch behaviour more similar. With the exception that on a normal launch there will always be a new session bus and not being able to save parameters is not fatal. This also enables to launch programs directly with autolaunch (not very usefull though).

2007-10-31 Havoc Pennington <hp@redhat.com>

* `bus/selinux.c` (`log_audit_callback`): rewrite to use `_dbus_string_copy_to_buffer_with_nul()`

* `dbus/dbus-string.c` (`_dbus_string_copy_to_buffer`): change to NOT nul-terminate the buffer; fail an assertion if there is not enough space in the target buffer. This fixes two bugs where `copy_to_buffer` was used to copy the binary bytes in a UUID, where

nul termination did not make sense. Bug reported by David Castelow.

(dbus_string_copy_to_buffer_with_nul): new function that always nul-terminates the buffer, and fails an assertion if there is not enough space in the buffer.

2007-10-23 Havoc Pennington <hp@redhat.com>

* bus/bus.c (bus_context_new): use the new name here

* bus/selinux.c (bus_selinux_audit_init): rename from audit_init() to avoid possible libc conflict, and declare it in .h file to avoid a warning

2007-10-19 Havoc Pennington <hp@redhat.com>

* bus/bus.c (bus_context_new): put audit_init() in HAVE_SELINUX

2007-10-19 Havoc Pennington <hp@redhat.com>

* bus/bus.c (bus_context_new): put the audit_init() in here instead, which I believe ends up being the same as where it was before, though I'm not sure I understand why it goes here.

* dbus/dbus-sysdeps-util-unix.c (dbus_change_to_daemon_user): remove audit_init() from here, this file can't depend on code in bus/ directory

2007-10-16 Simon McVittie <simon.mcvittie@collabora.co.uk>

* configure.in: *Actually* fix detection of i486 atomic ops - my previous attempt at a fix would always enable them due to wrong quoting. Patch from Colin Walters <walters@verbum.org>

2007-10-11 Simon McVittie <simon.mcvittie@collabora.co.uk>

* configure.in: enable Autoconf's AC_C_INLINE to avoid compilation failure with gcc -ansi
* dbus/dbus-macros.h, dbus/dbus-arch-deps.h.in: Use new macro _DBUS_GNUC_EXTENSION (the same as G_GNUC_EXTENSION) to avoid -ansi warnings about use of "long long".
* dbus/dbus-server-socket.c: remove unused variable when assertions are disabled
* dbus/dbus-marshall-validate.c: avoid empty statements by removing stray semicolons

```
* tools/dbus-launch.c: convert C++-style comment to C-style, add
{}
for clarity
* .gitignore: ignore vi swapfiles
* dbus/dbus-errors.h, dbus/dbus-errors.c: Add DBUS_ERROR_INIT
macro,
equivalent to calling dbus_error_init() on an uninitialized
DBusError
* dbus/dbus-address.c, dbus/dbus-auth-script.c, dbus/dbus-auth-
util.c,
dbus/dbus-connection.c, dbus/dbus-internals.c, dbus/dbus-
keyring.c,
dbus/dbus-message-util.c, dbus/dbus-server.c, dbus/dbus-sha.c,
dbus/dbus-spawn-win.c, dbus/dbus-spawn.c,
dbus/dbus-sysdeps-util-win.c, dbus/dbus-transport-socket.c,
dbus/dbus-transport.c, dbus/dbus-userdb.c: use that macro instead
of
calling dbus_error_init() where it's clearly equivalent
* configure.in, dbus/dbus-sysdeps.h, dbus/dbus-sysdeps-unix.c:
Fix
detection of i486 atomic ops. Previously, the attempts to
determine
support at compile-time on Darwin were causing the i486 atomic
ops to
be used on *all* i386 or x86-64 GCC builds (AH_VERBATIM can't be
conditionalized like we were trying to).
```

2007-10-10 Simon McVittie <simon.mcvittie@collabora.co.uk>

```
* dbus/dbus-errors.c, dbus/dbus-protocol.h: Add new error
org.freedesktop.DBus.Error.ObjectPathInUse
* dbus/dbus-object-tree.h, dbus/dbus-object-tree.c,
dbus/dbus-connection.c, dbus/dbus-connection.h: add new functions
dbus_connection_try_register_object_path and
dbus_connection_try_register_fallback, which raise
ObjectPathInUse
rather than asserting, to make object path registration less
painful
for bindings
* .gitignore: add various things that weren't in .cvsignore
because
CVS implicitly ignored them; generally bring up to date
```

2007-10-09 John (J5) Palmieri <johnp@redhat.com>

```
* tools/run-with-tmp-session-bus.sh: Fix env exports for better
portability (#9280)
* tools/dbus-send.1: Document syntax for container types in dbus-
send
man file (#9553) - patch from Jack Spaar <jspaar at
users.sourceforge.net>
```

[Both OK for MIT/X11 relicensing -smcv]

2007-10-09 Simon McVittie <simon.mcvittie@collabora.co.uk>

- * doc/dbus-specification.xml: Specifically forbid empty structs (#7969)
- * doc/dbus-specification.xml: Patches from Kristoffer Lund to clarify description of DBUS_COOKIE_SHA1 (#10184) and allowable contents of a variant (#10185, amended as per Havoc's comments)

[All of the above are OK for MIT/X11 licensing]

2007-10-03 John (J5) Palmieri <johnp@redhat.com>

- * dbus/dbus-internals.h: fd.o bug #11678 Don't error out if compiler does not support vararg macros. `_dbus_verbose` is the only function that does this so make it a noop if vararg macros are not supported
- * bus/selinux.c, dbus/dbus-sysdeps-util-unix.c: fd.o bug #12429 Reverse check to setpcap and only init audit if we were root (patch by Dan Walsh <dwalsh@redhat.com>, https://bugs.freedesktop.org/show_bug.cgi?id=12429). Reverse `we_were_root` check to setpcap if we were root. Also only init audit if we were root. So error dbus message will not show up when policy reload happens. `dbus -session` will no longer try to send audit message, only system will.
- * configure.in: fd.o bug #11872 improve linker test for `--gc-sections`. Patch by Tim Mooney <enchanter at users.sourceforge.net>
- * configure.in, dbus/dbus-sysdeps.c: fd.o bug #11872 fix `clearenv` for systems that do not have it. Patch from Brian Cameron <brian.cameron at sun.com>
- * tools/dbus-launch.c: fd.o bug #12547 remove superfluous `if`. Also convert tabs to spaces
- * configure.in, bus/Makefile.am, dbus/Makefile.am: Correctly implement `-fPIC` and `-fPIE`. For security reasons we want position independent code for libraries and position independent executable for executables. Before we were just enabling `-fPIC`. Now we correctly enable `-fPIC` and `-PIE` for `libdbus` and the bus respectively.

Proper `LD_FLAGS` are set for each also.

2007-09-20 Ryan Lortie <desrt@desrt.ca>

Add argument path matching support. Bug #11066.

* dbus/signals.c (struct DBusMatchRule, bus_match_rule_new, bus_match_rule_set_arg, bus_match_rule_parse_arg_match, match_rule_matches): Add support for parsing and matching on arg0path='/some/path' type rules.

* dbus/signals.h (bus_match_rule_set_arg): change to take const DBusString instead of const char * for the string to match against.

* dbus/dbus-bus.c: add a quick note to dbus_bus_add_match documentation about the path matching.

* doc/dbus-specification.xml: add a more detailed description of the changes here.

2007-09-19 Ryan Lortie <desrt@desrt.ca>

Add support for compacting DBusStrings to release wasted memory.

* dbus/dbus-string.[ch] (compact, _dbus_string_compact, _dbus_string_lock): new compact function to free up allocated memory that is no longer used.

* dbus/dbus-message.c (load_message): call _dbus_string_compact on the message loader buffer.

* dbus/dbus-transport-socket.c (do_reading, do_writing): call _dbus_string_compact on the incoming/outgoing "encoded" buffers.

* dbus/dbus-string-util.c (_dbus_string_test): add a few tests for string compacting.

2007-09-13 Ryan Lortie <desrt@desrt.ca>

* HACKING: add more explicit git branch/tag instructions

2007-09-13 Ryan Lortie <desrt@desrt.ca>

migrate from cvs to git (cvs2svn -> git-svnimport).

* HACKING: update release/branch/tag instructions
* */.cvsignore: rename to .gitignore

also, clean up tags and branch names to conform to HACKING

2007-08-17 William Jon McCann <mccann@jhu.edu>

* update-dbus-docs.sh: upload DTD to server

2007-08-17 Havoc Pennington <hp@redhat.com>

* tools/dbus-launch-x11.c (set_address_in_x11): fix from Michael Lorenz to use long not int with XChangeProperty format 32

* dbus/dbus-sysdeps-util-unix.c
(_dbus_write_pid_to_file_and_pipe): factor this out, and use the same code in _dbus_become_daemon (where the parent writes the pid file and to the pid pipe) and in bus_context_new (where the daemon writes its own pid file and to its own pid pipe)

* bus/bus.c (bus_context_new): close the pid pipe after we print to it. Also, don't write the pid to the pipe twice when we fork, someone reported this bug a long time ago.

2007-08-03 Havoc Pennington <hp@redhat.com>

* configure.in: add major/minor/micro version number AC_SUBST

* dbus/dbus-arch-deps.h.in (DBUS_MAJOR_VERSION, DBUS_MINOR_VERSION, DBUS_MICRO_VERSION, DBUS_VERSION_STRING, DBUS_VERSION): collection of macros to get version of library we are compiled against.

* dbus/dbus-misc.c (dbus_get_version): new function, to get version of library we are linked against at runtime.

2007-07-30 Havoc Pennington <hp@redhat.com>

* bus/activation-helper.c (check_bus_name): don't use _dbus_check_valid_bus_name() which is only around with --enable-checks, instead use _dbus_validate_bus_name(). Bug #11766 from Diego <diego@pemas.net>

2007-07-27 Havoc Pennington <hp@redhat.com>

* configure.in: post-release version bump

2007-07-27 Havoc Pennington <hp@redhat.com>

* release 1.1.2

2007-07-26 Havoc Pennington <hp@redhat.com>

* bus/config-parser-trivial.c (check_return_values): disable a test that hardcoded the bus user's name

* bus/dispatch.c (bus_dispatch_test_conf): remove the "if (!use_launcher)" around the tests, they were only failing because we didn't pass through all the expected errors from the helper.

* bus/activation-exit-codes.h
(BUS_SPAWN_EXIT_CODE_CHILD_SIGNALED): add a code for child segfaulting
(BUS_SPAWN_EXIT_CODE_GENERIC_FAILURE): make "1" be a generic failure code, so if a third party launch helper were written it could just always return 1 on failure.

2007-07-24 Daniel P. Berrange <dan@berrange.com>

* bus/dbus-daemon.1: Add docs on new syntax options for the bus address strings

* dbus/dbus-address.c: Allow * in addresses (for binding to all addresses).

* dbus/dbus-sysdeps.h:

* dbus/dbus-sysdeps-unix.c: Re-write to use getaddrinfo instead of gethostbyname to enable protocol independant name lookup, making IPv6 work

* dbus/dbus-server-socket.h:

* dbus/dbus-server-socket.c: Add support for 'family' in the address string to specify ipv4 vs ipv6. Use a port string to allow for service resolution. Allow for binding to multiple sockets at once in case of dual IPv4 & IPv6 stacks.

* dbus/dbus-server-unix.c: Pass in an array of file descriptors instead of a single one.

* dbus/dbus-transport-socket.h:

* dbus/dbus-transport-socket.c: Add support for 'family' in the address string to specify ipv4 vs ipv6. Use a port string to allow for service resolution.

2007-07-24 Havoc Pennington <hp@redhat.com>

* configure.in: add AM_PROG_CC_C_O to allow per-target CPPFLAGS

* bus/dispatch.c (bus_dispatch_test_conf): Fix up setting TEST_LAUNCH_HELPER_CONFIG to include the full path, and enable test_shell_fail_service_auto_start when use_launcher==TRUE

* bus/activation-helper-bin.c (convert_error_to_exit_code): pass through the INVALID_ARGS error so the test suite works

* bus/activation.c (handle_activation_exit_error): return DBUS_ERROR_NO_MEMORY if we get BUS_SPAWN_EXIT_CODE_NO_MEMORY

* dbus/dbus-spawn.c (_dbus_babysitter_get_child_exit_status):
return only the exit code of the child, not the entire thingy
from
waitpid(), and make the return value indicate whether the child
exited normally (with a status code)

* bus/bus.c (process_config_first_time_only): _dbus_strdup works
on NULL so no need to check
(process_config_every_time): move servicehelper init here, so we
reload it on HUP or config file change

* bus/Makefile.am (install-data-hook): remove comment because
Emacs make mode seems to be grumpy about it

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/Makefile.am:
* bus/test-system.c: (die), (check_memleaks), (test_pre_hook),
(test_post_hook), (main):
Add back the test-system.c file - not sure now this got ignored
in the
diff. I blame git.

2007-07-24 Richard Hughes <richard@hughsie.com>

* configure.in:
Use ustar to generate the tarball; this fixes the make distcheck
problem when the data files do not fit in the archive:
tar: dbus-1.1.2/test/data/valid-service-
files/org.freedesktop.DBus.
TestSuiteShellEchoServiceFail.service.in: file name is too
long (max 99); not dumped

We have to have the 'long' names as the service helper matches by
filename rather than by the name in the service file.

2007-07-24 Richard Hughes <richard@hughsie.com>

* configure.in:
* test/Makefile.am:
* test/data/invalid-service-files-
system/org.freedesktop.DBus.TestS
uiteNoExec.service.in:
* test/data/invalid-service-files-
system/org.freedesktop.DBus.TestS
uiteNoService.service.in:
* test/data/invalid-service-files-
system/org.freedesktop.DBus.TestS
uiteNoUser.service.in:
* test/data/valid-config-files-system/debug-allow-all-
fail.conf.in:

```
* test/data/valid-config-files-system/debug-allow-all-
pass.conf.in:
* test/data/valid-config-files/debug-allow-all-sha1.conf.in:
* test/data/valid-config-files/debug-allow-all.conf.in:
* test/data/valid-service-files-
system/org.freedesktop.DBus.TestSui
teEchoService.service.in:
* test/data/valid-service-files-
system/org.freedesktop.DBus.TestSui
teSegfaultService.service.in:
* test/data/valid-service-files-
system/org.freedesktop.DBus.TestSui
teShellEchoServiceFail.service.in:
* test/data/valid-service-files-
system/org.freedesktop.DBus.TestSui
teShellEchoServiceSuccess.service.in:
* test/data/valid-service-files/debug-echo.service.in:
* test/data/valid-service-files/debug-segfault.service.in:
* test/data/valid-service-files/debug-shell-echo-fail.service.in:
* test/data/valid-service-files/debug-shell-echo-
success.service.in:
* test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoS
ervice.service.in:
* test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfa
ultService.service.in:
* test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShell
EchoServiceFail.service.in:
* test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShell
EchoServiceSuccess.service.in:
Add the data files needed by the system activation unit checks.
```

2007-07-24 Richard Hughes <richard@hughsie.com>

```
* bus/dispatch.c: (check_segfault_service_no_auto_start),
(check_launch_service_file_missing),
(check_launch_service_user_missing),
(check_launch_service_exec_missing),
(check_launch_service_service_missing), (bus_dispatch_test_conf),
(bus_dispatch_test_conf_fail), (bus_dispatch_test):
Add unit tests for system activation. Most are copied from the
session activation tests, but some didn't apply when using a
launcher.
```

2007-07-24 Richard Hughes <richard@hughsie.com>

```
* bus/activation.c: (bus_activation_activate_service):
If the bus uses a service-launcher, then use the setuid launcher.
```


2007-07-24 Richard Hughes <richard@hughsie.com>

* configure.in:
Add the needed library exports for the new laucher.

2007-07-24 Richard Hughes <richard@hughsie.com>

* configure.in:
Check for `-Wl,--gc-sections` so we can really reduce the size of
the
setuid binary.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/activation.c: (handle_activation_exit_error),
(babysitter_watch_callback):
Map the child exit status integer to a proper dbus error.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/bus.c: (process_config_first_time_only),
(process_config_every_time), (bus_context_unref),
(bus_context_get_servicehelper):
* bus/bus.h:
Add the concept of a service-helper and allow it's value to be
read.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/activation.c: (bus_activation_entry_unref),
(update_desktop_file_entry):
Add the concept of, and read the value of user from the desktop
file.
The user string is not required unless we are using system
activation.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/activation.c:
* bus/desktop-file.h:
Move the defines into the header file, as we use these in the
lauch
helper as well as the desktop file parsing.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/.cvsignore:
Add the autogenerated binary files.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/Makefile.am:

* bus/test.h:
Add the build glue for the launch helper, and also add the launch-helper
OOM checks into make check. I've probably broken the build, give me 2.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/test-launch-helper.c: (die), (check_memleaks),
(test_post_hook), (bus_activation_helper_oom_test), (main):
Add a test wrapper to allow OOM checks on the launch helper.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/activation-helper-bin.c: (convert_error_to_exit_code),
(main):
* bus/activation-helper.c: (desktop_file_for_name),
(clear_environment), (check_permissions), (check_service_name),
(get_parameters_for_service), (switch_user),
(exec_for_correct_user), (check_bus_name), (get_correct_parser),
(launch_bus_name), (check_dbus_user), (run_launch_helper):
* bus/activation-helper.h:
Add the initial launch-helper. This is split into a main section
and a
binary loader that allows us to launch the main section in another
test
harness to do stuff like OOM testing. No build glue yet.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/Makefile.am:
* bus/config-parser.c: (bus_config_parser_unref),
(start_busconfig_child), (bus_config_parser_end_element),
(servicehelper_path), (bus_config_parser_content),
(bus_config_parser_finished),
(bus_config_parser_get_servicehelper),
(test_default_session_servicedirs),
(test_default_system_servicedirs), (bus_config_parser_test):
* bus/config-parser.h:
Make the config-parser code use the common config code.
Also add the session and systemd dirs stuff, and make the config
parser
aware of the servicehelper field.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/system.conf.in:
Add new servicehelper fields to the default system.conf file.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/config-parser-trivial.c: (service_dirs_find_dir),

```
(service_dirs_append_link_unique_or_free),
(bus_config_parser_new),
  (bus_config_parser_unref), (bus_config_parser_start_element),
  (bus_config_parser_end_element), (bus_config_parser_content),
  (bus_config_parser_finished), (bus_config_parser_get_user),
  (bus_config_parser_get_type),
(bus_config_parser_get_service_dirs),
  (check_return_values), (do_load), (check_loader_oom_func),
  (process_test_valid_subdir), (make_full_path),
(check_file_valid),
  (bus_config_parser_trivial_test):
* bus/config-parser-trivial.h:
  Add a security sensitive stripped down config parser for the
setuid
  launcher. This file only reads what it needs, and doesn't try to
do
  anything remotely clever like including external files.
  It is not intended to validate the config file; it is expected
that
  config-parser will do that before the setuid program tries to
read it.
```

2007-07-24 Richard Hughes <richard@hughsie.com>

```
* bus/config-parser-common.c:
  (bus_config_parser_element_name_to_type),
  (bus_config_parser_element_type_to_name):
* bus/config-parser-common.h:
  We don't want to run the whole config parser with all it's deps
in the
  setuid program. We need to implement a stripped down config
parser just
  for the launcher, and to do so I need some common functions and
defines; add them here.
```

2007-07-24 Richard Hughes <richard@hughsie.com>

```
* dbus/dbus-sysdeps-unix.c:
  (_dbus_get_standard_system_servicedirs):
* dbus/dbus-sysdeps-win.c:
  Provide a way to get the standard system servicedirs, just like
we do
  for the session service dirs. These should be separate, as there
may
  be a security issue starting up some session stuff as root.
  The use-case for the same binary starting up per-system_and_
per-session is also not valid.
```

2007-07-24 Richard Hughes <richard@hughsie.com>

```
* bus/dbus-daemon.1.in:
```

Add standard_system_servicedirs and servicehelper into the man file and explain what each does.

2007-07-24 Richard Hughes <richard@hughsie.com>

* doc/busconfig.dtd:
Add servicehelper into the dtd, it will soon be a valid part of the config file.

2007-07-24 Richard Hughes <richard@hughsie.com>

* dbus/dbus-spawn.c: (read_data),
(_dbus_babysitter_get_child_exit_status):
* dbus/dbus-spawn.h:
Add a function so we can get access to the exit status of the launch helper.
By providing the return code and not the error we can leave the 'what does this mean?' to the bus launch code and not include it in the dbus directory.

2007-07-24 Richard Hughes <richard@hughsie.com>

* bus/activation-exit-codes.h:
Add defines which specify the output codes of the launch helper.
We have to use exit codes as this is the only way we can return failure type without going grotty things like redirecting possibly-nonsecure stderr into the error.

2007-07-24 Richard Hughes <richard@hughsie.com>

* dbus/dbus-protocol.h:
Add new error names needed for the launch helper.

2007-07-24 Richard Hughes <richard@hughsie.com>

* dbus/dbus-sysdeps.c: (_dbus_clearenv):
* dbus/dbus-sysdeps.h:
Add a wrapper for clearenv.

2007-07-24 Richard Hughes <richard@hughsie.com>

* doc/system-activation.txt:
Add design document for the system activation parts. I'll shortly be committing many patches that add system activation using a setuid launcher into CVS, so expect things to be broken for a few hours.

2007-07-19 Ralf Habacker <ralf.habacker@freenet.de>

- * cmake/modules/FindKDEWIN.cmake: fixed comment
- * cmake/modules/FindKDEWIN32.cmake: removed obsolete cmake module

2007-07-18 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-message.c (dbus_message_get_cached) (dbus_message_cache_or_finalize): don't mess with message from message cache outside of the cache lock. Bug #9164 from Jonathan Matthew.

2007-07-13 Havoc Pennington <hp@redhat.com>

- * Add indent-tabs-mode: nil to all file headers.

2007-07-12 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps-util.c (_dbus_sysdeps_test): invert the test for parsing hex as double to be sure it fails to work
- * dbus/dbus-sysdeps.c (_dbus_string_parse_double): don't allow hex numbers.

2007-07-10 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-connection.c (struct DBusConnection): Fix from Olivier Hochreutiner to avoid trying to protect individual bits in a word with different locks (make dispatch_acquired and io_path_acquired dbus_bool_t rather than bitfields)

2007-07-09 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-win.c, dbus-sysdeps-win.h, dbus-sysdeps-win-util.c, dbus-sysdeps-spawn-win.c: synced with windbus sources

2007-07-07 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-win.c
 (_dbus_getsid): remove unused jump label
 (_dbus_read_credentials_socket): _dbus_string_init could fail,
 check initialisation
 (_dbus_get_working_dir, _dbus_init_working_dir): remove unused
 functions

2007-07-04 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/modules/FindKDEWIN.cmake: search in <program files
dir>\win32libs
for windows supplementary packages too

2007-06-30 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/dbus/CMakeLists.txt: handle userdb as unix file

2007-06-30 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c, dbus/dbus-sysdeps-win.h: removed
obsolete
DBusUserInfo code

2007-06-30 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c (_dbus_daemon_init): reduced compiler
warnings

2007-06-23 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-auth-script.c (_dbus_auth_script_run): added
UNIX_ONLY and
WIN_ONLY commands for auth scripts
* test/data/auth/external-root.auth-script: limit execution to
unix

2007-06-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-transport.c (auth_via_default_rules): made the verbose
spam
cross-platform

2007-06-21 Havoc Pennington <hp@redhat.com>

* dbus/dbus-watch.c (dbus_watch_get_fd): 1) its behavior should
not be the same as before, the ABI has never been declared stable
on Windows and 2) do not commit to cross-platform files without
posting the exact patch to the mailing list

* dbus/dbus-sysdeps-util.c (_dbus_sysdeps_test): as I have
pointed
out before, either 0xff needs to work on both platforms, or none
of the dbus code can rely on it working. That means the options
are 1) audit the code for anywhere that relies on 0xff working,
if
none found ideally add a test that it *doesn't* work and make
unix
reject it explicitly, but in any case this test would go away or
2) make it work on Windows also, then we don't have to figure out
whether we rely on it.

And in either case, post the exact patch to the mailing list and don't just commit.

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/CMakeLists.txt: added VERSION_PATCH to be able to distinguish win32 binary release from dbus versioning

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c: deleted local DBusCredentials structure
 (_dbus_getsid): new function
 (_dbus_read_credentials_socket): used correct function
 (_dbus_append_user_from_current_process,
 _dbus_credentials_add_from_current_process): added real
 sid reading
 (_dbus_credentials_parse_and_add_desired, _dbus_parse_uid):
 deleted

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-util.c (_dbus_sysdeps_test): don't check 0xff as floating point, this isn't supported on win32 math implementation

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c (_dbus_homedir_from_username,
 _dbus_homedir_from_current_process,
 _dbus_append_desired_identity):
 removed obsolete functions

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c, dbus/dbus-sysdeps-util-win.c,
 dbus/dbus-sysdeps-win.h: disabled uid/sid conversation stuff
 implementation by Peter Kueimmel

2007-06-21 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-watch.c (dbus_watch_get_fd): this function is deprecated and its behavior should be as before until all client code is migrated.

2007-06-19 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-util-win.c, tools/dbus-launch-win.c:
 msvc7.1 fixes by Jaroslaw Staniek tested with mingw

2007-06-19 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c, dbus/dbus-sysdeps-spawn-win.c,
dbus/dbus-sysdeps-win.h: disabled DBusSocket implementation
by Peter Kuemmel

2007-06-18 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: removed obsolete patches

2007-06-18 Havoc Pennington <hp@redhat.com>

* configure.in: bump version to 1.1.2 so CVS is higher than last
release (this is not the 1.1.2 release)

2007-06-18 Havoc Pennington <hp@redhat.com>

* Release 1.1.1

2007-06-18 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: document
org.freedesktop.DBus.GetId()

* bus/driver.c (bus_driver_handle_get_id): implement
org.freedesktop.DBus.GetId()

* bus/bus.c (bus_context_new): generate a unique ID for each bus
context

* dbus/dbus-connection.c (dbus_connection_get_server_id): new
function

* dbus/dbus-bus.c (dbus_bus_get_id): new function

* dbus/dbus-server.c (dbus_server_get_id): new function

2007-06-18 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-unix.c (_dbus_read_credentials_socket): clean
this up a little bit, to try and understand why telnet'ing to a
server and sending a non-nul byte didn't disconnect immediately;
now it seems that it does disconnect immediately as it should,
though I don't understand what has changed.

2007-06-18 Havoc Pennington <hp@redhat.com>

* dbus/dbus-watch.c (dbus_watch_get_socket)
(dbus_watch_get_unix_fd): new API to match DBusConnection
(dbus_watch_get_fd): deprecate this

Throughout: just s/dbus_watch_get_fd/dbus_watch_get_socket/g for
now since all the transports use sockets anyway

2007-06-16 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-macros.h, dbus/dbus-message.c, dbus/dbus-message.h: renamed DBUS_GNUC_DEPRECATED to DBUS_DEPRECATED and extended to msvc compiler

2007-06-15 Ralf Habacker <ralf.habacker@freenet.de>

- * cmake/CMakeLists.txt: use local include header first
- * dbus/dbus-sysdeps-win.c: mingw fix of DbusCredential struct

2007-06-15 Ralf Habacker <ralf.habacker@freenet.de>

- * cmake/ConfigureChecks.cmake, cmake/config.h.cmake: added check for HAVE_ERRNO_H
- * cmake/dbus/CMakeLists.txt: added missing files
- * dbus/dbus-transport-win.c/.h: new files
- * dbus/dbus-sysdeps-win.c,.h: added required _unix functions to make dbus compilable on win32
- * dbus/dbus-sysdeps-win-utils.c,.h: moved some functions to dbus-sysdeps-win.c
- * dbus-win.patch: removed applied or obsolete patches

Note: dbus-win32 is now compilable, no guarantee that it runs without any problems

2007-06-15 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps-unix.c (_dbus_append_session_config_file) (_dbus_append_system_config_file): new functions
- * bus/main.c (main): use _dbus_append_system_config_file() and _dbus_append_session_config_file()
- * dbus/Makefile.am (INCLUDES): move DBUS_SYSTEM_CONFIG_FILE and DBUS_SESSION_CONFIG_FILE into this makefile

2007-06-15 Havoc Pennington <hp@redhat.com>

- * dbus/dbus-sysdeps.c (_dbus_set_errno_to_zero) (_dbus_get_is_errno_nonzero, _dbus_get_is_errno_eintr) (_dbus_strerror_from_errno): family of functions to abstract errno, though these are somewhat bogus (really we should make our socket wrappers not use errno probably - the issue is that any usage of errno that isn't socket-related probably is not cross-platform, so should either be in a unix-only file that can

use errno directly, or is a bug - these general errno wrappers hide issues of this nature in non-socket code, while socket-specific API changes would not since sockets are allowed cross-platform)

2007-06-14 Havoc Pennington <hp@redhat.com>

* bus/dispatch.c (check_get_connection_unix_process_id): mop up getpid() (noticed by Peter KÄmmel) and adapt the test to expect a "pid unknown" error when running on Windows.

2007-06-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-unix.c
(_dbus_credentials_parse_and_add_user):
delete this function since it was effectively the same as
_dbus_credentials_add_from_username()

2007-06-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-auth.c: adapt to keyring changes

* dbus/dbus-keyring.c: change to avoid using user ID and home directory directly; instead use a keyring-location-from-credentials function in dbus-sysdeps

* fix to use _dbus_append_user_from_current_process() instead of _dbus_username_from_current_process() or _dbus_append_desired_identity().

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* reverted global rename of function
_dbus_username_from_current_process.
It needs too much tests to verify that the change does not break anything.
I had overseen that the signatures are different and requires non trivial changes.
This is one *major* disadvantage of emulating oop functionality with c.
You are responsible for cleaning every object on every function return point
which could be a nightmare if you are not working with dbus all the days.

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-auth.c
(handle_client_initial_response_cookie_shal_mech):
fixed usage of _dbus_append_desired_identity()

* dbus/dbus-sysdeps.h (_dbus_username_from_current_process):
removed prototype

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps.c: moved global lock system_users from dbus-userdb.c

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* global rename of function _dbus_username_from_current_process
to _dbus_append_desired_identity.
Approved by Havoc Pennington

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c: disabled DBusUserInfo related code
(_dbus_append_desired_identity,
_dbus_windows_user_is_process_owner):
new win32 functions as counterpart of unix related

(_dbus_send_credentials_socket, _dbus_read_credentials_socket):
renamed from ..._unix_socket
(_dbus_send_credentials_unix_socket): removed obsolete
function

* dbus/dbus-sysdeps-win-util.c: disabled DBusGroupInfo related
code

(_dbus_verify_daemon_user, _dbus_change_to_daemon_user):
new win32 functions as counterpart of unix related

2007-06-14 Simon McVittie <simon.mcvittie@collabora.co.uk>

* doc/dbus-specification.xml: say the protocol version is 1
instead of
0 (patch from Kristoffer LundÃ©n, fd.o#10033) and remove the
FIXME
about removing protocol version from messages (as per Havoc's
comment
on that bug)

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c (_dbus_pid_for_log, _dbus_flush_caches):
new win32 functions as counterpart of unix related

2007-06-14 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/modules/FindKDEWIN.cmake,
cmake/modules/FindKDEWIN_Packager.cmake,
cmake/modules/Win32Macros.cmake: new files from the
kdewin32 project

* cmake/CMakeLists.txt: cleaned support for kdewin installer and win32 explorer wrapper

2007-06-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_iter_open_container): Fix broken return_if_fail (kind of scary that test suite does not cover this)

2007-06-13 Havoc Pennington <hp@redhat.com>

* dbus/dbus-server-socket.c (_dbus_server_listen_socket): support all_interfaces=true|false for tcp servers

* dbus/dbus-sysdeps-unix.c (_dbus_listen_tcp_socket): support inaddr_any flag

* bus/selinux.c: fix some missing includes

* dbus/dbus-server-socket.c (_dbus_server_listen_socket): allow port to simply be omitted in addition to specifying 0

2007-06-13 Havoc Pennington <hp@redhat.com>

* configure.ac, bus/selinux.c, dbus/dbus-sysdeps-unix-util.c: add libaudit support, no clue what this means really but now we have it. Patches from Fedora package.

* bus/bus.c (bus_context_new): move selinux initialization after changing to daemon user, patch from Fedora package

* dbus/dbus-transport.c (auth_via_unix_user_function): fix a typo

2007-06-12 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c (dbus_message_iter_open_container): improve the checks/warnings for contained_signature a bit

2007-06-12 Havoc Pennington <hp@redhat.com>

* dbus/dbus-marshal-recursive.c (write_or_verify_typecode): improve the warning a bit if you write extra data into a message

2007-06-12 Havoc Pennington <hp@redhat.com>

* dbus/dbus-auth.c (sha1_handle_second_client_response) (handle_server_data_anonymous_mech): add the process ID from socket credentials, if available, even if not using EXTERNAL

* dbus/dbus-transport.c (auth_via_default_rules): support allow_anonymous flag

* dbus/dbus-connection.c (dbus_connection_get_is_anonymous) (dbus_connection_set_allow_anonymous): new API for controlling anonymous access

2007-06-09 Havoc Pennington <hp@redhat.com>

* dbus/dbus-string.c (_dbus_string_pop_line): fix this not to think an empty line is the end of the file. Also, fix some whitespace.

* dbus/dbus-string-util.c: add more tests for _dbus_string_pop_line() revealing that it thinks an empty line is the end of the file, which broke dbus-auth-script.c so it didn't really run the scripts

* dbus/dbus-auth.c: add ANONYMOUS mechanism

* dbus/dbus-auth-script.c (_dbus_auth_script_run): fix to detect an empty/no-op auth script; add commands to check that we have or don't have the expected credentials

2007-06-09 Havoc Pennington <hp@redhat.com>

* bus/policy.c (bus_policy_create_client_policy): gracefully continue if the connection has no unix user - just don't apply any unix user dependent rules.

* bus/config-parser.c: remove dbus-userdb.h usage

* bus/bus.c: remove dbus-userdb.h usage

* dbus/dbus-transport.c (_dbus_transport_get_is_authenticated): support Windows user function; also, fix the logic for checking auth as root in the default auth code (broken in the previous commit)

* dbus/dbus-connection.c (dbus_connection_set_windows_user_function): new function (dbus_connection_get_windows_user): new function

2007-06-09 Havoc Pennington <hp@redhat.com>

* bus/dispatch.c (check_get_connection_unix_process_id): adapt since sysdeps-unix.h stuff isn't included anymore

* bus/bus.c (bus_context_new): use more abstract functions to change user, so they can be no-ops on Windows

* dbus/dbus-credentials.c, dbus/dbus-credentials.h, dbus/dbus-credentials-util.c: new files containing a fully opaque DBusCredentials data type to replace the old not opaque one.

* configure.in (DBUS_UNIX): define DBUS_UNIX to match DBUS_WIN on windows

* dbus/dbus-userdb.h: prohibit on Windows, next step is to clean up the uses of it in bus/*.c and factor out the parts of cookie auth that depend on it

2007-06-07 Havoc Pennington <hp@redhat.com>

* dbus/dbus-message.c: improve some docs related to reading values from a message iter

2007-06-02 Ralf Habacker <ralf.habacker@freenet.de>

* cmake: added cygwin compile support

2007-06-01 Ralf Habacker <ralf.habacker@freenet.de>

* tools/dbus-launch-win.c: new file, replaces script wrapper on win32.

* cmake/dbus-launch.bat.cmake: removed obsolete file

2007-05-31 Ralf Habacker <ralf.habacker@freenet.de>

* bus/main.c (main): uses _dbus_get_config_file_name() to detect session.conf location on win32.

* dbus-sysdeps-win.h
(_dbus_get_config_file_name, _dbus_file_exists):
new prototyp, undefined interface after including windows.h
because
it makes trouble when a paramater is named interface.

* dbus-sysdeps-win.c (_dbus_get_install_root,
_dbus_get_config_file_name, _dbus_file_exists): new functions.

2007-05-27 Ralf Habacker <ralf.habacker@freenet.de>

* bus/policy.c, dbus/dbus-internals.c: fixed inconsistant line endings
as reported by Peter Kämmel.

2007-05-25 John (J5) Palmieri <johnp@redhat.com>

* Released 1.1.0

2007-05-25 John (J5) Palmieri <johnp@redhat.com>

* Split NEWS and ChangeLog into a .pre-1-0 file as per HACKING
We forgot to do this during the 1.0 release and it makes sense

now as we get ready to release the first 1.1.0 development tarball

2007-05-25 John (J5) Palmieri <johnp@redhat.com>

- * create directory test/data/valid-config-files/session.d so that make check passes

2007-05-25 John (J5) Palmieri <johnp@redhat.com>

- * INSTALL: remove dependancies for bindings since they no longer are a part of core

2007-05-25 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-server-win.c, dbus/dbus-server-win.h: new file with empty function `_dbus_server_listen_platform_specific()`.

- * dbus/dbus-server.c (listen_funcs): uses `_dbus_server_listen_platform_specific` on any platform.

2007-05-25 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-win.c (fill_win_user_info_homedir): not getting user info from a domain controller isn't an error, the computer may run in a non domain context.

2007-05-25 Ralf Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-win.c (`_dbus_full_duplex_pipe`): fixed assertion.

2007-05-24 Simon McVittie <simon.mcvittie@collabora.co.uk>

- * bus/config-parser.c (test_service_dir_matches): fixed ordering for unix.

- * bus/config-parser.c (test_default_session_servicedirs): made allocation of `_progs` platform independent.

2007-05-23 Havoc Pennington <hp@redhat.com>

- * bus/Makefile.am (install-data-hook): create session.d

- * bus/session.conf.in: add session.d for the session bus, so security policy can be extended

2007-05-22 Ralf.Habacker <ralf.habacker@freenet.de>

- * cmake/CMakeLists.txt: fixed creating of `TEST_..._BINARY` to make

bus-test able to find the binaries.

2007-05-21 Simon McVittie <simon.mcvittie@collabora.co.uk>

* acinclude.m4, configure.in: In recent autotools, `{datadir}` is defined in terms of `{datarootdir}`, so `EXPANDED_DATADIR` needs to be expanded recursively. Rather than fixing `configure.in` to do this, I grabbed the `AS_AC_EXPAND` macro from `autostars.sf.net`, which seems to be commonly used.

2007-05-21 Simon McVittie <simon.mcvittie@collabora.co.uk>

* `update-dbus-docs.sh`: Assorted improvements:
- Default user if `$FDUSER` is not set is the ssh default (set in `~/.ssh/config` or based on the local username), not a hard-coded "johnp"
- Temporary checkout directory is created securely (preventing symlink attacks), if `mktemp(1)` is available
- Use `make -C` rather than `cd && make && cd ..`

2007-05-21 Simon McVittie <simon.mcvittie@collabora.co.uk>

* HACKING: Point to correct mailing list

2007-05-21 Simon McVittie <simon.mcvittie@collabora.co.uk>

* `doc/dbus-specification.xml`: explicitly specify that `STRING` cannot contain embedded NULs.

2007-05-20 Ralf.Habacker <ralf.habacker@freenet.de>

* `dbus/dbus-internal.c`: fix inline problem on win32.

2007-05-20 Ralf.Habacker <ralf.habacker@freenet.de>

* `dbus/dbus-sysdeps-win.c` (`fill_win_user_info_homedir`): not reaching domain controller isn't an error, converted error message to verbose message.

2007-05-19 Ralf.Habacker <ralf.habacker@freenet.de>

* `dbus/dbus-test.c` (`dbus_internal_do_not_use_run_tests`): disabled tests not running on wince.

2007-05-19 Ralf.Habacker <ralf.habacker@freenet.de>

* dbus/dbus-internals.c (_dbus_verbose_init): win32 requires 'varname=' to unset a environment variable.

2007-05-19 Ralf.Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps.c (_dbus_setenv): win32 requires 'varname=' to unset a environment variable.

2007-05-19 Ralf.Habacker <ralf.habacker@freenet.de>

* bus/policy.c (bus_policy_create_client_policy): win32 temporary fix until the userdb stuff is ported completly.

2007-05-19 Ralf.Habacker <ralf.habacker@freenet.de>

* dbus/dbus-server.c (listen_funcs):
_dbus_server_listen_platform_specific is empty on win32,
limited to unix only.

* dbus/dbus-server.c (valid_addresses): limit unix
only addresses to unix.

2007-05-18 Havoc Pennington <hp@redhat.com>

* doc/dbus-specification.xml: add a GetAll to the Properties
interface.

2007-05-17 Ralf.Habacker <ralf.habacker@freenet.de>

* bus\config-parser.c (test_default_session_servicedirs):
win32 fix.

2007-05-17 Ralf.Habacker <ralf.habacker@freenet.de>

* configure.in: define constant DBUS_UNIX.

2007-05-14 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c (_dbus_printf_string_upper_bound):
compile fix for MS Platform SDK 6
patch from Michael Luschas <mluschas@gmail.com>

2007-05-10 John (J5) Palmieri <johnp@redhat.com>

* dbus-1.pc.in: add daemondir to pc file
patch from Brian Cameron <brian.cameron at sun.com>

2007-05-04 Simon McVittie <simon.mcvittie@collabora.co.uk>

* doc/dbus-specification.xml: mention the reserved
o.fd.DBus.Local
interface and the corresponding /o/fd/DBus/Local object path;
describe them as reserved and forbid using them in a sent
message,
mentioning that in the reference implementation this will cause
the
connection to be dropped. Patch approved by Havoc.

2007-04-28 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/: don't install test applications and service files,
moved CMAKE_DEBUG_POSTFIX to top level CMakeLists.txt

2007-04-27 Havoc Pennington <hp@redhat.com>

* dbus/dbus-sysdeps-unix.c (_dbus_open_socket): fix #10781 from
Tobias Nygren, checking pointer to fd vs. 0 rather than checking
the fd itself

2007-04-26 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/: added debug postfixes to debug exe's for
easier debugging.
* dbus/dbus-sysdeps-win.c (_dbus_win_set_error_from_win_error):
print error code in case no string message is available.

2007-04-06 Simon McVittie <simon.mcvittie@collabora.co.uk>

* dbus/dbus-message-util.c, dbus/dbus-message.c,
dbus/dbus-message.h: Add API to convert a DBusMessage to/from a
byte array. Patch from Dafydd Harries
<dafydd.harries@collabora.co.uk>,
approved by Havoc Pennington.

2007-04-03 Timo Hoenig <thoenig@suse.de>

* dbus/dbus-address.c (dbus_parse_address): Do not accept zero-
length address.
* dbus/dbus-address.c (_dbus_address_test): Add test for zero-
length address.

2007-03-25 Ralf.Habacker <ralf.habacker@freenet.de>

* cmake/dbus/CMakeLists.txt: debug postfix also for mingw.

2007-03-16 Ralf.Habacker <ralf.habacker@freenet.de>

* cmake/modules/FindExpat.cmake: fix package detection on win32.
* README.win: update install instructions.

2007-03-16 Ralf.Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps.h (_dbus_split_paths_and_append): new prototyp
(_DBUS_PATH_SEPARATOR): new macro.

- * dbus/dbus-sysdeps.c (_dbus_split_paths_and_append): merged from dbus/dbus-sysdeps-unix.c and dbus/dbus-sysdeps-win.c.

2007-03-15 Ralf.Habacker <ralf.habacker@freenet.de>

- * bus/config-parser.c, bus/policy.c, bus/policy.h, bus/dbus-daemon.1.in, bus/session.conf.in: added eavesdrop support for replies - patch by olli.salli at collabora.co.uk approved by Havoc Pennington.

2007-03-15 Ralf.Habacker <ralf.habacker@freenet.de>

- * dbus\dbus-sysdeps-win-thread.c: renamed to dbus-sysdeps-thread-win.c, it is a platform depending file similar to dbus-sysdeps-pthread.c.

2007-03-15 Ralf.Habacker <ralf.habacker@freenet.de>

- * cmake\doc\CMakeLists.txt: added preliminary xml doc support, needs cmake Find script.

2007-03-14 Ralf.Habacker <ralf.habacker@freenet.de>

- * cmake: add doxygen support

2007-03-14 Ralf.Habacker <ralf.habacker@freenet.de>

- * cmake/config.h.cmake: WINCE fixes, defined DBUS_UNIX.

2007-03-13 Ralf.Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-util-win.c (dbus_become_daemon): win32 compile fix.

2007-03-13 Ralf.Habacker <ralf.habacker@freenet.de>

- * dbus-win.patch: removed obsolete patches.

2007-03-13 Ralf.Habacker <ralf.habacker@freenet.de>

- * dbus/dbus-sysdeps-win.c: added zero byte sending and receiving after connection start up

2007-03-11 Havoc Pennington <hp@redhat.com>

* tools/dbus-launch.c (do_close_stderr): fix C89 problem and formatting problem

* Mostly fix the DbusPipe mess.
- put line break after function return types
- put space before parens
- do not pass structs around by value
- don't use dbus_strerror after calling supposedly cross-platform api
- don't name pipe variables "fd"
- abstract special fd numbers like -1 and 1

2007-03-12 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.h, dbus/dbus-sysdeps-win.c, dbus/dbus-sysdeps-util-win.c: renamed functions _dbus_xxx_file() to _dbus_file_xxx() to reflect struct name DBusFile.

2007-03-12 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-util-unix.c (_dbus_become_daemon): fix _dbus_pid_fd check.

2007-03-10 Ralf Habacker <ralf.habacker@freenet.de>

* tools/dbus-print-message.c (print_message): added printing of the reply serial to method returns and errors, so you can actually figure out the message/reply pairs - patch by olli.salli at collabora.co.uk.

2007-03-10 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: removed committed patches.

2007-03-10 Ralf Habacker <ralf.habacker@freenet.de>

* bus/bus.c, bus/bus.h, bus/main.c, bus/test.c, dbus/dbus-sysdeps-unix.c, dbus/dbus-sysdeps-util-unix.c, dbus/dbus-sysdeps-util-win.c, dbus/dbus-sysdeps-win.c, dbus/dbus-sysdeps.h: renamed _dbus_xxx_pipe to _dbus_pipe_xxx, completed _dbus_pipe support.

2007-03-10 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps.h (_dbus_listen_tcp_socket): changed type or port to pointer, because the port is given back.

* dbus/dbus-server-socket.c (_dbus_server_new_for_tcp_socket): implemented returning tcp port. Skipping port parameter and non integer port values in config <listen> statement needs more effort.

* dbus/dbus-sysdeps-unix.c, dbus/dbus-sysdeps-win.c
(_dbus_listen_tcp_socket): return the real used tcp port.

* bus/dbus-daemon.1.in: added <listen> tcp examples

2007-03-09 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/config.h.cmake: win32 msvc bug fix

2007-03-09 Ralf Habacker <ralf.habacker@freenet.de>

* cmake/config.h.cmake: fixed DBUS_WINxx defines,
using _WINCE does not work.

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: removed _dbus_write_pipe() patch, it is now
committed.

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* bus/bus.c, dbus/dbus-sysdeps-unix.c, dbus/dbus-sysdeps.h:
rename pipe related write() function calls to _dbus_write_pipe().

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: added bus/config-loader-libexpat.c patch,
uses DBUS_WIN for alls win32 related #ifdefs,
some minor cleanups

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: updated patch for bus/dispatch.c.

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-win.patch: dbus-connection.c
(dbus_connection_get_unix_user,
dbus_connection_get_unix_process_id): Let return them valid user
id's,
otherwise bus-test fails. How to replace on win32 ?.

* dbus/dbus-sysdeps-win.c (fill_win_user_info_homedir):
fix memory leak.

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win.c (_dbus_win_set_error_from_win_error):
use dbus friendly error name.
(fill_win_user_info_homedir): print user name in error case too.

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>
* cmake/ConfigureChecks.cmake: fixed socketpair check

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>
* bus/dispatch.c: disabled segfault test on win32 for now

2007-03-08 Ralf Habacker <ralf.habacker@freenet.de>
* configure.in, cmake/ConfigureChecks.cmake: added check for setrlimit.
* test/test-segfault.c: only include setrlimit stuff only when available.

2007-03-07 Ralf Habacker <ralf.habacker@freenet.de>
* test/test-segfault.c: unix compile fix.
* dbus-win.patch: removed obsolete patches.

2007-03-07 Ralf Habacker <ralf.habacker@freenet.de>
* bus/activation.c: removed obsolete include header.
* test/test-segfault.c: win32 compile fix, rlimit isn't available on win32.
* dbus-win.patch: removed some more patches, they are applied or obsolete

2007-03-06 Ralf Habacker <ralf.habacker@freenet.de>
* bus-win.patch: fixes unix listen problems, dbus-test now runs.

2007-03-06 Ralf Habacker <ralf.habacker@freenet.de>
* cmake/dbus/CMakeLists.txt, cmake/bus/CMakeLists.txt, cmake/CMakeLists.txt: win32 compile fix

2007-03-04 Ralf Habacker <ralf.habacker@freenet.de>
* dbus-win.patch, README.win: added available win32 patches from windbus project (<http://sf.net/projects/windbus>)

2007-03-04 Ralf Habacker <ralf.habacker@freenet.de>
* bus/activation.c: (bus_activation_activate_service): fixed call to _dbus_spawn_async_with_babysitter().

2007-03-04 Ralf Habacker <ralf.habacker@freenet.de>
* dbus/dbus-spawn.c, dbus/dbus-spawn.h (_dbus_spawn_async_with_babysitter):

added environment pointer as function parameter, used on win32.

* test/spawn-test.c: fixed call to above mentioned function.

2007-03-04 Ralf Habacker <ralf.habacker@freenet.de>

* configure.in, test/test-sleep-forever.c, test/test-names.c:
added configure check for unistd.h.

2007-03-04 Ralf Habacker <ralf.habacker@freenet.de>

* test/Makefile.am: fixed test data copy problem in
out of source build, when sources came from svn or cvs.

2007-03-03 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/*-win.*, bus/*-win.*: added win32 platform related
files. These files are only added to the cmake build system.
The missing dbus-win.patch file will be added later.

2007-03-03 Ralf Habacker <ralf.habacker@freenet.de>

* cmake: new directory, contains cmake build support.
See <http://www.cmake.org> for more informations.
Currently only unix will be buildable because some
win32 required files are still missing.

2007-03-03 Thiago Macieira <thiago@kde.org>

* dbus/dbus-sysdeps-unix.c: capture the dbus-launch stderr
output and add it to the DBusError message we return.

* tools/dbus-launch.1:

* tools/dbus-launch.c: Add option --close-stderr to, well,
close stderr before starting dbus-daemon.

2007-01-31 Havoc Pennington <hp@redhat.com>

* bus/dbus-daemon.1.in: write a section in the man page on
running
a test daemon for debugging purposes

2007-01-26 Havoc Pennington <hp@redhat.com>

* bus/session.conf.in: override all the default limits with much
higher limits on the session bus, there is no reason the session
bus should have low limits

* bus/config-parser.c (bus_config_parser_new): increase default
limits so they are less likely to be hit; in particular the max
replies per connection was way too low

2006-01-25 Simon McVittie <simon.mcvittie@collabora.co.uk>

* doc/dbus-tutorial.xml: Replace Python section of tutorial with a pointer to the tutorial maintained as part of dbus-python

2006-12-31 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-unix.c: unix compile fix, moved atomic_exchange_and_add() from dbus/dbus-sysdeps.c to here, it's used by _dbus_atomic_inc() and _dbus_atomic_dec().

2006-12-31 Ralf Habacker <ralf.habacker@freenet.de>

* tools/dbus-monitor.c: gettimeofday() is not available on windows so we have to provide our own. It's taken from lgpl'd kdedwin32 package. - Patches from Christian Ehrlicher

2006-12-31 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-unix.c: moved _dbus_atomic_inc/dec() from dbus/dbus-sysdeps.c, windows version of _dbus_atomic_inc/dec() is in dbus-sysdeps-win.c (not in this patch).

* dbus/dbus-sysdeps.h: DBusAtomic::value is long on windows to fit with InterlockedInc/Decrement. - Patches from Christian Ehrlicher

2006-12-31 Ralf Habacker <ralf.habacker@freenet.de>

* tools/dbus-send.c, tools/dbus-monitor.c: win32 compile fix.

2006-12-31 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-marshal-recursive.c (type DBusTypeReaderClass): fix mispaced const statement. - Patch from KÄmmel

2006-12-19 Ray Strobe <rstroke@redhat.com>

* bus/bus.c (process_config_every_time): don't overwrite existing bus context activation object until after we've checked that the new activation is valid.

* bus/main.c (signal_handler), (handle_reload_watch): don't call exit() on failure, instead make do and keep going. (close_reload_pipe): new function to turn off hangup-causes-config-reload behavior if an unexpected error occurs

2006-12-13 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-sysdeps-win-thread.c (_dbus_condvar_wait_win32):
correctness fix. - Patch from Christian Ehrlicher

2006-12-13 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-internals.h: msvc also knows about __FUNCTION__,
we should also use it. - Patch from Christian Ehrlicher

2006-12-13 Ralf Habacker <ralf.habacker@freenet.de>

* dbus-sysdeps-util.c: added win32 related tests

2006-12-12 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-string.c (_dbus_string_pop_line),
bus/desktop-file.c (parse_section_start,
parse_comment_or_blank,parse_key_value,): uses
_dbus_string_find_eol() to support platform independent eol
style.

2006-12-12 Ralf Habacker <ralf.habacker@freenet.de>

* dbus/dbus-string.[ch] (_dbus_string_find_eol): new function.
* dbus/dbus-string-util.c (_dbus_string_test): added testcases
for
_dbus_string_find_eol().
Approved by: Havoc Pennington.

2006-12-12 Tim Dijkstra <tim@famdijkstra.org>

* configure.in: Added switch to disable user_database caching.
* dbus/dbus-userdb-util.c, dbus/dbus-userdb.c: Add ifdefs to
be able disable user_database caching.

2006-12-12 Tim Dijkstra <tim@famdijkstra.org>

* bus/bus.c, bus/bus.h: Remove DBusUserDatabase from the
BusContext
struct. It is unnecessary we have a global one already. Also
remove
bus_context_get_user_database function, it is no longer needed.
Flush the global database on reload.

* dbus/dbus-userdb-util.c: Replace _dbus_user_database_get_groups
with _dbus_groups_from_uid. It no longer needs a
DBusUserDatabase.

* dbus/dbus-userdb.c, dbus/dbus-userdb.h:

Add `_dbus_user_database_flush_system`.
Make more functions `DBUS_USERDB_INCLUDES_PRIVATE`.
Small unrelated change in `_dbus_is_a_number`: change
`_dbus_string_parse_int` to `_dbus_string_parse_uint`.

* `bus/connection.c`: Change call to `_dbus_user_database_get_groups`
to
`_dbus_groups_from_uid`.

* `bus/policy.c`, `bus/policy.h`: Change call to
`_dbus_user_database_get_groups` to `_dbus_groups_from_uid`. Remove
`DBusUserDatabase` from `bus_policy_allow_user` prototype, it no
longer
needs it.

2006-12-12 John (J5) Palmieri <johnp@redhat.com>

* `bus/signal.c`: Fix `match_rule_equal` errata
(CVE-2006-6107 - Patch from Kimmo Hamalainen
<kimmo.hamalainen@nokia.com>)

2006-11-19 Thiago Macieira <thiago@kde.org>

* `dbus/dbus-sysdeps-pthread.c` (`_dbus_pthread_mutex_lock`,
`_dbus_pthread_condvar_wait`,
`_dbus_pthread_condvar_wait_timeout`): set `pmutex->holder` to
`pthread_self()` after coming back from a conditional variable
wait as well as in one codepath where it was forgotten.
Approved by: Havoc Pennington.

2006-11-17 Havoc Pennington <hp@redhat.com>

* `update-dbus-docs.sh`: allow setting `fd.org` username via env
variable. Make it run autogen with `--enable-xml-docs=yes`
`--enable-doxygen-docs=yes` so configure will fail if the required
tools are missing.

2006-11-17 Havoc Pennington <hp@redhat.com>

* `doc/dbus-faq.xml`: minor FAQ tweaks

2006-11-14 Havoc Pennington <hp@redhat.com>

* `dbus/dbus-misc.c`, `dbus/dbus-misc.h`: Move
`dbus_get_local_machine_id()` to its own file, no substantive
changes. There are a couple other things we might want to add
that
are "misc" so moving out of `dbus-connection.[hc]` which is big
enough already.

2006-11-14 Havoc Pennington <hp@redhat.com>

* dbus/dbus-internals.c (`_dbus_generate_uuid`): The spec said the UUID had the timestamp last, but the implementation had it first; move it to last since I think it's a tiny bit nicer (easier to compare at a glance, faster to sort, less code), and will not cause any practical compatibility problems. Also, always convert the timestamp to big endian.

* doc/dbus-specification.xml: Clean up the docs on the UUID.

* tools/dbus-uuidgen.1: more prominently say it is not suitable as a replacement for regular `uuidgen/RFC4122`.

2006-11-14 John (J5) Palmieri <johnp@redhat.com>

* dbus/dbus-threads.h: fix `DBUS_THREAD_FUNCTIONS_ALL_MASK` to have the correct value so we don't assert when initializing recursive threads

* test/name-test/test-thread-init.c: call `dbus_threads_init_default` instead of `_dbus_threads_init_debug` since it is more of a real world test

File = ChangeLog.~1~

2008-04-16 John (J5) Palmieri <johnp@redhat.com>

* tools/dbus-monitor.c: take out the trailing comma in the `ProfileAttributeFlags` enum as it isn't in our style guides and causes some compilers to error out

File = CheckForAbstractSockets.c

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <errno.h>
```

```
int main() {
    int listen_fd;
    struct sockaddr_un addr;

    listen_fd = socket (PF_UNIX, SOCK_STREAM, 0);
```

```

if (listen_fd < 0)
{
    fprintf (stderr, "socket() failed: %s\n", strerror (errno));
    exit (1);
}

memset (&addr, '\0', sizeof (addr));
addr.sun_family = AF_UNIX;
strcpy (addr.sun_path, "X/tmp/dbus-fake-socket-path-used-in-
configure-test");
addr.sun_path[0] = '\0'; /* this is what makes it abstract */

if (bind (listen_fd, (struct sockaddr*) &addr, SUN_LEN (&addr)) < 0)
{
    fprintf (stderr, "Abstract socket namespace bind() failed:
%s\n",
            strerror (errno));
    exit (1);
}
else
    exit (0);
}

```

File = CheckPrototypeExists.cmake

```

# - Check if the prototype for a function exists.
# CHECK_PROTOTYPE_EXISTS (FUNCTION HEADER VARIABLE)
#
# FUNCTION - the name of the function you are looking for
# HEADER - the header(s) where the prototype should be declared
# VARIABLE - variable to store the result
#
# The following variables may be set before calling this macro to
# modify the way the check is run:
#
# CMAKE_REQUIRED_FLAGS = string of compile command line flags
# CMAKE_REQUIRED_DEFINITIONS = list of macros to define (-DFOO=bar)
# CMAKE_REQUIRED_INCLUDES = list of include directories

INCLUDE(CheckCXXSourceCompiles)

MACRO (CHECK_PROTOTYPE_EXISTS _SYMBOL _HEADER _RESULT)
    SET(_INCLUDE_FILES)
    FOREACH (it ${_HEADER})
        SET(_INCLUDE_FILES "${_INCLUDE_FILES}#include <${it}>\n")
    ENDFOREACH (it)

    SET(_CHECK_PROTO_EXISTS_SOURCE_CODE "
${_INCLUDE_FILES}
int main()
{

```

```

#ifndef ${_SYMBOL}
    int i = sizeof(&${_SYMBOL});
#endif
    return 0;
}
")
    CHECK_CXX_SOURCE_COMPILES("${_CHECK_PROTO_EXISTS_SOURCE_CODE}"
${_RESULT})
ENDMACRO (CHECK_PROTOTYPE_EXISTS _SYMBOL _HEADER _RESULT)

```

File = CheckStructMember.cmake

```

# - Check if the given struct or class has the specified member
variable
# CHECK_STRUCT_MEMBER (STRUCT MEMBER HEADER VARIABLE)
#
# STRUCT - the name of the struct or class you are interested in
# MEMBER - the member which existence you want to check
# HEADER - the header(s) where the prototype should be declared
# VARIABLE - variable to store the result
#
# The following variables may be set before calling this macro to
# modify the way the check is run:
#
# CMAKE_REQUIRED_FLAGS = string of compile command line flags
# CMAKE_REQUIRED_DEFINITIONS = list of macros to define (-DFOO=bar)
# CMAKE_REQUIRED_INCLUDES = list of include directories

```

INCLUDE(CheckCXXSourceCompiles)

```

MACRO (CHECK_STRUCT_MEMBER _STRUCT _MEMBER _HEADER _RESULT)
    SET(_INCLUDE_FILES)
    FOREACH (it ${_HEADER})
        SET(_INCLUDE_FILES "${_INCLUDE_FILES}#include <${it}>\n")
    ENDFOREACH (it)

    SET(_CHECK_STRUCT_MEMBER_SOURCE_CODE "
${_INCLUDE_FILES}
int main()
{
    ${_STRUCT}* tmp;
    tmp->${_MEMBER};
    return 0;
}
")
    CHECK_CXX_SOURCE_COMPILES("${_CHECK_STRUCT_MEMBER_SOURCE_CODE}"
${_RESULT})
ENDMACRO (CHECK_STRUCT_MEMBER)

```

File = circular-1.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
<include>circular-1.conf</include>
</busconfig>
```

File = circular-2.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
<include>circular-3.conf</include>
</busconfig>
```

File = circular-3.conf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
<include>circular-2.conf</include>
</busconfig>
```

File = cleanup-man-pages.sh

```
#!/bin/sh

## This script cleans up private/internal API from the man pages
## generated by Doxygen. This brings the man pages down from 7 megs
## to 2 megs and avoids namespace-polluting man pages. It's probably
## pretty specific to GNU utilities. Patches gladly accepted to make
## it work without them.

## You would run this after building dbus and after running "doxygen
## Doxyfile"

die() {
    echo "$*" 1>&2
    exit 1
}
```

```

MANDIR=$1
if test x"$MANDIR" = x ; then
    MANDIR=doc/api/man/man3dbus
fi

cd "$MANDIR" || die "Could not cd to $MANDIR"

test -d keep || mkdir keep || die "Could not create $MANDIR/keep
directory"
test -d nuke || mkdir nuke || die "Could not create $MANDIR/nuke
directory"

## blacklist
(find . -maxdepth 1 -name "_" | xargs -I ITEMS /bin/mv ITEMS nuke) ||
die "could not move all underscore-prefixed items"
(find . -maxdepth 1 -name "DBus*Internal*" | xargs -I ITEMS /bin/mv
ITEMS nuke) || die "could not move all internal-containing items"
(find . -maxdepth 1 -name "dbus*_internal_" | xargs -I ITEMS /bin/mv
ITEMS nuke) || die "could not move all internal-containing items"

## this is kind of unmaintainable, but I guess it's no huge disaster
if we miss something.
## this list should only contain man pages with >1 line, i.e. with
real content; the
## one-line cross-references get cleaned up next.
for I in DBusCounter.* DBusCredentials.* DBusDataSlot.*
DBusDataSlotAllocator.* DBusDataSlotList.* \
    DBusDirIter.* DBusFakeMutex.* DBusFreedElement.* DBusGroupInfo.*
DBusGUID.* DBusHashEntry.* \
    DBusHashIter.* DBusHashTable.* DBusHeader.* DBusHeaderField.*
DBusKey.* DBusKeyring.* DBusList.* \
    DBusMarshal.* DBusMD5.* DBusMemBlock.* DBusMemPool.*
DBusMessageGenerator.* DBusMessageLoader.* \
    DBusMessageRealIter.* DBusObjectSubtree.* DBusObjectTree.*
DBusPollFD.* DBusReal.* \
    DBusResources.* DBusServerDebugPipe.* DBusServerSocket.*
DBusServerUnix.* \
    DBusServerVTable.* DBusSHA.* DBusSHAContext.*
DBusSignatureRealIter.* DBusStat.* DBusString.* \
    DBusSysdeps.* DBusSysdepsUnix.* DBusTimeoutList.* DBusTransport*
DBusTypeReader.* DBusTypeWriter.* \
    DBusUserInfo.* DBusWatchList.* ; do
    if test -f "$I" ; then
        /bin/mv "$I" nuke || die "could not move $I to $MANDIR/nuke"
    fi
done

## many files just contain ".so man3dbus/DBusStringInternals.3dbus" or
the like,
## if these point to something we nuked, we want to also nuke the
pointer.
for I in * ; do

```

```

    if test -f "$I" ; then
        LINES=`wc -l "$I" | cut -d ' ' -f 1`
        if test x"$LINES" = x1 ; then
            REF_TO=`cat "$I" | sed -e 's/\.so man3dbus\///g'`
            ## echo "$I points to $REF_TO"
            if ! test -f "$REF_TO" ; then
                /bin/mv "$I" nuke || die "could not move $I to
$MANDIR/nuke"
            fi
        fi
    fi
done

## whitelist
(find . -maxdepth 1 -name "dbus_*" | xargs -I ITEMS /bin/mv ITEMS
keep) || die "could not move all dbus-prefixed items"
(find . -maxdepth 1 -name "DBUS_*" | xargs -I ITEMS /bin/mv ITEMS
keep) || die "could not move all DBUS_-prefixed items"
(find . -maxdepth 1 -name "DBus*" | xargs -I ITEMS /bin/mv ITEMS keep)
|| die "could not move all DBus-prefixed items"

## everything else is assumed irrelevant, this is mostly struct fields
## from the public headers
(find . -maxdepth 1 -type f | xargs -I ITEMS /bin/mv ITEMS nuke) ||
die "could not move remaining items"

NUKE_COUNT=`find nuke -type f -name "*" | wc -l`
KEEP_COUNT=`find keep -type f -name "*" | wc -l`
MISSING_COUNT=`find . -maxdepth 1 -type f -name "*" | wc -l`

echo "$KEEP_COUNT man pages kept and $NUKE_COUNT man pages to remove"
echo "$MISSING_COUNT not handled"

(find keep -type f -name "*" | xargs -I ITEMS /bin/mv ITEMS .) || die
"could not move kept items back"

rmdir keep || die "could not remove $MANDIR/keep"

echo "Man pages to be installed are in $MANDIR and man pages to ignore
are in $MANDIR/nuke"

exit 0

File = client-out-of-mechanisms.auth-script

## this tests that tests that the client disconnects when it's out of
## known mechanisms

CLIENT
EXPECT_COMMAND AUTH

```



```
SEND 'REJECTED BONGO_MD5'  
EXPECT_STATE NEED_DISCONNECT
```

```
File = CMakeLists.txt
```

```
project(tools)
```

```
set (dbus_send_SOURCES  
    ../../tools/dbus-print-message.c  
    ../../tools/dbus-print-message.h  
    ../../tools/dbus-send.c  
)
```

```
set (dbus_monitor_SOURCES  
    ../../tools/dbus-monitor.c  
    ../../tools/dbus-print-message.c  
    ../../tools/dbus-print-message.h  
)
```

```
if (WIN32)  
set (dbus_launch_SOURCES  
    ../../tools/dbus-launch-win.c  
)  
else (WIN32)  
set (dbus_launch_SOURCES  
    ../../tools/dbus-launch.c  
)  
endif (WIN32)
```

```
if (DBUS_BUILD_X11)  
set (dbus_launch_SOURCES  
    ${dbus_launch_SOURCES}  
    ../../tools/dbus-launch-x11.c  
)  
endif (DBUS_BUILD_X11)
```

```
set (dbus_cleanup_sockets_SOURCES  
    ../../tools/dbus-cleanup-sockets.c  
)
```

```
add_executable(dbus-send ${dbus_send_SOURCES})  
target_link_libraries(dbus-send ${DBUS_LIBRARIES})  
install_targets(/bin dbus-send )
```

```
add_executable(dbus-launch ${dbus_launch_SOURCES})  
target_link_libraries(dbus-launch )  
if (DBUS_BUILD_X11)  
    target_link_libraries(dbus-launch ${X11_LIBRARIES} )  
endif (DBUS_BUILD_X11)  
install_targets(/bin dbus-launch )
```

```

add_executable(dbus-monitor ${dbus_monitor_SOURCES})
target_link_libraries(dbus-monitor ${DBUS_LIBRARIES})
install_targets(/bin dbus-monitor )

File = CMakeLists.txt.~1~

project(bus)

add_definitions(-DDBUS_COMPILATION)

SET(EFENCE "")
SET(BUS_DIR ${CMAKE_SOURCE_DIR}/../bus)

set (config_DATA
    session.conf
    system.conf
)

# config files for installation
CONFIGURE_FILE( "${BUS_DIR}/session.conf.in"
"${CMAKE_CURRENT_BINARY_DIR}/session.conf" IMMEDIATE @ONLY)
CONFIGURE_FILE( "system.conf.cmake"
"${CMAKE_CURRENT_BINARY_DIR}/system.conf" IMMEDIATE @ONLY)

# copy services for local daemon start to local service dir data/dbus-
1/services
SET (SERVICE_FILES test/data/valid-service-files)
FILE(GLOB FILES "${CMAKE_SOURCE_DIR}/../${SERVICE_FILES}/*.service.in"
)
FOREACH(FILE ${FILES})
    GET_FILENAME_COMPONENT(FILENAME ${FILE} NAME_WE)
    SET (TARGET ${CMAKE_BINARY_DIR}/data/dbus-
1/services/${FILENAME}.service)
    IF (CONFIG_VERBOSE)
        MESSAGE("FROM: ${FILE}\nTO: ${TARGET}\n")
    ENDIF (CONFIG_VERBOSE)
    configure_file(${FILE} ${TARGET} )
ENDFOREACH(FILE)

if(DBUS_USE_EXPAT)
    SET (XML_SOURCES ${BUS_DIR}/config-loader-expat.c)
else(DBUS_USE_EXPAT)
    SET (XML_SOURCES ${BUS_DIR}/config-loader-libxml.c)
endif (DBUS_USE_EXPAT)

# after next cvs update
#set (DIR_WATCH_SOURCE ${BUS_DIR}/dir-watch-default.c)
set (DIR_WATCH_SOURCE )

```

```

set (BUS_SOURCES
    ${BUS_DIR}/activation.c
    ${BUS_DIR}/activation.h
    ${BUS_DIR}/bus.c
    ${BUS_DIR}/bus.h
    ${BUS_DIR}/config-parser.c
    ${BUS_DIR}/config-parser.h
    ${BUS_DIR}/config-parser-common.c
    ${BUS_DIR}/config-parser-common.h
#    ${BUS_DIR}/config-parser-trivial.c
    ${BUS_DIR}/connection.c
    ${BUS_DIR}/connection.h
    ${BUS_DIR}/desktop-file.c
    ${BUS_DIR}/desktop-file.h
    ${BUS_DIR}/dir-watch-default.c
    ${BUS_DIR}/dir-watch.h
    ${BUS_DIR}/dispatch.c
    ${BUS_DIR}/dispatch.h
    ${BUS_DIR}/driver.c
    ${BUS_DIR}/driver.h
    ${BUS_DIR}/expirelist.c
    ${BUS_DIR}/expirelist.h
    ${BUS_DIR}/policy.c
    ${BUS_DIR}/policy.h
    ${BUS_DIR}/selinux.h
    ${BUS_DIR}/selinux.c
    ${BUS_DIR}/services.c
    ${BUS_DIR}/services.h
    ${BUS_DIR}/signals.c
    ${BUS_DIR}/signals.h
    ${BUS_DIR}/test.c
    ${BUS_DIR}/test.h
    ${BUS_DIR}/utils.c
    ${BUS_DIR}/utils.h
    ${XML_SOURCES}
    ${DIR_WATCH_SOURCE}
)
if(DBUS_ENABLE_STATS)
    list(APPEND BUS_SOURCES
        ${BUS_DIR}/stats.c
        ${BUS_DIR}/stats.h
    )
endif()

include_directories(${XML_INCLUDE_DIR})

add_executable(dbus-daemon ${BUS_SOURCES} ${BUS_DIR}/main.c)
target_link_libraries(dbus-daemon ${DBUS_INTERNAL_LIBRARIES}
    ${XML_LIBRARY})
set_target_properties(dbus-daemon PROPERTIES OUTPUT_NAME
    ${DBUS_DAEMON_NAME})

```

```

set_target_properties(dbus-daemon PROPERTIES COMPILE_FLAGS
${DBUS_INTERNAL_CLIENT_DEFINITIONS})

install_targets(/bin dbus-daemon)
install_files(/etc/dbus-1 FILES ${config_DATA})
install(DIRECTORY . DESTINATION etc/dbus-1/session.d FILES_MATCHING
PATTERN "*.conf")

if (DBUS_SERVICE)
    set (dbus_service_SOURCES
        ${BUS_DIR}/bus-service-win.c
        # TODO: add additional files
        #     ${BUS_DIR}/service-main.c
        #     ${BUS_SOURCES}
    )

    add_executable(dbus-service ${dbus_service_SOURCES} )
    target_link_libraries(dbus-service ${DBUS_INTERNAL_LIBRARIES}
${XML_LIBRARY})
    set_target_properties(dbus-service PROPERTIES COMPILE_FLAGS
${DBUS_INTERNAL_CLIENT_DEFINITIONS})
    install_targets(/bin dbus-service )
endif (DBUS_SERVICE)

if (DBUS_BUILD_TESTS)
    add_executable(bus-test ${BUS_SOURCES} ${BUS_DIR}/test-main.c)
    target_link_libraries(bus-test ${DBUS_INTERNAL_LIBRARIES}
${XML_LIBRARY})
    set_target_properties(bus-test PROPERTIES COMPILE_FLAGS
${DBUS_INTERNAL_CLIENT_DEFINITIONS})
    add_test(bus-test ${EXECUTABLE_OUTPUT_PATH}/bus-test
${CMAKE_BINARY_DIR}/test/data)
endif (DBUS_BUILD_TESTS)

if(MSVC)
    project_source_group(${GROUP_CODE} bus_test_SOURCES dummy)
endif(MSVC)

## mop up the gcov files
#clean-local:
#    /bin/rm *.bb *.bbg *.da *.gcov || true

#install-data-hook:
#    $(mkinstalldirs) $(DESTDIR)/$(localstatedir)/run/dbus
#    $(mkinstalldirs) $(DESTDIR)/$(configdir)/system.d
#    $(mkinstalldirs) $(DESTDIR)/$(datadir)/dbus-1/services

##install_file(${configdir}/system.d FILE

set(LAUNCH_HELPER_SOURCES ${XML_SOURCES}
    ${BUS_DIR}/config-parser-common.c

```

```

    ${BUS_DIR}/config-parser-trivial.c
    ${BUS_DIR}/desktop-file.c
    ${BUS_DIR}/utils.c
    ${BUS_DIR}/activation-helper.c
)

if(NOT WIN32)
# TODO PENDING(kdab) fix build on windows (activation-helper.c)
  add_executable(dbus-daemon-launch-helper ${LAUNCH_HELPER_SOURCES}
    ${BUS_DIR}/activation-helper-bin.c )
  target_link_libraries(dbus-daemon-launch-helper
    ${DBUS_INTERNAL_LIBRARIES} ${XML_LIBRARY} )

  add_executable(dbus-daemon-launch-helper-test
    ${LAUNCH_HELPER_SOURCES} ${BUS_DIR}/activation-helper-bin.c)
  set_target_properties(dbus-daemon-launch-helper-test PROPERTIES
    COMPILE_FLAGS "-DACTIVATION_LAUNCHER_TEST")
  target_link_libraries(dbus-daemon-launch-helper-test
    ${DBUS_INTERNAL_LIBRARIES} ${XML_LIBRARY} )

  add_executable(bus-test-launch-helper ${LAUNCH_HELPER_SOURCES}
    ${BUS_DIR}/test-launch-helper.c)
  set_target_properties(bus-test-launch-helper PROPERTIES
    COMPILE_FLAGS "-DACTIVATION_LAUNCHER_TEST -
    DACTIVATION_LAUNCHER_DO_OOM")
  target_link_libraries(bus-test-launch-helper
    ${DBUS_INTERNAL_LIBRARIES} ${XML_LIBRARY} )
  add_test(bus-test-launch-helper ${EXECUTABLE_OUTPUT_PATH}/bus-test-
    launch-helper )

endif(NOT WIN32)

#### Init scripts fun
#SCRIPT_IN_FILES=messagebus.in
#      rc.messagebus.in

## Red Hat start
#if DBUS_INIT_SCRIPTS_RED_HAT

#initddir=$(sysconfdir)/rc.d/init.d

#initd_SCRIPTS=
#      messagebus

#endif
# ## Red Hat end

## Slackware start
#if DBUS_INIT_SCRIPTS_SLACKWARE

#initddir=$(sysconfdir)/rc.d/

```

```

#initd_SCRIPTS=
#   rc.messagebus

#endif
## Slackware end

#MAN_IN_FILES=dbus-daemon.1.in
#man_MANS = dbus-daemon.1

#### Extra dist

#EXTRA_DIST=$(CONFIG_IN_FILES) $(SCRIPT_IN_FILES) $(man_MANS)
$(MAN_IN_FILES)

File = CMakeLists.txt.~2~

project(dbus)

# we need to be up to date
CMAKE_MINIMUM_REQUIRED(VERSION 2.4.4 FATAL_ERROR)
if(COMMAND cmake_policy)
    cmake_policy(SET CMP0003 NEW)
endif(COMMAND cmake_policy)

# where to look first for cmake modules, before ${CMAKE_ROOT}/Modules/
is checked
set(CMAKE_MODULE_PATH "${CMAKE_SOURCE_DIR}/modules")

# detect version
include(MacrosAutotools)
autoversion(../configure.ac dbus)
# used by file version info
set (DBUS_PATCH_VERSION "0")

include(Macros)
TIMESTAMP(DBUS_BUILD_TIMESTAMP)

##### basic vars #####

if (DBUSDIR)
    set(DBUS_INSTALL_DIR "${DBUSDIR}")
endif (DBUSDIR)
if ($ENV{DBUSDIR})
    set(DBUS_INSTALL_DIR "$ENV{DBUSDIR}")
endif ($ENV{DBUSDIR})

if (DBUS_INSTALL_DIR)

```

```

        set(CMAKE_INSTALL_PREFIX "${DBUS_INSTALL_DIR}" CACHE PATH
"install prefix" FORCE)
else (DBUS_INSTALL_DIR)
    set(DBUS_INSTALL_DIR "${CMAKE_INSTALL_PREFIX}")
endif (DBUS_INSTALL_DIR)

# autotools style
if (NOT DATAROOTDIR)
    set (DATAROOTDIR share)
endif()

if (NOT DATADIR)
    set (DATADIR ${DATAROOTDIR})
endif()

if (NOT DOCDIR)
    SET(DOCDIR ${DATAROOTDIR}/doc/dbus)
endif()

if (NOT DBUS_DATADIR)
    SET(DBUS_DATADIR ${DATADIR})
endif()

set(prefix                ${DBUS_INSTALL_DIR})
set(exec_prefix            ${prefix})
set(EXPANDED_LIBDIR        ${DBUS_INSTALL_DIR}/lib)
set(EXPANDED_INCLUDEDIR    ${DBUS_INSTALL_DIR}/include)
set(EXPANDED_BINDIR        ${DBUS_INSTALL_DIR}/bin)
set(EXPANDED_SYSCONFDIR    ${DBUS_INSTALL_DIR}/etc)
set(EXPANDED_DATADIR       ${DBUS_INSTALL_DIR}/${DBUS_DATADIR})
set(DBUS_MACHINE_UUID_FILE ${DBUS_INSTALL_DIR}/lib/dbus/machine-id)
set(DBUS_BINDIR            ${EXPANDED_BINDIR})
set(DBUS_DAEMONDIR         ${EXPANDED_BINDIR})

#enable building of shared library
SET(BUILD_SHARED_LIBS ON)

if (CYGWIN)
    set (WIN32)
endif (CYGWIN)

# search for required packages
if (WIN32)
    # include local header first to avoid using old installed header
    set (CMAKE_INCLUDE_PATH ${CMAKE_INCLUDE_PATH}
${CMAKE_SOURCE_DIR}/..)
    find_package(LibIconv)
    include(Win32Macros)
    addExplorerWrapper(${CMAKE_PROJECT_NAME})
endif (WIN32)

```

```

option (DBUS_USE_EXPAT "Use expat (== ON) or libxml2 (==OFF)" ON)

if(NOT WIN32)
    option (DBUS_ENABLE_ABSTRACT_SOCKETS "enable support for abstract
sockets" ON)
    set (CMAKE_THREAD_PREFER_PTHREAD ON)
    include (FindThreads)
endif(NOT WIN32)

#AC_ARG_ENABLE(asserts, AS_HELP_STRING([--enable-asserts],[include
assertion
checks]),enable_asserts=$enableval,enable_asserts=$USE_MAINTAINER_MODE
)
option (DBUS_DISABLE_ASSERTS "Disable assertion checking" OFF)

option (DBUS_ENABLE_STATS "enable bus daemon usage statistics" OFF)

option (DBUS_ENABLE_STATS "enable bus daemon usage statistics" OFF)

if (DBUS_USE_EXPAT)
    find_package(LibExpat)
else ()
    find_package(LibXml2)
endif ()
find_package(X11)

# analogous to AC_USE_SYSTEM_EXTENSIONS in configure.ac
add_definitions(-D_POSIX_C_SOURCE=199309L -D_GNU_SOURCE)

# do config checks
INCLUDE(ConfigureChecks.cmake)

# @TODO: how to remove last dir from ${CMAKE_SOURCE_DIR} ?
SET(DBUS_SOURCE_DIR ${CMAKE_SOURCE_DIR}/..)

# make some more macros available
include (MacroLibrary)

if(VCS)
    set(DBUS_VERBOSE_C_S 1 CACHE STRING "verbose mode" FORCE)
endif(VCS)

if(WIN32)
    set(CMAKE_DEBUG_POSTFIX "d")
    if(MSVC)
        # controll folders in msvc projects
        include(ProjectSourceGroup)
        if(NOT GROUP_CODE)
            #set(GROUP_CODE split) #cmake default
            set(GROUP_CODE flat)
        endif(NOT GROUP_CODE)
    endif(MSVC)
endif(WIN32)

```



```

        ADD_DEFINITIONS(-D_CRT_SECURE_NO_DEPRECATED -
D_CRT_NONSTDC_NO_DEPRECATED)

        # Use the highest warning level
        if (WALL)
            set(WALL 1 CACHE STRING "all warnings" FORCE)
            set(CMAKE_CXX_WARNING_LEVEL 4 CACHE STRING "warning
level" FORCE)

            if(CMAKE_CXX_FLAGS MATCHES "/W[0-4]")
                STRING(REGEX REPLACE "/W[0-4]" "/W4"
CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS}")
            else(CMAKE_CXX_FLAGS MATCHES "/W[0-4]")
                SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} /W4")
            endif(CMAKE_CXX_FLAGS MATCHES "/W[0-4]")

            if(CMAKE_C_FLAGS MATCHES "/W[0-4]")
                STRING(REGEX REPLACE "/W[0-4]" "/W4"
CMAKE_C_FLAGS "${CMAKE_C_FLAGS}")
            else(CMAKE_C_FLAGS MATCHES "/W[0-4]")
                SET(CMAKE_C_FLAGS "${CMAKE_C_FLAGS} /W4")
            endif(CMAKE_C_FLAGS MATCHES "/W[0-4]")
        else (WALL)
            set(CMAKE_CXX_WARNING_LEVEL 3 CACHE STRING "warning
level" FORCE)
        endif (WALL)

        SET(MSVC_W_ERROR " /we4028 /we4013 /we4133 /we4047
/we4031 /we4002 /we4003 /we4114")
        SET(MSVC_W_DISABLE " /wd4127 /wd4090 /wd4101 /wd4244")

        SET(CMAKE_C_FLAGS_DEBUG "${CMAKE_C_FLAGS_DEBUG}
/FIconfig.h ${MSVC_W_ERROR} ${MSVC_W_DISABLE}")
        SET(CMAKE_C_FLAGS_RELEASE "${CMAKE_C_FLAGS_RELEASE}
/FIconfig.h ${MSVC_W_ERROR} ${MSVC_W_DISABLE}")
        endif(MSVC)
    endif(WIN32)

    if (UNIX AND NOT DBUS_DISABLE_ASSERTS)
        # required for backtrace
        SET(CMAKE_C_FLAGS_DEBUG "${CMAKE_C_FLAGS_DEBUG} -Wl,--export-
dynamic")
        SET(CMAKE_C_FLAGS_RELEASE "${CMAKE_C_FLAGS_RELEASE} -Wl,--export-
dynamic")
        add_definitions(-DDBUS_BUILT_R_DYNAMIC)
    endif (UNIX AND NOT DBUS_DISABLE_ASSERTS)

    SET(CMAKE_C_FLAGS_DEBUG "${CMAKE_C_FLAGS_DEBUG} -D_DEBUG")

#####
###

```

```

# Windows CE (>= 5.0.0)
#
# WinCE support now relies on the presence of platform files, found in
cmake/modules/platform
# Cmake 2.8.0 doesn't include WinCE platform files by default, but
working ones can be found
# on CMake's bugtracker :
# http://public.kitware.com/Bug/view.php?id=7919
#
# for cmake 2.8.0 get the following patch only :
#
http://public.kitware.com/Bug/file_download.php?file_id=2944&type=bug
#
# after applying the patch, you can enable the WinCE build by
specifying :
# cmake [...] -DCMAKE_SYSTEM_NAME=WinCE -DCMAKE_SYSTEM_VERSION=X.XX
# (where X.XX is your actual WinCE version, e.g. 5.02 for Windows
Mobile 6)
#
# Note that you should have a proper cross-compilation environment set
up prior to running
# cmake, ie. the PATH, INCLUDE and LIB env vars pointing to your CE
SDK/toolchain.
#
if(WINCE)

MESSAGE("Building for WinCE (${CMAKE_SYSTEM_VERSION})")

endif(WINCE)
#####
###

ENABLE_TESTING()

#####
###
# Disallow in-source build
#macro_ensure_out_of_source_build("dbus requires an out of source
build. Please create a separate build directory and run 'cmake
path_to_dbus [options]' there.")

# ... and warn in case of an earlier in-source build
#set(generatedFileInSourceDir EXISTS ${dbus_SOURCE_DIR}/config.h)
#if(${generatedFileInSourceDir})
# message(STATUS "config.h exists in your source directory.")
#endif(${generatedFileInSourceDir})
#####
###

if (WIN32 OR CYGWIN)
    set (LIBRARY_OUTPUT_PATH ${CMAKE_BINARY_DIR}/bin)

```

```

else (WIN32 OR CYGWIN)
    set (LIBRARY_OUTPUT_PATH ${CMAKE_BINARY_DIR}/lib)
endif (WIN32 OR CYGWIN)

set (EXECUTABLE_OUTPUT_PATH ${CMAKE_BINARY_DIR}/bin)

# for including config.h and for includes like <dir/foo.h>
include_directories( ${CMAKE_SOURCE_DIR}/.. ${CMAKE_BINARY_DIR}
${CMAKE_INCLUDE_PATH} )

# linker search directories
link_directories( ${DBUS_LIB_DIR} ${LIBRARY_OUTPUT_PATH} )
include_directories( ${CMAKE_LIBRARY_PATH} )

set(DBUS_INCLUDES)

ENABLE_TESTING()

##### command line options #####
# TODO: take check from configure.in

#AC_ARG_ENABLE(tests, AS_HELP_STRING([--enable-tests],[enable unit
test code]),enable_tests=$enableval,enable_tests=$USE_MAINTAINER_MODE)
option (DBUS_BUILD_TESTS "enable unit test code" ON)

if(DBUS_BUILD_TESTS)
    add_definitions(-DDBUS_BUILD_TESTS -DDBUS_ENABLE_EMBEDDED_TESTS)
endif(DBUS_BUILD_TESTS)

option (DBUS_USE_OUTPUT_DEBUG_STRING "enable win32 debug port for
message output" OFF)
if(DBUS_USE_OUTPUT_DEBUG_STRING)
    add_definitions(-DDBUS_USE_OUTPUT_DEBUG_STRING)
endif(DBUS_USE_OUTPUT_DEBUG_STRING)

if(WIN32)
    # win32 dbus service support - this support is not complete
    option (DBUS_SERVICE "enable dbus service installer" OFF)
endif(WIN32)

#AC_ARG_ENABLE(ansi, AS_HELP_STRING([--enable-ansi],[enable -ansi -
pedantic gcc flags]),enable_ansi=$enableval,enable_ansi=no)
option (DBUS_ENABLE_ANSI "enable -ansi -pedantic gcc flags" OFF)
if(DBUS_ENABLE_ANSI)
    if(NOT MSVC)
        add_definitions(-ansi -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE
-pedantic)
    else(NOT MSVC)
        add_definitions(-Za -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -
W4)
    endif(NOT MSVC)
endif(DBUS_ENABLE_ANSI)

```

```

#AC_ARG_ENABLE(verbose-mode, AS_HELP_STRING([--enable-verbose-
mode],[support verbose debug
mode]),enable_verbose_mode=$enableval,enable_verbose_mode=$USE_MAINTAI
NER_MODE)
option (DBUS_ENABLE_VERBOSE_MODE "support verbose debug mode" ON)

#AC_ARG_ENABLE(checks, AS_HELP_STRING([--enable-checks],[include
sanity checks on public
API]),enable_checks=$enableval,enable_checks=yes)
option (DBUS_DISABLE_CHECKS "Disable public API sanity checking" OFF)

if(NOT MSVC)
    #AC_ARG_ENABLE(gcov, AS_HELP_STRING([--enable-gcov],[compile with
coverage profiling instrumentation (gcc
only)]),enable_gcov=$enableval,enable_gcov=no)
    option (DBUS_GCOV_ENABLED "compile with coverage profiling
instrumentation (gcc only)" OFF)
    if(DBUS_GCOV_ENABLED)
        add_definitions(-fprofile-arcs -ftest-coverage)
        # FIXME!!!!
        ## remove optimization
        # CFLAGS=`echo "$CFLAGS" | sed -e 's/-O[0-9]*//g'`
    endif(DBUS_GCOV_ENABLED)
endif(NOT MSVC)

#AC_ARG_ENABLE(abstract-sockets, AS_HELP_STRING([--enable-abstract-
sockets],[use abstract socket namespace (linux
only)]),enable_abstract_sockets=$enableval,enable_abstract_sockets=aut
o)
# -> moved before include(ConfigurationChecks.cmake)

#AC_ARG_ENABLE(selinux, AS_HELP_STRING([--enable-selinux],[build with
SELinux support]),enable_selinux=$enableval,enable_selinux=auto)
#selinux missing

#AC_ARG_ENABLE(dnotify, AS_HELP_STRING([--enable-dnotify],[build with
dnotify support (linux
only)]),enable_dnotify=$enableval,enable_dnotify=auto)
if("${CMAKE_SYSTEM_NAME}" STREQUAL "Linux")
    option (DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX "build with dnotify
support (linux only)" ON) # add a check !
endif("${CMAKE_SYSTEM_NAME}" STREQUAL "Linux")

#AC_ARG_ENABLE(kqueue, AS_HELP_STRING([--enable-kqueue],[build with
kqueue support (FreeBSD
only)]),enable_kqueue=$enableval,enable_kqueue=auto)
#missing

#AC_ARG_ENABLE(console-owner-file, AS_HELP_STRING([--enable-console-
owner-file],[enable console owner

```

```

file)),enable_console_owner_file=$enablelevel,enable_console_owner_file=
auto)
STRING(TOUPPER ${CMAKE_SYSTEM_NAME} sysname)
if("${sysname}" MATCHES ".*SOLARIS.*")
    option (HAVE_CONSOLE_OWNER_FILE "enable console owner file
(solaris only)" ON)
    if(HAVE_CONSOLE_OWNER_FILE)
        set (DBUS_CONSOLE_OWNER_FILE "/dev/console" CACHE STRING
"Directory to check for console ownership")
    endif(HAVE_CONSOLE_OWNER_FILE)
endif("${sysname}" MATCHES ".*SOLARIS.*")

#AC_ARG_WITH(xml, AS_HELP_STRING([--with-xml=[libxml/expat]], [XML
library to use]))
if(NOT LIBXML2_FOUND AND NOT LIBEXPAT_FOUND)
    message(FATAL "Neither expat nor libxml2 found!")
endif(NOT LIBXML2_FOUND AND NOT LIBEXPAT_FOUND)

if(DBUS_USE_EXPAT)
    SET(XML_LIB "Expat")
    SET(XML_LIBRARY      ${LIBEXPAT_LIBRARIES})
    SET(XML_INCLUDE_DIR ${LIBEXPAT_INCLUDE_DIR})
else(DBUS_USE_EXPAT)
    SET(XML_LIB "LibXML2")
    SET(XML_LIBRARY      ${LIBXML2_LIBRARIES})
    SET(XML_INCLUDE_DIR ${LIBXML2_INCLUDE_DIR})
endif(DBUS_USE_EXPAT)

#AC_ARG_WITH(init-scripts, AS_HELP_STRING([--with-init-
scripts=[redhat]], [Style of init scripts to install]))
#AC_ARG_WITH(session-socket-dir, AS_HELP_STRING([--with-session-
socket-dir=[dirname]], [Where to put sockets for the per-login-session
message bus]))
#AC_ARG_WITH(test-socket-dir, AS_HELP_STRING([--with-test-socket-
dir=[dirname]], [Where to put sockets for make check]))
#AC_ARG_WITH(system-pid-file, AS_HELP_STRING([--with-system-pid-
file=[pidfile]], [PID file for systemwide daemon]))
#AC_ARG_WITH(system-socket, AS_HELP_STRING([--with-system-
socket=[filename]], [UNIX domain socket for systemwide daemon]))
#AC_ARG_WITH(console-auth-dir, AS_HELP_STRING([--with-console-auth-
dir=[dirname]], [directory to check for console ownership]))
#AC_ARG_WITH(console-owner-file, AS_HELP_STRING([--with-console-owner-
file=[filename]], [file whose owner determines current console owner]))
#AC_ARG_WITH(dbus_user, AS_HELP_STRING([--with-dbus-user=<user>], [User
for running the DBUS daemon (messagebus)]))
#AC_ARG_WITH(dbus_daemon_dir, AS_HELP_STRING([--with-dbus-
daemon_dir=[dirname]], [Directory for installing the DBUS daemon]))
# all missing or hardcoded for now

# 'hidden' ones
set(atomic_int OFF)

```

```

set(atomic_int486 OFF)
if(CMAKE_COMPILER_IS_GNUCC AND NOT DBUS_ENABLE_ANSI)
    FIND_PROGRAM(UNAME_EXECUTABLE
                 NAMES uname
                 PATHS /bin /usr/bin /usr/local/bin
                 c:/Programme/MSys/bin d:/Programme/MSys/bin)

    if(UNAME_EXECUTABLE)
        EXECUTE_PROCESS(COMMAND ${UNAME_EXECUTABLE} "-m"
                        OUTPUT_VARIABLE UNAME_OUTPUT)

        if("UNAME_OUTPUT" MATCHES "^.*[i0123]86.*$")
            set(atomic_int ON)
        else("UNAME_OUTPUT" MATCHES "^.*[i0123]86.*$")
            if("UNAME_OUTPUT" MATCHES "^.*[i?86.*$")
                set(atomic_int ON)
                set(atomic_int_486 ON)
            endif("UNAME_OUTPUT" MATCHES "^.*[i?86.*$")
        endif("UNAME_OUTPUT" MATCHES "^.*[i0123]86.*$")
    endif(UNAME_EXECUTABLE)
endif(CMAKE_COMPILER_IS_GNUCC AND NOT DBUS_ENABLE_ANSI)

set (DBUS_HAVE_ATOMIC_INT ${atomic_int} CACHE STRING "Some atomic
integer implementation present")
set (DBUS_USE_ATOMIC_INT_486 ${atomic_int_486} CACHE STRING "Use
atomic integer implementation for 486")

if(X11_FOUND)
    option (DBUS_BUILD_X11 "Build with X11 autolaunch support " ON)
endif(X11_FOUND)

# test binary names
if (WIN32)
    # Automake calls this EXEEXT, and CMake doesn't have a standard
name
    # for it; follow Automake's naming convention so we can share .in
files
    set (EXEEXT ".exe")
endif(WIN32)

if (MSVC_IDE)
    if(CMAKE_BUILD_TYPE MATCHES Debug)
        set(IDE_BIN /Debug )
        message(STATUS)
        message(STATUS "Visual Studio: test programs will only work
with 'Debug' configuration!")
        message(STATUS "To run tests with 'Release' configuration
use -DCMAKE_BUILD_TYPE=Release")
        message(STATUS "Add '..\\..\\test\\data' to the command
line option of the test programs")
        message(STATUS)
    else(CMAKE_BUILD_TYPE MATCHES Debug)

```

```

        set(IDE_BIN /Release)
        message(STATUS)
        message(STATUS "Visual Studio: test programs will only work
with 'Release' configuration!")
        message(STATUS "To run tests with 'Debug' configuration use
-DCMAKE_BUILD_TYPE=Debug")
        message(STATUS "Add '..\\..\\test\\data' to the command
line option of the test programs")
        message(STATUS)
    endif(CMAKE_BUILD_TYPE MATCHES Debug)
    set (TEST_PATH_FORCE FORCE)
    FILE(REMOVE ${CMAKE_BINARY_DIR}/data/dbus-1/services)
endif (MSVC_IDE)

#### Find socket directories
if (NOT $ENV{TMPDIR} STREQUAL "")
    set (DBUS_SESSION_SOCKET_DIR $ENV{TMPDIR})
else (NOT $ENV{TMPDIR} STREQUAL "")
    if (NOT $ENV{TEMP} STREQUAL "")
        set (DBUS_SESSION_SOCKET_DIR $ENV{TEMP})
    else (NOT $ENV{TEMP} STREQUAL "")
        if (NOT $ENV{TMP} STREQUAL "")
            set (DBUS_SESSION_SOCKET_DIR $ENV{TMP})
        else (NOT $ENV{TMP} STREQUAL "")
            if (WIN32)
                #Should never happen, both TMP and TEMP seem always set
on Windows
                message(FATAL "Could not determine a usable temporary
directory")
            else(WIN32)
                set (DBUS_SESSION_SOCKET_DIR /tmp)
            endif(WIN32)
        endif (NOT $ENV{TMP} STREQUAL "")
    endif (NOT $ENV{TEMP} STREQUAL "")
endif (NOT $ENV{TMPDIR} STREQUAL "")

#AC_ARG_WITH(test-socket-dir, AS_HELP_STRING([--with-test-socket-
dir=[dirname]], [Where to put sockets for make check]))

#AC_ARG_WITH(system-pid-file, AS_HELP_STRING([--with-system-pid-
file=[pidfile]], [PID file for systemwide daemon]))

#if ! test -z "$with_system_pid_file"; then
#   DBUS_SYSTEM_PID_FILE=$with_system_pid_file
#elif test x$operating_system = xredhat ; then
#   DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/messagebus.pid
#else
#   DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/dbus/pid
#fi
# TODO: fix redhet
if (WIN32)
    # bus-test expects a non empty string

```

```

    set (DBUS_SYSTEM_PID_FILE "/dbus-pid")
else (WIN32)
    set (DBUS_SYSTEM_PID_FILE ${EXPANDED_LOCALSTATEDIR}/run/dbus/pid)
endif (WIN32)

#AC_ARG_WITH(system-socket, AS_HELP_STRING([--with-system-socket=[filename]], [UNIX domain socket for systemwide daemon]))

#AC_ARG_WITH(console-auth-dir, AS_HELP_STRING([--with-console-auth-dir=[dirname]], [directory to check for console ownership]))

if (WIN32)
    set (DBUS_CONSOLE_AUTH_DIR "")
else (WIN32)
    set (DBUS_CONSOLE_AUTH_DIR "/var/run/console/")
endif (WIN32)

#AC_ARG_WITH(dbus_user, AS_HELP_STRING([--with-dbus-user=<user>], [User for running the DBUS daemon (messagebus)]))

set (DBUS_USER )

if (WIN32)
    set (DBUS_SYSTEM_BUS_DEFAULT_ADDRESS "nonce-tcp:" CACHE STRING
"system bus default address")
    set (DBUS_SESSION_BUS_DEFAULT_ADDRESS "nonce-tcp:" CACHE STRING
"session bus default address")

    set (DBUS_SYSTEM_CONFIG_FILE "etc/dbus-1/system.conf")
    set (DBUS_SESSION_CONFIG_FILE "etc/dbus-1/session.conf")
    # bus-test expects a non empty string
    set (DBUS_USER "Administrator")
else (WIN32)
    set (DBUS_SYSTEM_BUS_DEFAULT_ADDRESS "unix:tmpdir=" CACHE STRING
"system bus default address")
    set (DBUS_SESSION_BUS_DEFAULT_ADDRESS
"unix:path=${DBUS_SESSION_SOCKET_DIR}" CACHE STRING "session bus
default address")
    set (sysconfdir "")
    set (configdir ${sysconfdir}/dbus-1 )
    set (DBUS_SYSTEM_CONFIG_FILE ${configdir}/system.conf)
    set (DBUS_SESSION_CONFIG_FILE ${configdir}/session.conf)
    set (DBUS_USER "root")
endif (WIN32)

set (DBUS_DAEMON_NAME "dbus-daemon" CACHE STRING "The name of the dbus
daemon executable")

##### create config.h #####

#include(ConfigurationChecks.cmake)

```



```

# better use flags for gcc
if (MINGW)
    set (HAVE_GNUC_VARARGS 1)
endif(MINGW)

# compiler definitions
add_definitions(-DHAVE_CONFIG_H=1)
add_definitions(${DBUS_BUS_CFLAGS})

if (DBUS_BUILD_TESTS)
    # set variables used for the .in files (substituted by
configure_file) in test/data:
    set(DBUS_TEST_EXEC ${EXECUTABLE_OUTPUT_PATH}${IDE_BIN})
    set(DBUS_TEST_DATA ${CMAKE_BINARY_DIR}/test/data)
    set(TEST_SOCKET_DIR ${DBUS_SESSION_SOCKET_DIR} )
    set(TEST_LAUNCH_HELPER_BINARY ${EXECUTABLE_OUTPUT_PATH}/dbus-
daemon-launch-helper-test)
    if (UNIX)
        set (TEST_LISTEN "unix:tmpdir=${TEST_SOCKET_DIR}")
    endif (UNIX)
    if (WIN32)
        set (TEST_LISTEN "tcp:host=localhost")
    endif (WIN32)
endif (DBUS_BUILD_TESTS)

set(DBUS_LIBRARIES dbus-1)
set(DBUS_INTERNAL_LIBRARIES dbus-internal)

# settings for building and using static internal lib
# important note: DBUS_INTERNAL_XXXXX_DEFINITIONS must *not* be set
when building dbus-1 library
set (DBUS_INTERNAL_ADD_LIBRARY_OPTIONS STATIC)
set (DBUS_INTERNAL_LIBRARY_DEFINITIONS "-DDBUS_STATIC_BUILD")
set (DBUS_INTERNAL_CLIENT_DEFINITIONS "-DDBUS_STATIC_BUILD")

configure_file(${CMAKE_CURRENT_SOURCE_DIR}/config.h.cmake
${CMAKE_CURRENT_BINARY_DIR}/config.h )

if (WIN32)
configure_file(${CMAKE_CURRENT_SOURCE_DIR}/dbus-env.bat.cmake
${CMAKE_BINARY_DIR}/bin/dbus-env.bat )
install_files(/bin FILES ${CMAKE_BINARY_DIR}/bin/dbus-env.bat)
endif()

add_definitions(-DHAVE_CONFIG_H=1)

##### subdirs #####

add_subdirectory( dbus )
add_subdirectory( bus )

```

```

if (DBUS_BUILD_TESTS)
    add_subdirectory( test )
endif (DBUS_BUILD_TESTS)
add_subdirectory( tools )
add_subdirectory( doc )

OPTION(DBUS_INSTALL_SYSTEM_LIBS "install required system libraries"
OFF)
MESSAGE(" ")
MESSAGE("set -DDBUS_INSTALL_SYSTEM_LIBS=1 to install runtime libraries
too")
MESSAGE("set DBUSDIR (environment or cmake option) to overwrite the
default install directory ")
MESSAGE(" ")
MESSAGE(" ")
GET_FILENAME_COMPONENT(C_COMPILER ${CMAKE_C_COMPILER} NAME)
GET_FILENAME_COMPONENT(CXX_COMPILER ${CMAKE_CXX_COMPILER} NAME)

message("                D-BUS ${DBUS_VERSION}
")
message("                =====
")
message("
")
message("    install prefix:                ${prefix}
")
message("    install exec_prefix:           ${exec_prefix}
")
message("    install libdir:                ${EXPANDED_LIBDIR}
")
message("    install bindir:                ${EXPANDED_BINDIR}
")
message("    install sysconfdir:           ${EXPANDED_SYSCONFDIR}
")
#message("    install localstatedir:       ${EXPANDED_LOCALSTATEDIR}
")
message("    install datadir:              ${EXPANDED_DATADIR}
")
message("    source code location:         ${DBUS_SOURCE_DIR}
")
message("    build dir:                    ${CMAKE_BINARY_DIR}
")
message("    c compiler:                   ${C_COMPILER}
")
message("    cflags:                        ${CMAKE_C_FLAGS}
")
message("    cflags debug:                 ${CMAKE_C_FLAGS_DEBUG}
")
message("    cflags release:               ${CMAKE_C_FLAGS_RELEASE}
")

```

```

message("      cxx compiler:                ${CXX_COMPILER}")
message("      cxxflags:                       ${CMAKE_CXX_FLAGS}")
message("      cxxflags debug:                  ${CMAKE_CXX_FLAGS_DEBUG}")
message("      cxxflags release:                ${CMAKE_CXX_FLAGS_RELEASE}")
message("      64-bit int:                      ${DBUS_INT64_TYPE}")
message("      32-bit int:                      ${DBUS_INT32_TYPE}")
message("      16-bit int:                      ${DBUS_INT16_TYPE}")
message("      Doxygen:                         ${DOXYGEN}")
message("      Docbook Generator:              ${DOCBOOK_GENERATOR_NAME}")

#message("      Maintainer mode:                ${USE_MAINTAINER_MODE}")
message("      gcc coverage profiling:          ${DBUS_GCOV_ENABLED}")
message("      Building unit tests:             ${DBUS_BUILD_TESTS}")
message("      Building verbose mode:           ${DBUS_ENABLE_VERBOSE_MODE}")
message("      Building w/o assertions:         ${DBUS_DISABLE_ASSERTS}")
message("      Building w/o checks:             ${DBUS_DISABLE_CHECKS}")
message("      Building bus stats API:          ${DBUS_ENABLE_STATS}")
message("      installing system libs:          ${DBUS_INSTALL_SYSTEM_LIBS}")
#message("      Building SELinux support:         ${have_selinux}")
#message("      Building dnotify support:        ${have_dnotify}")
message("      Building Doxygen docs:           ${DBUS_ENABLE_DOXYGEN_DOCS}")
message("      Building XML docs:               ${DBUS_ENABLE_XML_DOCS}")
#message("      Gettext libs (empty OK):         ${INTLLIBS}")
message("      Using XML parser:                ${XML_LIB}")
message("      Daemon executable name:         ${DBUS_DAEMON_NAME}")
if (WIN32)
message("      System bus address:             ${DBUS_SYSTEM_BUS_DEFAULT_ADDRESS} ")

```

```

message("      Session bus address:
${DBUS_SESSION_BUS_DEFAULT_ADDRESS} ")
else (WIN32)
#message("      Init scripts style:      ${with_init_scripts}
")
#message("      Abstract socket names:    ${have_abstract_sockets}
")
message("      System bus socket:      ${DBUS_SYSTEM_SOCKET}
")
message("      System bus address:
${DBUS_SYSTEM_BUS_DEFAULT_ADDRESS} ")
message("      System bus PID file:      ${DBUS_SYSTEM_PID_FILE}
")
message("      Session bus socket dir:    ${DBUS_SESSION_SOCKET_DIR}
")
message("      Console auth dir:          ${DBUS_CONSOLE_AUTH_DIR}
")
message("      System bus user:            ${DBUS_USER}
")
message("      'make check' socket dir:    ${TEST_SOCKET_DIR}
")
endif (WIN32)
message("      Test listen address:      ${TEST_LISTEN}
")
if (MSVC)
message("      build timestamp:          ${DBUS_BUILD_TIMESTAMP}
")
endif (MSVC)

MESSAGE(" ")
if (DBUS_BUILD_TESTS)
    message("NOTE: building with unit tests increases the size of the
installed library and renders it insecure.")
endif(DBUS_BUILD_TESTS)

if (DBUS_BUILD_TESTS AND DBUS_DISABLE_ASSERTS)
    message("NOTE: building with unit tests but without assertions
means tests may not properly report failures (this configuration is
only useful when doing something like profiling the tests)")
endif(DBUS_BUILD_TESTS AND DBUS_DISABLE_ASSERTS)

if (DBUS_GCOV_ENABLED)
    message("NOTE: building with coverage profiling is definitely for
developers only.")
endif(DBUS_GCOV_ENABLED)

if (DBUS_ENABLE_VERBOSE_MODE)
    message("NOTE: building with verbose mode increases library size,
may slightly increase security risk, and decreases performance.")
endif(DBUS_ENABLE_VERBOSE_MODE)

if(NOT DBUS_DISABLE_ASSERTS)

```

```

    message("NOTE: building with assertions increases library size and
decreases performance.")
endif(NOT DBUS_DISABLE_ASSERTS)

if (DBUS_DISABLE_CHECKS)
    message("NOTE: building without checks for arguments passed to
public API makes it harder to debug apps using D-BUS, but will
slightly decrease D-BUS library size and very slightly improve
performance.")
endif(DBUS_DISABLE_CHECKS)
MESSAGE(" ")

INCLUDE(modules/CPackInstallConfig.cmake)

add_custom_target(help-options
    cmake -LH
    WORKING_DIRECTORY ${CMAKE_BINARY_DIR}
)

File = CMakeLists.txt.~3~

project(dbus-lib)

SET(DBUS_DIR ${CMAKE_SOURCE_DIR}/../dbus)

configure_file(${DBUS_DIR}/dbus-arch-deps.h.in
${CMAKE_CURRENT_BINARY_DIR}/dbus-arch-deps.h )

add_definitions(-DDBUS_COMPILATION)

set (dbusinclude_HEADERS
    ${DBUS_DIR}/dbus.h
    ${DBUS_DIR}/dbus-address.h
    ${DBUS_DIR}/dbus-bus.h
    ${DBUS_DIR}/dbus-connection.h
    ${DBUS_DIR}/dbus-errors.h
    ${DBUS_DIR}/dbus-macros.h
    ${DBUS_DIR}/dbus-memory.h
    ${DBUS_DIR}/dbus-message.h
    ${DBUS_DIR}/dbus-misc.h
    ${DBUS_DIR}/dbus-pending-call.h
    ${DBUS_DIR}/dbus-protocol.h
    ${DBUS_DIR}/dbus-server.h
    ${DBUS_DIR}/dbus-shared.h
    ${DBUS_DIR}/dbus-signature.h
    ${DBUS_DIR}/dbus-syntax.h
    ${DBUS_DIR}/dbus-threads.h
    ${DBUS_DIR}/dbus-types.h
    dbus-arch-deps.h
)

```

```

### source code that goes in the installed client library
### and is specific to library functionality
set (DBUS_LIB_SOURCES
    ${DBUS_DIR}/dbus-address.c
    ${DBUS_DIR}/dbus-auth.c
    ${DBUS_DIR}/dbus-bus.c
    ${DBUS_DIR}/dbus-connection.c
    ${DBUS_DIR}/dbus-credentials.c
    ${DBUS_DIR}/dbus-errors.c
    ${DBUS_DIR}/dbus-keyring.c
    ${DBUS_DIR}/dbus-marshal-header.c
    ${DBUS_DIR}/dbus-marshal-byteswap.c
    ${DBUS_DIR}/dbus-marshal-recursive.c
    ${DBUS_DIR}/dbus-marshal-validate.c
    ${DBUS_DIR}/dbus-message.c
    ${DBUS_DIR}/dbus-misc.c
    ${DBUS_DIR}/dbus-nonce.c
    ${DBUS_DIR}/dbus-object-tree.c
    ${DBUS_DIR}/dbus-pending-call.c
    ${DBUS_DIR}/dbus-resources.c
    ${DBUS_DIR}/dbus-server.c
    ${DBUS_DIR}/dbus-server-socket.c
    ${DBUS_DIR}/dbus-server-debug-pipe.c
    ${DBUS_DIR}/dbus-sha.c
    ${DBUS_DIR}/dbus-signature.c
    ${DBUS_DIR}/dbus-syntax.c
    ${DBUS_DIR}/dbus-timeout.c
    ${DBUS_DIR}/dbus-threads.c
    ${DBUS_DIR}/dbus-transport.c
    ${DBUS_DIR}/dbus-transport-socket.c
    ${DBUS_DIR}/dbus-watch.c
)

if (UNIX)
    set (DBUS_LIB_SOURCES ${DBUS_LIB_SOURCES}
        ${DBUS_DIR}/dbus-transport-unix.c
        ${DBUS_DIR}/dbus-server-unix.c
    )
else (UNIX)
    set (DBUS_LIB_SOURCES ${DBUS_LIB_SOURCES}
        ${DBUS_DIR}/dbus-transport-win.c
        ${DBUS_DIR}/dbus-server-win.c
    )
endif (UNIX)

set (DBUS_LIB_HEADERS
    ${DBUS_DIR}/dbus-auth.h
    ${DBUS_DIR}/dbus-connection-internal.h
    ${DBUS_DIR}/dbus-credentials.h
    ${DBUS_DIR}/dbus-keyring.h

```

```

    ${DBUS_DIR}/dbus-marshal-header.h
    ${DBUS_DIR}/dbus-marshal-byteswap.h
    ${DBUS_DIR}/dbus-marshal-recursive.h
    ${DBUS_DIR}/dbus-marshal-validate.h
    ${DBUS_DIR}/dbus-message-internal.h
    ${DBUS_DIR}/dbus-message-private.h
    ${DBUS_DIR}/dbus-misc.h
    ${DBUS_DIR}/dbus-object-tree.h
    ${DBUS_DIR}/dbus-protocol.h
    ${DBUS_DIR}/dbus-resources.h
    ${DBUS_DIR}/dbus-server-debug-pipe.h
    ${DBUS_DIR}/dbus-server-protected.h
    ${DBUS_DIR}/dbus-server-unix.h
    ${DBUS_DIR}/dbus-sha.h
    ${DBUS_DIR}/dbus-timeout.h
    ${DBUS_DIR}/dbus-threads.h
    ${DBUS_DIR}/dbus-threads-internal.h
    ${DBUS_DIR}/dbus-transport.h
    ${DBUS_DIR}/dbus-transport-protected.h
    ${DBUS_DIR}/dbus-watch.h
    ${CMAKE_BINARY_DIR}/config.h
)
if (UNIX)
    set (DBUS_LIB_HEADERS ${DBUS_LIB_HEADERS}
        ${DBUS_DIR}/dbus-transport-unix.h
    )
else (UNIX)
    set (DBUS_LIB_HEADERS ${DBUS_LIB_HEADERS}
        ${DBUS_DIR}/dbus-transport-win.h
    )
endif (UNIX)

### source code that goes in the installed client library
### AND is generic utility functionality used by the
### daemon or test programs (all symbols in here should
### be underscore-prefixed)
set (DBUS_SHARED_SOURCES
    ${DBUS_DIR}/dbus-dataslot.c
    ${DBUS_DIR}/dbus-file.c
    ${DBUS_DIR}/dbus-hash.c
    ${DBUS_DIR}/dbus-internals.c
    ${DBUS_DIR}/dbus-list.c
    ${DBUS_DIR}/dbus-marshal-basic.c
    ${DBUS_DIR}/dbus-memory.c
    ${DBUS_DIR}/dbus-mempool.c
    ${DBUS_DIR}/dbus-string.c
    ${DBUS_DIR}/dbus-sysdeps.c
    ${DBUS_DIR}/dbus-pipe.c
)

set (DBUS_SHARED_HEADERS

```

```

    ${DBUS_DIR}/dbus-dataslot.h
    ${DBUS_DIR}/dbus-file.h
    ${DBUS_DIR}/dbus-hash.h
    ${DBUS_DIR}/dbus-internals.h
    ${DBUS_DIR}/dbus-list.h
    ${DBUS_DIR}/dbus-marshal-basic.h
    ${DBUS_DIR}/dbus-mempool.h
    ${DBUS_DIR}/dbus-string.h
    ${DBUS_DIR}/dbus-string-private.h
    ${DBUS_DIR}/dbus-pipe.h
    ${DBUS_DIR}/dbus-sysdeps.h
)

### source code that is generic utility functionality used
### by the bus daemon or test apps, but is NOT included
### in the D-BUS client library (all symbols in here
### should be underscore-prefixed but don't really need
### to be unless they move to DBUS_SHARED_SOURCES later)
set (DBUS_UTIL_SOURCES
    ${DBUS_DIR}/dbus-auth-script.c
    ${DBUS_DIR}/dbus-auth-util.c
    ${DBUS_DIR}/dbus-credentials-util.c
    ${DBUS_DIR}/dbus-mainloop.c
    ${DBUS_DIR}/dbus-marshal-byteswap-util.c
    ${DBUS_DIR}/dbus-marshal-recursive-util.c
    ${DBUS_DIR}/dbus-marshal-validate-util.c
    ${DBUS_DIR}/dbus-message-factory.c
    ${DBUS_DIR}/dbus-message-util.c
    ${DBUS_DIR}/dbus-shell.c
    ${DBUS_DIR}/dbus-socket-set.c
    ${DBUS_DIR}/dbus-socket-set-poll.c
    ${DBUS_DIR}/dbus-string-util.c
    ${DBUS_DIR}/dbus-sysdeps-util.c
)

if (DBUS_BUILD_TESTS)
    set (DBUS_UTIL_SOURCES
        ${DBUS_UTIL_SOURCES}
        ${DBUS_DIR}/dbus-test.c
    )
endif (DBUS_BUILD_TESTS)

set (DBUS_UTIL_HEADERS
    ${DBUS_DIR}/dbus-auth-script.h
    ${DBUS_DIR}/dbus-mainloop.h
    ${DBUS_DIR}/dbus-message-factory.h
    ${DBUS_DIR}/dbus-shell.h
    ${DBUS_DIR}/dbus-socket-set.h
    ${DBUS_DIR}/dbus-spawn.h
    ${DBUS_DIR}/dbus-test.h
)

```



```

### platform specific settings
if (WIN32)
    set (DBUS_SHARED_SOURCES ${DBUS_SHARED_SOURCES}
        ${DBUS_DIR}/dbus-file-win.c
        ${DBUS_DIR}/dbus-sysdeps-win.c
        ${DBUS_DIR}/dbus-pipe-win.c
        ${DBUS_DIR}/dbus-sysdeps-thread-win.c
    )
    set (DBUS_SHARED_HEADERS ${DBUS_SHARED_HEADERS}
        ${DBUS_DIR}/dbus-sockets-win.h
        ${DBUS_DIR}/dbus-sysdeps-win.h
    )
    set (DBUS_UTIL_SOURCES ${DBUS_UTIL_SOURCES}
        ${DBUS_DIR}/dbus-spawn-win.c
        ${DBUS_DIR}/dbus-sysdeps-util-win.c
    )
    if(WINCE)
        set (DBUS_SHARED_SOURCES ${DBUS_SHARED_SOURCES}
            ${DBUS_DIR}/dbus-sysdeps-wince-glue.c
        )
        set (DBUS_SHARED_HEADERS ${DBUS_SHARED_HEADERS}
            ${DBUS_DIR}/dbus-sysdeps-wince-glue.h
        )
    endif(WINCE)
else (WIN32)
    set (DBUS_SHARED_SOURCES ${DBUS_SHARED_SOURCES}
        ${DBUS_DIR}/dbus-file-unix.c
        ${DBUS_DIR}/dbus-pipe-unix.c
        ${DBUS_DIR}/dbus-sysdeps-unix.c
        ${DBUS_DIR}/dbus-sysdeps-pthread.c
        ${DBUS_DIR}/dbus-userdb.c
        ${DBUS_DIR}/sd-daemon.c
    )
    set (DBUS_SHARED_HEADERS ${DBUS_SHARED_HEADERS}
        ${DBUS_DIR}/dbus-server-unix.h
        ${DBUS_DIR}/dbus-transport-unix.h
        ${DBUS_DIR}/dbus-sysdeps-unix.h
        ${DBUS_DIR}/dbus-userdb.h
        ${DBUS_DIR}/sd-daemon.h
    )
    set (DBUS_UTIL_SOURCES ${DBUS_UTIL_SOURCES}
        ${DBUS_DIR}/dbus-spawn.c
        ${DBUS_DIR}/dbus-userdb-util.c
        ${DBUS_DIR}/dbus-sysdeps-util-unix.c
    )
endif (WIN32)

set(libdbus_SOURCES
    ${DBUS_LIB_SOURCES}
    ${DBUS_SHARED_SOURCES}
)

```

```

set(libdbus_HEADERS
    ${DBUS_LIB_HEADERS}
    ${DBUS_SHARED_HEADERS}
)
if(MSVC)
    set(BUILD_FILEVERSION
${DBUS_MAJOR_VERSION},${DBUS_MINOR_VERSION},${DBUS_MICRO_VERSION},${DB
US_PATCH_VERSION})
    set(BUILD_TIMESTAMP ${DBUS_BUILD_TIMESTAMP})

    configure_file(${DBUS_DIR}/versioninfo.rc.in
${CMAKE_CURRENT_BINARY_DIR}/versioninfo.rc)
    file(WRITE ${CMAKE_CURRENT_BINARY_DIR}/afxres.h "")
    list(APPEND libdbus_SOURCES versioninfo.rc)
    set_source_files_properties(versioninfo.rc COMPILE_FLAGS "-
D _LINE_=1")
endif(MSVC)

if(MSVC_IDE)
    project_source_group(${GROUP_CODE} DBUS_LIB_SOURCES
DBUS_LIB_HEADERS)
    project_source_group(${GROUP_CODE} DBUS_SHARED_SOURCES
DBUS_SHARED_HEADERS)
    project_source_group(${GROUP_CODE} DBUS_UTIL_SOURCES
DBUS_UTIL_SOURCES)
endif(MSVC_IDE)

### Client library

add_library(dbus-1 SHARED
    ${libdbus_SOURCES}
    ${libdbus_HEADERS}
)
if(WIN32)
    if(WINCE)
        target_link_libraries(dbus-1 ws2)
    else(WINCE)
        target_link_libraries(dbus-1 ws2_32 advapi32 netapi32)
    endif(WINCE)
else(WIN32)
    target_link_libraries(dbus-1 ${CMAKE_THREAD_LIBS_INIT})
endif(WIN32)

install_targets(/lib dbus-1 )
install_files(/include/dbus FILES ${dbusinclud_HEADERS})

### Internal library, used for the daemon, tools and tests, compiled
statically.

add_library(dbus-internal ${DBUS_INTERNAL_ADD_LIBRARY_OPTIONS}
    ${DBUS_LIB_SOURCES}
    ${DBUS_LIB_HEADERS}

```

```

        ${DBUS_SHARED_SOURCES}
        ${DBUS_SHARED_HEADERS}
        ${DBUS_UTIL_SOURCES}
        ${DBUS_UTIL_HEADERS}
    )
    target_link_libraries(dbus-internal)
    set_target_properties(dbus-internal PROPERTIES COMPILE_FLAGS
    ${DBUS_INTERNAL_LIBRARY_DEFINITIONS})
    if(WIN32)
        if(WINCE)
            target_link_libraries(dbus-internal ws2)
        else(WINCE)
            target_link_libraries(dbus-internal ws2_32 advapi32 netapi32)
        endif(WINCE)
    else(WIN32)
        target_link_libraries(dbus-internal ${CMAKE_THREAD_LIBS_INIT})
    endif(WIN32)

    if (DBUS_BUILD_TESTS)
        set (TESTS_ENVIRONMENT
        "DBUS_TEST_DATA=${CMAKE_SOURCE_DIR}/test/data
        DBUS_TEST_HOMEDIR=${CMAKE_BUILD_DIR}/dbus")
        ADD_EXECUTABLE(dbus-test ${CMAKE_SOURCE_DIR}/../dbus/dbus-test-
        main.c)
        target_link_libraries(dbus-test ${DBUS_INTERNAL_LIBRARIES})
        add_test(dbus-test ${EXECUTABLE_OUTPUT_PATH}/dbus-test
        ${CMAKE_SOURCE_DIR}/../test/data)
        set_target_properties(dbus-test PROPERTIES COMPILE_FLAGS
        ${DBUS_INTERNAL_CLIENT_DEFINITIONS})
    ENDIF (DBUS_BUILD_TESTS)

    if (UNIX)
    # set version info
    ENDIF (UNIX)

    ## mop up the gcov files
    #clean-local:
    #/bin/rm *.bb *.bbg *.da *.gcov .libs/*.da .libs/*.bbg || true

File = CMakeLists.txt.~4~

find_package(Doxygen)

if(DOXYGEN_EXECUTABLE)
    OPTION(DBUS_ENABLE_DOXYGEN_DOCS "build DOXYGEN documentation
    (requires Doxygen)" ON)
endif(DOXYGEN_EXECUTABLE)

if (DBUS_ENABLE_DOXYGEN_DOCS)

```

```

    set (top_srcdir ${CMAKE_SOURCE_DIR}/..)
    configure_file(${CMAKE_SOURCE_DIR}/Doxyfile.cmake
${CMAKE_BINARY_DIR}/Doxyfile )
    add_custom_target(doc
        COMMAND ${DOXYGEN_EXECUTABLE} ${CMAKE_BINARY_DIR}/Doxyfile
    )
endif (DBUS_ENABLE_DOXYGEN_DOCS)

#
# find docbook generator
#
find_program(MEINPROC4_EXECUTABLE meinproc4)

find_program(XMLTO_EXECUTABLE xmlto)

if (MEINPROC4_EXECUTABLE OR XMLTO_EXECUTABLE)
    OPTION(DBUS_ENABLE_XML_DOCS "build XML documentation (requires
xmlto or meinproc4)" ON)
endif (MEINPROC4_EXECUTABLE OR XMLTO_EXECUTABLE)

if (MEINPROC4_EXECUTABLE)
    set(DOCBOOK_GENERATOR_NAME "meinproc4" PARENT_SCOPE)
    set(DBUS_XML_DOCS_ENABLED 1)
    if(WIN32)
        get_filename_component(_a ${MEINPROC4_EXECUTABLE} PATH)
        get_filename_component(_meinproc_install_path ${_a} PATH)
    else(WIN32)
        set(_meinproc_install_path ${CMAKE_INSTALL_PREFIX})
    endif(WIN32)
    set(STYLESHEET
"${_meinproc_install_path}/share/apps/ksgmltools2/docbook/xsl/html/doc
book.xsl")
endif (MEINPROC4_EXECUTABLE)

if (XMLTO_EXECUTABLE)
    set (DOCBOOK_GENERATOR_NAME "xmlto" PARENT_SCOPE)
    set(DBUS_XML_DOCS_ENABLED 1)
    MESSAGE(STATUS "xmlto docbook generator found")
endif (XMLTO_EXECUTABLE)

if (DBUS_ENABLE_XML_DOCS)

macro (DOCBOOK _sources _options)
    get_filename_component(_infile ${_sources} ABSOLUTE)
    get_filename_component(_basename ${_infile} NAME_WE)
    set(_outfile ${CMAKE_CURRENT_BINARY_DIR}/${_basename}.html)

    if (EXISTS ${_sources})
        if (MEINPROC4_EXECUTABLE)
            ADD_CUSTOM_TARGET(${_basename}.html ALL

```

```

        ${MEINPROC4_EXECUTABLE} --stylesheet ${STYLESHEET} -o
${_outfile} ${_infile}
        DEPENDS ${_infile}
        WORKING_DIRECTORY ${CMAKE_CURRENT_BINARY_DIR}
    )
endif ()
if (XMLTO_EXECUTABLE)
    ADD_CUSTOM_TARGET(${_basename}.html ALL
        ${XMLTO_EXECUTABLE} -vv ${_options} ${_infile}
        DEPENDS ${_infile}
        WORKING_DIRECTORY ${CMAKE_CURRENT_BINARY_DIR}
    )
endif ()
install(FILES ${_outfile} DESTINATION share/doc/dbus)
else ()
    MESSAGE(STATUS "skipping xml doc generating for ${_infile},
file not found")
endif ()

```

```
endmacro (DOCBOOK)
```

```
### copy tests to builddir so that generated tests and static tests
### are all in one place.
```

```
### todo how to add more filetypes
```

```
MACRO (COPYDIR _src _type)
    FOREACH(FILE_TYPE ${_type})
        FOREACH(DIR ${_src})
            FILE(GLOB FILES
"${CMAKE_SOURCE_DIR}/../${DIR}/${FILE_TYPE}" )
            FILE(MAKE_DIRECTORY ${CMAKE_BINARY_DIR}/${DIR})
            FOREACH(FILE ${FILES})
                GET_FILENAME_COMPONENT(FILENAME ${FILE} NAME)
                SET (TARGET ${CMAKE_BINARY_DIR}/${DIR}/${FILENAME})
                configure_file(${FILE} ${TARGET} COPYONLY)
                IF (CONFIG_VERBOSE)
                    MESSAGE("FROM: ${FILE}\nTO: ${TARGET}\n")
                ENDIF (CONFIG_VERBOSE)
            ENDFOREACH (FILE)
        ENDFOREACH (DIR)
    ENDFOREACH (FILE_TYPE)
ENDMACRO (COPYDIR)

```

```
COPYDIR(doc *.png)
```

```
COPYDIR(doc *.svg)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/../doc/dbus-test-plan.xml html-nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/../doc/dbus-tutorial.xml html-nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/../doc/dbus-specification.xml html-
nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/../doc/dbus-faq.xml html-nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/bus/dbus-daemon.xml html-nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/tools/dbus-monitor.xml html-nochunks)
```

```
DOCBOOK(${CMAKE_SOURCE_DIR}/tools/dbus-send.xml html-nochunks)
DOCBOOK(${CMAKE_SOURCE_DIR}/tools/dbus-launch.xml html-nochunks)
```

```
#
# handle html index file
#
configure_file(${CMAKE_CURRENT_SOURCE_DIR}/index.html.cmake
${CMAKE_CURRENT_BINARY_DIR}/index.html )
install(FILES ${CMAKE_CURRENT_BINARY_DIR}/index.html DESTINATION
share/doc/dbus)
```

```
set (EXTRA_DIST
    ${CMAKE_SOURCE_DIR}/../doc/busconfig.dtd
    ${CMAKE_SOURCE_DIR}/../doc/introspect.dtd
    ${CMAKE_SOURCE_DIR}/../doc/introspect.xsl
)
```

```
install(FILES ${EXTRA_DIST} DESTINATION share/doc/dbus)
```

```
endif(DBUS_ENABLE_XML_DOCS)
```

```
File = CMakeLists.txt.~5~
```

```
project(test)
```

```
add_definitions(${DBUS_INTERNAL_CLIENT_DEFINITIONS})
```

```
add_library(dbus-testutils STATIC
    ${CMAKE_SOURCE_DIR}/../test/test-utils.h
    ${CMAKE_SOURCE_DIR}/../test/test-utils.c
)
```

```
target_link_libraries(dbus-testutils ${DBUS_INTERNAL_LIBRARIES})
```

```
add_subdirectory( name-test )
```

```
set (test-service_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-service.c
)
```

```
set (test-names_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-names.c
)
```

```
set (break_loader_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/break-loader.c
)
```

```
set (test-shell-service_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-shell-service.c
)
```

```

set (shell-test_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/shell-test.c
)

set (spawn-test_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/spawn-test.c
)

set (test-exit_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-exit.c
)

set (test-segfault_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-segfault.c
)

set (test-sleep-forever_SOURCES
    ${CMAKE_SOURCE_DIR}/../test/test-sleep-forever.c
)

add_executable(test-service ${test-service_SOURCES})
target_link_libraries(test-service dbus-testutils)

add_executable(test-names ${test-names_SOURCES})
target_link_libraries(test-names dbus-testutils)

add_executable(shell-test ${shell-test_SOURCES})
target_link_libraries(shell-test ${DBUS_INTERNAL_LIBRARIES})
ADD_TEST(shell-test ${EXECUTABLE_OUTPUT_PATH}/shell-test${EXEEXT})

add_executable(test-shell-service ${test-shell-service_SOURCES})
target_link_libraries(test-shell-service dbus-testutils)

add_executable(spawn-test ${spawn-test_SOURCES})
target_link_libraries(spawn-test ${DBUS_INTERNAL_LIBRARIES})

add_executable(test-exit ${test-exit_SOURCES})
target_link_libraries(test-exit ${DBUS_INTERNAL_LIBRARIES})

add_executable(test-segfault ${test-segfault_SOURCES})
target_link_libraries(test-segfault ${DBUS_INTERNAL_LIBRARIES})

add_executable(test-sleep-forever ${test-sleep-forever_SOURCES})
target_link_libraries(test-sleep-forever ${DBUS_INTERNAL_LIBRARIES})

### keep these in creation order, i.e. uppermost dirs first
set (TESTDIRS
    test/data
    test/data/valid-messages
    test/data/invalid-messages
    test/data/incomplete-messages

```

```

test/data/auth
test/data/sha-1
test/data/valid-config-files
test/data/valid-config-files/basic.d
test/data/valid-config-files/system.d
test/data/valid-config-files-system
test/data/valid-introspection-files
test/data/valid-messages
test/data/valid-service-files
test/data/valid-service-files-system
test/data/invalid-config-files
test/data/invalid-config-files-system
test/data/invalid-messages
test/data/invalid-service-files-system
test/data/equiv-config-files
test/data/equiv-config-files/basic
test/data/equiv-config-files/basic/basic.d
test/data/equiv-config-files/entities
test/data/equiv-config-files/entities/basic.d
)
set (CONFIG_VERBOSE 0)

### copy tests to builddir so that generated tests and static tests
### are all in one place.
MESSAGE(STATUS "Copying test files to test directory")
FOREACH(FILE_TYPE *.message *.message-raw *.auth-script *.sha1 *.txt
*.conf *.service)
    FOREACH(DIR ${TESTDIRS})
        FILE(GLOB FILES "${CMAKE_SOURCE_DIR}/../${DIR}/${FILE_TYPE}" )
        FILE(MAKE_DIRECTORY ${CMAKE_BINARY_DIR}/${DIR})
        FOREACH(FILE ${FILES})
            GET_FILENAME_COMPONENT(FILENAME ${FILE} NAME)
            SET (TARGET ${CMAKE_BINARY_DIR}/${DIR}/${FILENAME})
            configure_file(${FILE} ${TARGET} COPYONLY)
            IF (CONFIG_VERBOSE)
                MESSAGE("${FILE}")
            ENDIF (CONFIG_VERBOSE)
        ENDFOREACH(FILE)
    ENDFOREACH(DIR)
ENDFOREACH(FILE_TYPE)

### generate test files
MESSAGE(STATUS "Generating test files from templates into test
directory")

FOREACH(FILE_TYPE *.conf.in *.service.in)
    FOREACH(DIR ${TESTDIRS})
        FILE(GLOB FILES "${CMAKE_SOURCE_DIR}/../${DIR}/${FILE_TYPE}" )
        FILE(MAKE_DIRECTORY ${CMAKE_BINARY_DIR}/${DIR})
        FOREACH(FILE ${FILES})
            GET_FILENAME_COMPONENT(FILENAME ${FILE} NAME)
            STRING(REGEX REPLACE "\\\.in$" "" FILENAME ${FILENAME})

```



```

        SET (TARGET ${CMAKE_BINARY_DIR}/${DIR}/${FILENAME})
        configure_file(${FILE} ${TARGET} @ONLY IMMEDIATE)
        IF (CONFIG_VERBOSE)
            MESSAGE("${FILE}")
        ENDIF (CONFIG_VERBOSE)
    ENDFOREACH(FILE)
ENDFOREACH(DIR)
ENDFOREACH(FILE_TYPE)

MESSAGE(STATUS "Copying generated bus config files to test directory")
set (OUTDIR ${CMAKE_BINARY_DIR}/test/data/valid-config-files)

FILE(GLOB FILES "${CMAKE_BINARY_DIR}/bus/*.conf" )
FILE(MAKE_DIRECTORY ${OUTDIR})
FOREACH(FILE ${FILES})
    GET_FILENAME_COMPONENT(FILENAME ${FILE} NAME)
    SET (TARGET ${OUTDIR}/${FILENAME})
    configure_file(${FILE} ${TARGET} COPYONLY)
    IF (CONFIG_VERBOSE)
        MESSAGE("FROM: ${FILE}\nTO: ${TARGET}\n")
    ENDIF (CONFIG_VERBOSE)
ENDFOREACH(FILE)

# todo: for installation the TEST_..._BINARY variables must reflect
# the
#     installation dir or has to be defined relative
#

File = CMakeLists.txt.~6~

if (DBUS_BUILD_TESTS)

set (NAMEtest-DIR ../../../../test/name-test)

add_definitions(-DDBUS_COMPILATION)

add_executable(test-pending-call-dispatch ${NAMEtest-DIR}/test-
pending-call-dispatch.c)
target_link_libraries(test-pending-call-dispatch
${DBUS_INTERNAL_LIBRARIES})
ADD_TEST(test-pending-call-dispatch ${EXECUTABLE_OUTPUT_PATH}/test-
pending-call-dispatch)

add_executable(test-pending-call-timeout ${NAMEtest-DIR}/test-pending-
call-timeout.c)
target_link_libraries(test-pending-call-timeout
${DBUS_INTERNAL_LIBRARIES})
ADD_TEST(test-pending-call-timeout ${EXECUTABLE_OUTPUT_PATH}/test-
pending-call-timeout)

```

```

add_executable(test-thread-init ${NAMEtest-DIR}/test-threads-init.c)
target_link_libraries(test-thread-init ${DBUS_INTERNAL_LIBRARIES})
ADD_TEST(test-thread-init ${EXECUTABLE_OUTPUT_PATH}/test-thread-init)

add_executable(test-ids ${NAMEtest-DIR}/test-ids.c)
target_link_libraries(test-ids ${DBUS_INTERNAL_LIBRARIES})
ADD_TEST(test-ids ${EXECUTABLE_OUTPUT_PATH}/test-ids)

add_executable(test-shutdown ${NAMEtest-DIR}/test-shutdown.c)
target_link_libraries(test-shutdown dbus-testutils)
ADD_TEST(test-shutdown ${EXECUTABLE_OUTPUT_PATH}/test-shutdown)

add_executable(test-privserver ${NAMEtest-DIR}/test-privserver.c)
target_link_libraries(test-privserver dbus-testutils)
ADD_TEST(test-privserver ${EXECUTABLE_OUTPUT_PATH}/test-privserver)

add_executable(test-privserver-client ${NAMEtest-DIR}/test-privserver-
client.c)
target_link_libraries(test-privserver-client dbus-testutils)
ADD_TEST(test-privserver-client ${EXECUTABLE_OUTPUT_PATH}/test-
privserver-client)

add_executable(test-autolaunch ${NAMEtest-DIR}/test-autolaunch.c)
target_link_libraries(test-autolaunch dbus-testutils)
ADD_TEST(test-autolaunch ${EXECUTABLE_OUTPUT_PATH}/test-autolaunch)

endif (DBUS_BUILD_TESTS)

```

File = compile

```

#!/bin/sh
# Wrapper for compilers which do not understand '-c -o'.

scriptversion=2012-10-14.11; # UTC

# Copyright (C) 1999-2012 Free Software Foundation, Inc.
# Written by Tom Tromey <tromey@cygnus.com>.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License

```

```

# along with this program.  If not, see
<http://www.gnu.org/licenses/>.

# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program that contains a
# configuration script generated by Autoconf, you may include it under
# the same distribution terms that you use for the rest of that
program.

# This file is maintained in Automake, please report
# bugs to <bug-automake@gnu.org> or send patches to
# <automake-patches@gnu.org>.

nl='
'

# We need space, tab and new line, in precisely that order.  Quoting
is
# there to prevent tools from complaining about whitespace usage.
IFS=" " "$nl"

file_conv=

# func_file_conv build_file lazy
# Convert a $build file to $host form and store it in $file
# Currently only supports Windows hosts.  If the determined conversion
# type is listed in (the comma separated) LAZY, no conversion will
# take place.
func_file_conv ()
{
  file=$1
  case $file in
    / | /[!/]*) # absolute file, and not a UNC file
      if test -z "$file_conv"; then
        # lazily determine how to convert abs files
        case `uname -s` in
          MINGW*)
            file_conv=mingw
            ;;
          CYGWIN*)
            file_conv=cygwin
            ;;
          *)
            file_conv=wine
            ;;
        esac
      fi
      case $file_conv/, $2, in
        *, $file_conv, *)
          ;;
        mingw/*)
          file=`cmd //C echo "$file " | sed -e 's/"\(.*\)" *$/\1/'`

```

```

        ;;
    cygwin/*)
        file=`cygpath -m "$file" || echo "$file"`
        ;;
    wine/*)
        file=`winepath -w "$file" || echo "$file"`
        ;;
    esac
    ;;
esac
}

# func_cl_dashL linkdir
# Make cl look for libraries in LINKDIR
func_cl_dashL ()
{
    func_file_conv "$1"
    if test -z "$lib_path"; then
        lib_path=$file
    else
        lib_path="$lib_path;$file"
    fi
    linker_opts="$linker_opts -LIBPATH:$file"
}

# func_cl_dashl library
# Do a library search-path lookup for cl
func_cl_dashl ()
{
    lib=$1
    found=no
    save_IFS=$IFS
    IFS=';'
    for dir in $lib_path $LIB
    do
        IFS=$save_IFS
        if $shared && test -f "$dir/$lib.dll.lib"; then
            found=yes
            lib=$dir/$lib.dll.lib
            break
        fi
        if test -f "$dir/$lib.lib"; then
            found=yes
            lib=$dir/$lib.lib
            break
        fi
        if test -f "$dir/lib$lib.a"; then
            found=yes
            lib=$dir/lib$lib.a
            break
        fi
    done
}

```

```

IFS=$save_IFS

if test "$found" != yes; then
    lib=$lib.lib
fi
}

# func_cl_wrapper cl arg...
# Adjust compile command to suit cl
func_cl_wrapper ()
{
    # Assume a capable shell
    lib_path=
    shared=:
    linker_opts=
    for arg
    do
        if test -n "$eat"; then
            eat=
        else
            case $1 in
            -o)
                # configure might choose to run compile as 'compile cc -o foo
foo.c'.
                eat=1
                case $2 in
                *.o | *.[oO][bB][jJ])
                    func_file_conv "$2"
                    set x "$@" -Fo"$file"
                    shift
                    ;;
                *)
                    func_file_conv "$2"
                    set x "$@" -Fe"$file"
                    shift
                    ;;
                esac
                ;;
            -I)
                eat=1
                func_file_conv "$2" mingw
                set x "$@" -I"$file"
                shift
                ;;
            -I*)
                func_file_conv "${1#-I}" mingw
                set x "$@" -I"$file"
                shift
                ;;
            -l)
                eat=1
                func_cl_dashl "$2"

```

```

    set x "$@" "$lib"
    shift
    ;;
-l*)
    func_cl_dashl "${1#-l}"
    set x "$@" "$lib"
    shift
    ;;
-L)
    eat=1
    func_cl_dashL "$2"
    ;;
-L*)
    func_cl_dashL "${1#-L}"
    ;;
-static)
    shared=false
    ;;
-Wl,*)
    arg=${1#-Wl,}
    save_ifs="$IFS"; IFS=', '
    for flag in $arg; do
        IFS="$save_ifs"
        linker_opts="$linker_opts $flag"
    done
    IFS="$save_ifs"
    ;;
-Xlinker)
    eat=1
    linker_opts="$linker_opts $2"
    ;;
-*)
    set x "$@" "$1"
    shift
    ;;
*.cc | *.CC | *.cxx | *.CXX | *.[cC]++)
    func_file_conv "$1"
    set x "$@" -Tp"$file"
    shift
    ;;
*.c | *.cpp | *.CPP | *.lib | *.LIB | *.Lib | *.OBJ | *.obj |
*.[oO])
    func_file_conv "$1" mingw
    set x "$@" "$file"
    shift
    ;;
*)
    set x "$@" "$1"
    shift
    ;;
esac
fi

```

```

        shift
    done
    if test -n "$linker_opts"; then
        linker_opts="-link$linker_opts"
    fi
    exec "$@" $linker_opts
    exit 1
}

eat=

case $1 in
    '')
        echo "$0: No command. Try '$0 --help' for more information."
1>&2
        exit 1;
        ;;
    -h | --h*)
        cat <<\EOF
Usage: compile [--help] [--version] PROGRAM [ARGS]

Wrapper for compilers which do not understand '-c -o'.
Remove '-o dest.o' from ARGS, run PROGRAM with the remaining
arguments, and rename the output as expected.

If you are trying to build a whole package this is not the
right script to run: please start by reading the file 'INSTALL'.

Report bugs to <bug-automake@gnu.org>.
EOF
        exit $?
        ;;
    -v | --v*)
        echo "compile $scriptversion"
        exit $?
        ;;
    cl | */[\]cl | cl.exe | */[\]cl.exe )
        func_cl_wrapper "$@" # Doesn't return...
        ;;
)
esac

ofile=
cfile=

for arg
do
    if test -n "$eat"; then
        eat=
    else
        case $1 in
            -o)

```

```

    # configure might choose to run compile as 'compile cc -o foo
foo.c'.
    # So we strip '-o arg' only if arg is an object.
    eat=1
    case $2 in
        *.o | *.obj)
            ofile=$2
            ;;
        *)
            set x "$@" -o "$2"
            shift
            ;;
    esac
    cfile=$1
    set x "$@" "$1"
    shift
    ;;
    *)
    set x "$@" "$1"
    shift
    ;;
    esac
fi
shift
done

if test -z "$ofile" || test -z "$cfile"; then
    # If no '-o' option was seen then we might have been invoked from a
    # pattern rule where we don't need one. That is ok -- this is a
    # normal compilation that the losing compiler can handle. If no
    # '.c' file was seen then we are probably linking. That is also
    # ok.
    exec "$@"
fi

# Name of file we expect compiler to create.
cofile=`echo "$cfile" | sed 's|^\.*[\\/]||; s|^[a-zA-Z]:||;
s|\.c$/\.o/'`

# Create the lock directory.
# Note: use '[/\\:.-]' here to ensure that we don't use the same name
# that we are using for the .o file. Also, base the name on the
expected
# object file name, since that is what matters with a parallel build.
lockdir=`echo "$cofile" | sed -e 's|[/\\:.-]||_g'`.d
while true; do
    if mkdir "$lockdir" >/dev/null 2>&1; then
        break
    fi
    sleep 1

```



```

done
# FIXME: race condition here if user kills between mkdir and trap.
trap "rmdir '$lockdir'; exit 1" 1 2 15

# Run the compile.
"$@"
ret=$?

if test -f "$cofile"; then
    test "$cofile" = "$ofile" || mv "$cofile" "$ofile"
elif test -f "${cofile}bj"; then
    test "${cofile}bj" = "$ofile" || mv "${cofile}bj" "$ofile"
fi

rmdir "$lockdir"
exit $ret

# Local Variables:
# mode: shell-script
# sh-indentation: 2
# eval: (add-hook 'write-file-hooks 'time-stamp)
# time-stamp-start: "scriptversion="
# time-stamp-format: "%:y-%02m-%02d.%02H"
# time-stamp-time-zone: "UTC"
# time-stamp-end: "; # UTC"
# End:

File = config-loader-expat.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-loader-expat.c  expat XML loader
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software

```

```
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA
```

```
*  
*/
```

```
#include <config.h>  
#include "config-parser.h"  
#include <dbus/dbus-internals.h>  
#include <expat.h>
```

```
static XML_Memory_Handling_Suite memsuite;
```

```
typedef struct  
{  
    BusConfigParser *parser;  
    const char *filename;  
    DBusString content;  
    DBusError *error;  
    dbus_bool_t failed;  
} ExpatParseContext;
```

```
static dbus_bool_t  
process_content (ExpatParseContext *context)  
{  
    if (context->failed)  
        return FALSE;  
  
    if (_dbus_string_get_length (&context->content) > 0)  
    {  
        if (!bus_config_parser_content (context->parser,  
                                        &context->content,  
                                        context->error))  
        {  
            context->failed = TRUE;  
            return FALSE;  
        }  
        _dbus_string_set_length (&context->content, 0);  
    }  
  
    return TRUE;  
}
```

```
static void  
expat_StartElementHandler (void *userData,  
                           const XML_Char *name,  
                           const XML_Char **atts)  
{  
    ExpatParseContext *context = userData;  
    int i;  
    char **names;  
    char **values;
```

```

/* Expat seems to suck and can't abort the parse if we
 * throw an error. Expat 2.0 is supposed to fix this.
 */
if (context->failed)
    return;

if (!process_content (context))
    return;

/* "atts" is key, value, key, value, NULL */
for (i = 0; atts[i] != NULL; ++i)
    ; /* nothing */

_dbus_assert (i % 2 == 0);
names = dbus_new0 (char *, i / 2 + 1);
values = dbus_new0 (char *, i / 2 + 1);

if (names == NULL || values == NULL)
{
    dbus_set_error (context->error, DBUS_ERROR_NO_MEMORY, NULL);
    context->failed = TRUE;
    dbus_free (names);
    dbus_free (values);
    return;
}

i = 0;
while (atts[i] != NULL)
{
    _dbus_assert (i % 2 == 0);
    names [i / 2] = (char*) atts[i];
    values[i / 2] = (char*) atts[i+1];

    i += 2;
}

if (!bus_config_parser_start_element (context->parser,
                                     name,
                                     (const char **) names,
                                     (const char **) values,
                                     context->error))
{
    dbus_free (names);
    dbus_free (values);
    context->failed = TRUE;
    return;
}

dbus_free (names);
dbus_free (values);
}

```

```

static void
expat_EndElementHandler (void          *userData,
                        const XML_Char *name)
{
    ExpatParseContext *context = userData;

    if (!process_content (context))
        return;

    if (!bus_config_parser_end_element (context->parser,
                                       name,
                                       context->error))
    {
        context->failed = TRUE;
        return;
    }
}

/* s is not 0 terminated. */
static void
expat_CharacterDataHandler (void          *userData,
                           const XML_Char *s,
                           int            len)
{
    ExpatParseContext *context = userData;
    if (context->failed)
        return;

    if (!_dbus_string_append_len (&context->content,
                                 s, len))
    {
        dbus_set_error (context->error, DBUS_ERROR_NO_MEMORY, NULL);
        context->failed = TRUE;
        return;
    }
}

BusConfigParser*
bus_config_load (const DBusString *file,
                dbus_bool_t       is_toplevel,
                const BusConfigParser *parent,
                DBusError         *error)
{
    XML_Parser expat;
    const char *filename;
    BusConfigParser *parser;
    ExpatParseContext context;
    DBusString dirname;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

```



```

{
    DBusString data;
    const char *data_str;

    if (!_dbus_string_init (&data))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto failed;
    }

    if (!_dbus_file_get_contents (&data, file, error))
    {
        _dbus_string_free (&data);
        goto failed;
    }

    data_str = _dbus_string_get_const_data (&data);

    if (!XML_Parse (expat, data_str, _dbus_string_get_length (&data),
TRUE))
    {
        if (context.error != NULL &&
!dbus_error_is_set (context.error))
        {
            enum XML_Error e;

            e = XML_GetErrorCode (expat);
            if (e == XML_ERROR_NO_MEMORY)
                dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            else
                dbus_set_error (error, DBUS_ERROR_FAILED,
                "Error in file %s, line %d, column %d:
%s\n",
                filename,
                XML_GetCurrentLineNumber (expat),
                XML_GetCurrentColumnNumber (expat),
                XML_ErrorString (e));
        }

        _dbus_string_free (&data);
        goto failed;
    }

    _dbus_string_free (&data);

    if (context.failed)
        goto failed;
}

if (!bus_config_parser_finished (parser, error))
    goto failed;

```

```

    _dbus_string_free (&dirname);
    _dbus_string_free (&context.content);
    XML_ParserFree (expat);

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    return parser;

failed:
    _DBUS_ASSERT_ERROR_IS_SET (error);

    _dbus_string_free (&dirname);
    _dbus_string_free (&context.content);
    if (expat)
        XML_ParserFree (expat);
    if (parser)
        bus_config_parser_unref (parser);
    return NULL;
}

```

File = config-loader-libxml.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-loader-libxml.c  libxml2 XML loader
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#include "config-parser.h"
#include <dbus/dbus-internals.h>

```

```

#include <libxml/xmlreader.h>
#include <libxml/parser.h>
#include <libxml/globals.h>
#include <libxml/xmlmemory.h>
#ifdef HAVE_ERRNO_H
#include <errno.h>
#endif
#include <string.h>

/* About the error handling:
 * - setup a "structured" error handler that catches structural
 *   errors and some oom errors
 * - assume that a libxml function returning an error code means
 *   out-of-memory
 */
#define __DBUS_MAYBE_SET_OOM(e) (dbus_error_is_set(e) ? (void)0 :
__DBUS_SET_OOM(e))

static dbus_bool_t
xml_text_start_element (BusConfigParser *parser,
                        xmlTextReader *reader,
                        DBusError *error)
{
    const char *name;
    int n_attributes;
    const char **attribute_names, **attribute_values;
    dbus_bool_t ret;
    int i, status, is_empty;

    __DBUS_ASSERT_ERROR_IS_CLEAR (error);

    ret = FALSE;
    attribute_names = NULL;
    attribute_values = NULL;

    name = xmlTextReaderConstName (reader);
    n_attributes = xmlTextReaderAttributeCount (reader);
    is_empty = xmlTextReaderIsEmptyElement (reader);

    if (name == NULL || n_attributes < 0 || is_empty == -1)
    {
        __DBUS_MAYBE_SET_OOM (error);
        goto out;
    }

    attribute_names = dbus_new0 (const char *, n_attributes + 1);
    attribute_values = dbus_new0 (const char *, n_attributes + 1);
    if (attribute_names == NULL || attribute_values == NULL)
    {
        __DBUS_SET_OOM (error);
        goto out;
    }
}

```



```

    }
    i = 0;
    while ((status = xmlTextReaderMoveToNextAttribute (reader)) == 1)
    {
        _dbus_assert (i < n_attributes);
        attribute_names[i] = xmlTextReaderConstName (reader);
        attribute_values[i] = xmlTextReaderConstValue (reader);
        if (attribute_names[i] == NULL || attribute_values[i] == NULL)
        {
            _DBUS_MAYBE_SET_OOM (error);
            goto out;
        }
        i++;
    }
    if (status == -1)
    {
        _DBUS_MAYBE_SET_OOM (error);
        goto out;
    }
    _dbus_assert (i == n_attributes);

    ret = bus_config_parser_start_element (parser, name,
                                           attribute_names, attribute_values,
                                           error);

    if (ret && is_empty == 1)
        ret = bus_config_parser_end_element (parser, name, error);

out:
    dbus_free (attribute_names);
    dbus_free (attribute_values);

    return ret;
}

static void xml_shut_up (void *ctx, const char *msg, ...)
{
    return;
}

static void
xml_text_reader_error (void *arg, xmlErrorPtr xml_error)
{
    DBusError *error = arg;

#ifdef 0
    _dbus_verbose ("XML_ERROR level=%d, domain=%d, code=%d, msg=%s\n",
                  xml_error->level, xml_error->domain,
                  xml_error->code, xml_error->message);
#endif

    if (!dbus_error_is_set (error))
    {

```

```

        if (xml_error->code == XML_ERR_NO_MEMORY)
            _DBUS_SET_OOM (error);
        else if (xml_error->level == XML_ERR_ERROR ||
                xml_error->level == XML_ERR_FATAL)
            dbus_set_error (error, DBUS_ERROR_FAILED,
                           "Error loading config file: '%s'",
                           xml_error->message);
    }
}

```

BusConfigParser*

```

bus_config_load (const DBusString      *file,
                 dbus_bool_t          is_toplevel,
                 const BusConfigParser *parent,
                 DBusError             *error)

```

```

{
    xmlTextReader *reader;
    BusConfigParser *parser;
    DBusString dirname, data;
    DBusError tmp_error;
    int ret;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    parser = NULL;
    reader = NULL;

    if (!_dbus_string_init (&dirname))
    {
        _DBUS_SET_OOM (error);
        return NULL;
    }

    if (!_dbus_string_init (&data))
    {
        _DBUS_SET_OOM (error);
        _dbus_string_free (&dirname);
        return NULL;
    }

    if (is_toplevel)
    {
        /* xmlMemSetup only fails if one of the functions is NULL */
        xmlMemSetup (dbus_free,
                    dbus_malloc,
                    dbus_realloc,
                    _dbus_strdup);
        xmlInitParser ();
        xmlSetGenericErrorFunc (NULL, xml_shut_up);
    }
}

```

```

if (!_dbus_string_get_dirname (file, &dirname))
{
    _DBUS_SET_OOM (error);
    goto failed;
}

parser = bus_config_parser_new (&dirname, is_toplevel, parent);
if (parser == NULL)
{
    _DBUS_SET_OOM (error);
    goto failed;
}

if (!_dbus_file_get_contents (&data, file, error))
    goto failed;

reader = xmlReaderForMemory (_dbus_string_get_const_data (&data),
                             _dbus_string_get_length (&data),
                             NULL, NULL, 0);
if (reader == NULL)
{
    _DBUS_SET_OOM (error);
    goto failed;
}

xmlTextReaderSetParserProp (reader, XML_PARSER_SUBST_ENTITIES, 1);

dbus_error_init (&tmp_error);
xmlTextReaderSetStructuredErrorHandler (reader,
xml_text_reader_error, &tmp_error);

while ((ret = xmlTextReaderRead (reader)) == 1)
{
    int type;

    if (dbus_error_is_set (&tmp_error))
        goto reader_out;

    type = xmlTextReaderNodeType (reader);
    if (type == -1)
    {
        _DBUS_MAYBE_SET_OOM (&tmp_error);
        goto reader_out;
    }

    switch ((xmlReaderTypes) type) {
    case XML_READER_TYPE_ELEMENT:
        xml_text_start_element (parser, reader, &tmp_error);
        break;

    case XML_READER_TYPE_TEXT:

```

```

case XML_READER_TYPE_CDATA:
{
    DBusString content;
    const char *value;
    value = xmlTextReaderConstValue (reader);
    if (value != NULL)
        {
            _dbus_string_init_const (&content, value);
            bus_config_parser_content (parser, &content, &tmp_error);
        }
    else
        _DBUS_MAYBE_SET_OOM (&tmp_error);
    break;
}

case XML_READER_TYPE_DOCUMENT_TYPE:
{
    const char *name;
    name = xmlTextReaderConstName (reader);
    if (name != NULL)
        bus_config_parser_check_doctype (parser, name, &tmp_error);
    else
        _DBUS_MAYBE_SET_OOM (&tmp_error);
    break;
}

case XML_READER_TYPE_END_ELEMENT:
{
    const char *name;
    name = xmlTextReaderConstName (reader);
    if (name != NULL)
        bus_config_parser_end_element (parser, name, &tmp_error);
    else
        _DBUS_MAYBE_SET_OOM (&tmp_error);
    break;
}

case XML_READER_TYPE_DOCUMENT:
case XML_READER_TYPE_DOCUMENT_FRAGMENT:
case XML_READER_TYPE_PROCESSING_INSTRUCTION:
case XML_READER_TYPE_COMMENT:
case XML_READER_TYPE_ENTITY:
case XML_READER_TYPE_NOTATION:
case XML_READER_TYPE_WHITESPACE:
case XML_READER_TYPE_SIGNIFICANT_WHITESPACE:
case XML_READER_TYPE_END_ENTITY:
case XML_READER_TYPE_XML_DECLARATION:
/* nothing to do, just read on */
break;

case XML_READER_TYPE_NONE:
case XML_READER_TYPE_ATTRIBUTE:

```

```

        case XML_READER_TYPE_ENTITY_REFERENCE:
            _dbus_assert_not_reached ("unexpected nodes in XML");
        }

        if (dbus_error_is_set (&tmp_error))
            goto reader_out;
    }

    if (ret == -1)
        _DBUS_MAYBE_SET_OOM (&tmp_error);

reader_out:
    xmlFreeTextReader (reader);
    reader = NULL;
    if (dbus_error_is_set (&tmp_error))
    {
        dbus_move_error (&tmp_error, error);
        goto failed;
    }

    if (!bus_config_parser_finished (parser, error))
        goto failed;
    _dbus_string_free (&dirname);
    _dbus_string_free (&data);
    if (is_toplevel)
        xmlCleanupParser();
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    return parser;

failed:
    _DBUS_ASSERT_ERROR_IS_SET (error);
    _dbus_string_free (&dirname);
    _dbus_string_free (&data);
    if (is_toplevel)
        xmlCleanupParser();
    if (parser)
        bus_config_parser_unref (parser);
    _dbus_assert (reader == NULL); /* must go to reader_out first */
    return NULL;
}

```

File = config-parser-common.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser-common.c Common defines and routines for config file
parsing
*
* Copyright (C) 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1

```

```
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include <dbus/dbus-internals.h>
#include <string.h>
```

```
#include "config-parser-common.h"
#include "utils.h"
```

```
ElementType
```

```
bus_config_parser_element_name_to_type (const char *name)
```

```
{
    if (strcmp (name, "none") == 0)
        {
            return ELEMENT_NONE;
        }
    else if (strcmp (name, "busconfig") == 0)
        {
            return ELEMENT_BUSCONFIG;
        }
    else if (strcmp (name, "user") == 0)
        {
            return ELEMENT_USER;
        }
    else if (strcmp (name, "auth") == 0)
        {
            return ELEMENT_AUTH;
        }
    else if (strcmp (name, "type") == 0)
        {
            return ELEMENT_CONFIGTYPE;
        }
    else if (strcmp (name, "fork") == 0)
        {
```

```
        return ELEMENT_FORK;
    }
else if (strcmp (name, "pidfile") == 0)
    {
        return ELEMENT_PIDFILE;
    }
else if (strcmp (name, "listen") == 0)
    {
        return ELEMENT_LISTEN;
    }
else if (strcmp (name, "auth") == 0)
    {
        return ELEMENT_AUTH;
    }
else if (strcmp (name, "allow") == 0)
    {
        return ELEMENT_ALLOW;
    }
else if (strcmp (name, "deny") == 0)
    {
        return ELEMENT_DENY;
    }
else if (strcmp (name, "servicehelper") == 0)
    {
        return ELEMENT_SERVICEHELPER;
    }
else if (strcmp (name, "includedir") == 0)
    {
        return ELEMENT_INCLUDEDIR;
    }
else if (strcmp (name, "standard_session_servicedirs") == 0)
    {
        return ELEMENT_STANDARD_SESSION_SERVICEDIRS;
    }
else if (strcmp (name, "standard_system_servicedirs") == 0)
    {
        return ELEMENT_STANDARD_SYSTEM_SERVICEDIRS;
    }
else if (strcmp (name, "servicedir") == 0)
    {
        return ELEMENT_SERVICEDIR;
    }
else if (strcmp (name, "include") == 0)
    {
        return ELEMENT_INCLUDE;
    }
else if (strcmp (name, "policy") == 0)
    {
        return ELEMENT_POLICY;
    }
else if (strcmp (name, "limit") == 0)
    {
```

```

        return ELEMENT_LIMIT;
    }
    else if (strcmp (name, "selinux") == 0)
    {
        return ELEMENT_SELINUX;
    }
    else if (strcmp (name, "associate") == 0)
    {
        return ELEMENT_ASSOCIATE;
    }
    else if (strcmp (name, "syslog") == 0)
    {
        return ELEMENT_SYSLOG;
    }
    else if (strcmp (name, "keep_umask") == 0)
    {
        return ELEMENT_KEEP_UMASK;
    }
    else if (strcmp (name, "allow_anonymous") == 0)
    {
        return ELEMENT_ALLOW_ANONYMOUS;
    }
    return ELEMENT_NONE;
}

const char*
bus_config_parser_element_type_to_name (ElementType type)
{
    switch (type)
    {
        case ELEMENT_NONE:
            return NULL;
        case ELEMENT_BUSCONFIG:
            return "busconfig";
        case ELEMENT_INCLUDE:
            return "include";
        case ELEMENT_USER:
            return "user";
        case ELEMENT_LISTEN:
            return "listen";
        case ELEMENT_AUTH:
            return "auth";
        case ELEMENT_POLICY:
            return "policy";
        case ELEMENT_LIMIT:
            return "limit";
        case ELEMENT_ALLOW:
            return "allow";
        case ELEMENT_DENY:
            return "deny";
        case ELEMENT_FORK:
            return "fork";
    }
}

```



```

case ELEMENT_PIDFILE:
    return "pidfile";
case ELEMENT_STANDARD_SESSION_SERVICEDIRS:
    return "standard_session_servicedirs";
case ELEMENT_STANDARD_SYSTEM_SERVICEDIRS:
    return "standard_system_servicedirs";
case ELEMENT_SERVICEDIR:
    return "servicedir";
case ELEMENT_SERVICEHELPER:
    return "servicehelper";
case ELEMENT_INCLUDEDIR:
    return "includedir";
case ELEMENT_CONFIGTYPE:
    return "type";
case ELEMENT_SELINUX:
    return "selinux";
case ELEMENT_ASSOCIATE:
    return "associate";
case ELEMENT_SYSLOG:
    return "syslog";
case ELEMENT_KEEP_UMASK:
    return "keep_umask";
case ELEMENT_ALLOW_ANONYMOUS:
    return "allow_anonymous";
}

_dbus_assert_not_reached ("bad element type");

return NULL;
}

```

File = config-parser-common.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser-common.h Common defines and routines for config file
parsing
*
* Copyright (C) 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,

```

```
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#ifndef BUS_CONFIG_PARSER_COMMON_H
#define BUS_CONFIG_PARSER_COMMON_H
```

```
typedef enum
```

```
{
    ELEMENT_NONE,
    ELEMENT_BUSCONFIG,
    ELEMENT_INCLUDE,
    ELEMENT_USER,
    ELEMENT_LISTEN,
    ELEMENT_AUTH,
    ELEMENT_POLICY,
    ELEMENT_LIMIT,
    ELEMENT_ALLOW,
    ELEMENT_DENY,
    ELEMENT_FORK,
    ELEMENT_PIDFILE,
    ELEMENT_SERVICEDIR,
    ELEMENT_SERVICEHELPER,
    ELEMENT_INCLUDEDIR,
    /* this is really <type>, but winioctl.h defines ELEMENT_TYPE */
    ELEMENT_CONFIGTYPE,
    ELEMENT_SELINUX,
    ELEMENT_ASSOCIATE,
    ELEMENT_STANDARD_SESSION_SERVICEDIRS,
    ELEMENT_STANDARD_SYSTEM_SERVICEDIRS,
    ELEMENT_KEEP_UMASK,
    ELEMENT_SYSLOG,
    ELEMENT_ALLOW_ANONYMOUS
} ElementType;
```

```
ElementType bus_config_parser_element_name_to_type (const char
*element_name);
const char* bus_config_parser_element_type_to_name (ElementType type);

#endif /* BUS_CONFIG_PARSER_COMMON_H */
```

```
File = config-parser-trivial.c
```

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser-trivial.c XML-library-agnostic configuration file
parser
*
*          Does not do includes or anything remotely
complicated.
*
* Copyright (C) 2003, 2004, 2007 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#include "config-parser-common.h"
#include "config-parser-trivial.h"
#include "utils.h"
#include <dbus/dbus-list.h>
#include <dbus/dbus-internals.h>
#include <string.h>

/**
 * TRIVIAL parser for bus configuration file.
 */
struct BusConfigParser
{
    ElementType type;
    DBusString user;          /**< User the dbus-daemon runs as
*/
    DBusString bus_type;     /**< Message bus type */
    DBusString service_helper; /**< Location of the setuid helper
*/
    DBusList *service_dirs;  /**< Directories to look for
services in */
};

```

```

static dbus_bool_t
service_dirs_find_dir (DBusList **service_dirs,
                      const char *dir)
{
    DBusList *link;

    _dbus_assert (dir != NULL);

    for (link = *service_dirs; link; link =
         _dbus_list_get_next_link(service_dirs, link))
    {
        const char *link_dir;

        link_dir = (const char *)link->data;
        if (strcmp (dir, link_dir) == 0)
            return TRUE;
    }

    return FALSE;
}

static void
service_dirs_append_link_unique_or_free (DBusList **service_dirs,
                                         DBusList *dir_link)
{
    if (!service_dirs_find_dir (service_dirs, dir_link->data))
    {
        _dbus_list_append_link (service_dirs, dir_link);
    }
    else
    {
        dbus_free (dir_link->data);
        _dbus_list_free_link (dir_link);
    }
}

BusConfigParser*
bus_config_parser_new (const DBusString          *basedir,
                      dbus_bool_t              is_toplevel,
                      const BusConfigParser     *parent)
{
    BusConfigParser *parser;

    parser = dbus_new0 (BusConfigParser, 1);
    if (parser == NULL)
        goto failed;

    parser->type = ELEMENT_NONE;

    /* init the lists */
    parser->service_dirs = NULL;

```

```

/* init the strings */
if (!_dbus_string_init (&parser->user))
    goto failed_parser;
if (!_dbus_string_init (&parser->bus_type))
    goto failed_type;
if (!_dbus_string_init (&parser->service_helper))
    goto failed_helper;

/* woot! */
return parser;

/* argh. we have to do this carefully because of OOM */
failed_helper:
    _dbus_string_free (&parser->bus_type);
failed_type:
    _dbus_string_free (&parser->user);
failed_parser:
    dbus_free (parser);
failed:
    return NULL;
}

void
bus_config_parser_unref (BusConfigParser *parser)
{
    _dbus_string_free (&parser->user);
    _dbus_string_free (&parser->service_helper);
    _dbus_string_free (&parser->bus_type);

    _dbus_list_foreach (&parser->service_dirs,
                        (DBusForeachFunction) dbus_free,
                        NULL);

    _dbus_list_clear (&parser->service_dirs);

    dbus_free (parser);
}

dbus_bool_t
bus_config_parser_check_doctype (BusConfigParser *parser,
                                const char *doctype,
                                DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (strcmp (doctype, "busconfig") != 0)
    {
        dbus_set_error (error,
                        DBUS_ERROR_FAILED,
                        "Configuration file has the wrong document type
%s",

```

```

        doctype);
    return FALSE;
}
else
    return TRUE;
}

dbus_bool_t
bus_config_parser_start_element (BusConfigParser *parser,
                                const char *element_name,
                                const char **attribute_names,
                                const char **attribute_values,
                                DBusError *error)
{
    /* we don't do processing of attribute names, we don't need to */
    parser->type = bus_config_parser_element_name_to_type
(element_name);

    switch (parser->type)
    {
        case ELEMENT_SERVICEHELPER:
        case ELEMENT_USER:
        case ELEMENT_CONFIGTYPE:
            /* content about to be handled */
            break;

        case ELEMENT_STANDARD_SYSTEM_SERVICEDIRS:
            {
                DBusList *link;
                DBusList *dirs;
                dirs = NULL;

                if (!_dbus_get_standard_system_servicedirs (&dirs))
                {
                    BUS_SET_OOM (error);
                    return FALSE;
                }

                while ((link = _dbus_list_pop_first_link (&dirs))
                    service_dirs_append_link_unique_or_free (&parser-
>service_dirs, link);
                    break;
                }

            default:
                {
                    /* we really don't care about the others... */
                    _dbus_verbose (" START We don't care about '%s' type '%i'\n",
element_name, parser->type);
                    break;
                }
            }
    }
}

```

```

    return TRUE;
}

dbus_bool_t
bus_config_parser_end_element (BusConfigParser *parser,
                               const char *element_name,
                               DBusError *error)
{
    /* we don't care */
    return TRUE;
}

dbus_bool_t
bus_config_parser_content (BusConfigParser *parser,
                           const DBusString *content,
                           DBusError *error)
{
    DBusString content_sane;
    dbus_bool_t retval;

    retval = FALSE;

    if (!_dbus_string_init (&content_sane))
    {
        BUS_SET_OOM (error);
        goto out;
    }
    if (!_dbus_string_copy (content, 0, &content_sane, 0))
    {
        BUS_SET_OOM (error);
        goto out_content;
    }

    /* rip out white space */
    _dbus_string_chop_white (&content_sane);
    if (_dbus_string_get_length (&content_sane) == 0)
    {
        /* optimise, there is no content */
        retval = TRUE;
        goto out_content;
    }

    switch (parser->type)
    {
    case ELEMENT_SERVICEDIR:
        {
            char *cpath;

            /* copy the sane data into a char array */
            if (!_dbus_string_copy_data (&content_sane, &cpath))
            {
                BUS_SET_OOM (error);
            }
        }
    }
}

```

```

        goto out_content;
    }

    /* append the dynamic char string to service dirs */
    if (!_dbus_list_append (&parser->service_dirs, cpath))
    {
        dbus_free (cpath);
        BUS_SET_OOM (error);
        goto out_content;
    }
}
break;

case ELEMENT_SERVICEHELPER:
{
    if (!_dbus_string_copy (&content_sane, 0, &parser->service_helper, 0))
    {
        BUS_SET_OOM (error);
        goto out_content;
    }
}
break;

case ELEMENT_USER:
{
    if (!_dbus_string_copy (&content_sane, 0, &parser->user, 0))
    {
        BUS_SET_OOM (error);
        goto out_content;
    }
}
break;

case ELEMENT_CONFIGTYPE:
{
    if (!_dbus_string_copy (&content_sane, 0, &parser->bus_type,
0))
    {
        BUS_SET_OOM (error);
        goto out_content;
    }
}
break;
default:
{
    /* we don't care about the others... really */
    _dbus_verbose (" CONTENTS We don't care about '%s' type
%i'\n", _dbus_string_get_const_data (&content_sane), parser->type);
    break;
}
}

```



```

    /* woot! */
    retval = TRUE;

out_content:
    _dbus_string_free (&content_sane);
out:
    return retval;
}

dbus_bool_t
bus_config_parser_finished (BusConfigParser *parser,
                            DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    _dbus_verbose ("finished scanning!\n");
    return TRUE;
}

const char*
bus_config_parser_get_user (BusConfigParser *parser)
{
    return _dbus_string_get_const_data (&parser->user);
}

const char*
bus_config_parser_get_type (BusConfigParser *parser)
{
    return _dbus_string_get_const_data (&parser->bus_type);
}

DBusList**
bus_config_parser_get_service_dirs (BusConfigParser *parser)
{
    return &parser->service_dirs;
}

#ifdef DBUS_BUILD_TESTS
#include <stdio.h>
#include "test.h"

typedef enum
{
    VALID,
    INVALID,
    UNKNOWN
} Validity;

static dbus_bool_t
check_return_values (const DBusString *full_path)
{
    BusConfigParser *parser;

```

```

DBusError error;
dbus_bool_t retval;
const char *user;
const char *type;
DBusList **dirs;

dbus_error_init (&error);
retval = FALSE;

printf ("Testing values from: %s\n", _dbus_string_get_const_data
(full_path));

parser = bus_config_load (full_path, TRUE, NULL, &error);
if (parser == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (&error);
    if (dbus_error_has_name (&error, DBUS_ERROR_NO_MEMORY))
        _dbus_verbose ("Failed to load valid file due to OOM\n");
    goto finish;
}
_DBUS_ASSERT_ERROR_IS_CLEAR (&error);

/* check user return value is okay */
user = bus_config_parser_get_user (parser);
if (user == NULL)
{
    _dbus_warn ("User was NULL!\n");
    goto finish;
}
#if 0
/* the username can be configured in configure.in so this test
doesn't work */
if (strcmp (user, "dbus") != 0)
{
    _dbus_warn ("User was invalid; '%s'!\n", user);
    goto finish;
}
printf ("    <user>dbus</user> OKAY!\n");
#endif

/* check type return value is okay */
type = bus_config_parser_get_type (parser);
if (type == NULL)
{
    _dbus_warn ("Type was NULL!\n");
    goto finish;
}
if (strcmp (type, "system") != 0)
{
    _dbus_warn ("Type was invalid; '%s'!\n", user);
    goto finish;
}

```

```

printf ("    <type>system</type> OKAY!\n");

/* check dirs return value is okay */
dirs = bus_config_parser_get_service_dirs (parser);
if (dirs == NULL)
{
    _dbus_warn ("Service dirs are NULL!\n");
    goto finish;
}
printf ("    <standard_system_service_dirs/> OKAY!\n");
/* NOTE: We tested the specific return values in the config-parser
tests */

/* woohoo! */
retval = TRUE;
finish:
if (parser != NULL)
    bus_config_parser_unref (parser);
dbus_error_free (&error);
return retval;
}

static dbus_bool_t
do_load (const DBusString *full_path,
         Validity          validity,
         dbus_bool_t       oom_possible)
{
    BusConfigParser *parser;
    DBusError error;

    dbus_error_init (&error);

    parser = bus_config_load (full_path, TRUE, NULL, &error);
    if (parser == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (&error);

        if (oom_possible &&
            dbus_error_has_name (&error, DBUS_ERROR_NO_MEMORY))
        {
            _dbus_verbose ("Failed to load valid file due to OOM\n");
            dbus_error_free (&error);
            return TRUE;
        }
        else if (validity == VALID)
        {
            _dbus_warn ("Failed to load valid file but still had memory:
%s\n",
                        error.message);

            dbus_error_free (&error);
            return FALSE;
        }
    }
}

```

```

        }
    else
    {
        dbus_error_free (&error);
        return TRUE;
    }
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (&error);

    bus_config_parser_unref (parser);

    if (validity == INVALID)
    {
        _dbus_warn ("Accepted invalid file\n");
        return FALSE;
    }

    return TRUE;
}
}

typedef struct
{
    const DBusString *full_path;
    Validity          validity;
} LoaderOomData;

static dbus_bool_t
check_loader_oom_func (void *data)
{
    LoaderOomData *d = data;

    return do_load (d->full_path, d->validity, TRUE);
}

static dbus_bool_t
process_test_valid_subdir (const DBusString *test_base_dir,
                           const char      *subdir,
                           Validity        validity)
{
    DBusString test_directory;
    DBusString filename;
    DBusDirIter *dir;
    dbus_bool_t retval;
    DBusError error;

    retval = FALSE;
    dir = NULL;

    if (!_dbus_string_init (&test_directory))

```

```

    _dbus_assert_not_reached ("didn't allocate test_directory\n");

    _dbus_string_init_const (&filename, subdir);

    if (!_dbus_string_copy (test_base_dir, 0,
                           &test_directory, 0))
        _dbus_assert_not_reached ("couldn't copy test_base_dir to
test_directory");

    if (!_dbus_concat_dir_and_file (&test_directory, &filename))
        _dbus_assert_not_reached ("couldn't allocate full path");

    _dbus_string_free (&filename);
    if (!_dbus_string_init (&filename))
        _dbus_assert_not_reached ("didn't allocate filename string\n");

    dbus_error_init (&error);
    dir = _dbus_directory_open (&test_directory, &error);
    if (dir == NULL)
    {
        _dbus_warn ("Could not open %s: %s\n",
                   _dbus_string_get_const_data (&test_directory),
                   error.message);
        dbus_error_free (&error);
        goto failed;
    }

    if (validity == VALID)
        printf ("Testing valid files:\n");
    else if (validity == INVALID)
        printf ("Testing invalid files:\n");
    else
        printf ("Testing unknown files:\n");

next:
    while (_dbus_directory_get_next_file (dir, &filename, &error))
    {
        DBusString full_path;
        LoaderOomData d;

        if (!_dbus_string_init (&full_path))
            _dbus_assert_not_reached ("couldn't init string");

        if (!_dbus_string_copy (&test_directory, 0, &full_path, 0))
            _dbus_assert_not_reached ("couldn't copy dir to full_path");

        if (!_dbus_concat_dir_and_file (&full_path, &filename))
            _dbus_assert_not_reached ("couldn't concat file to dir");

        if (!_dbus_string_ends_with_c_str (&full_path, ".conf"))
        {
            _dbus_verbose ("Skipping non-.conf file %s\n",

```

```

        _dbus_string_get_const_data (&filename));
    _dbus_string_free (&full_path);
    goto next;
}

printf ("    %s\n", _dbus_string_get_const_data (&filename));

_dbus_verbose (" expecting %s\n",
               validity == VALID ? "valid" :
               (validity == INVALID ? "invalid" :
               (validity == UNKNOWN ? "unknown" : "???")));

d.full_path = &full_path;
d.validity = validity;

/* FIXME hackaround for an expat problem, see
 * https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=124747
 * http://freedesktop.org/pipermail/dbus/2004-May/001153.html
 */
/* if (!_dbus_test_oom_handling ("config-loader",
check_loader_oom_func, &d)) */
if (!check_loader_oom_func (&d))
    _dbus_assert_not_reached ("test failed");

    _dbus_string_free (&full_path);
}

if (dbus_error_is_set (&error))
{
    _dbus_warn ("Could not get next file in %s: %s\n",
               _dbus_string_get_const_data (&test_directory),
               error.message);
    dbus_error_free (&error);
    goto failed;
}

retval = TRUE;

failed:

if (dir)
    _dbus_directory_close (dir);
_dbus_string_free (&test_directory);
_dbus_string_free (&filename);

return retval;
}

/* convenience function, do not reuse outside of TEST */
static dbus_bool_t
make_full_path (const DBusString *test_data_dir,
               const char      *subdir,

```

```

                                const char      *file,
                                DBusString      *full_path)
{
    DBusString filename;
    dbus_bool_t retval;

    retval = FALSE;

    if (!_dbus_string_init (full_path))
        {
            _dbus_warn ("couldn't allocate full path");
            goto finish;
        }

    if (!_dbus_string_copy (test_data_dir, 0, full_path, 0))
        {
            _dbus_warn ("couldn't allocate full path");
            goto finish;
        }

    _dbus_string_init_const (&filename, subdir);
    if (!_dbus_concat_dir_and_file (full_path, &filename))
        {
            _dbus_warn ("couldn't allocate full path");
            goto finish;
        }
    _dbus_string_free (&filename);

    _dbus_string_init_const (&filename, file);
    if (!_dbus_concat_dir_and_file (full_path, &filename))
        {
            _dbus_warn ("couldn't allocate full path");
            goto finish;
        }

    /* woot! */
    retval = TRUE;

finish:
    _dbus_string_free (&filename);
    return retval;
}

static dbus_bool_t
check_file_valid (DBusString *full_path,
                 Validity     validity)
{
    dbus_bool_t retval;

    if (validity == VALID)
        printf ("Testing valid file:\n");
    else if (validity == INVALID)

```

```

    printf ("Testing invalid file:\n");
else
    printf ("Testing unknown file:\n");

/* print the filename, just so we match the other output */
printf ("    %s\n", _dbus_string_get_const_data (full_path));

/* only test one file */
retval = do_load (full_path, validity, TRUE);

return retval;
}

dbus_bool_t
bus_config_parser_trivial_test (const DBusString *test_data_dir)
{
    DBusString full_path;
    dbus_bool_t retval;

    retval = FALSE;

    if (test_data_dir == NULL ||
        _dbus_string_get_length (test_data_dir) == 0)
    {
        printf ("No test data\n");
        return TRUE;
    }

    /* We already test default_session_servicedirs and
    default_system_servicedirs
    * in bus_config_parser_test() */
    if (!process_test_valid_subdir (test_data_dir, "valid-config-files",
    VALID))
        goto finish;

    /* we don't process all the invalid files, as the trivial parser
    can't hope
    * to validate them all for all different syntaxes. We just check
    one broken
    * file to see if junk is received */
    if (!make_full_path (test_data_dir, "invalid-config-files", "not-
    well-formed.conf", &full_path))
        goto finish;
    if (!check_file_valid (&full_path, INVALID))
        goto finish;
    _dbus_string_free (&full_path);

    /* just test if the check_file_valid works okay and we got sane
    values */
    if (!make_full_path (test_data_dir, "valid-config-files",
    "system.conf", &full_path))
        goto finish;

```



```

    if (!check_file_valid (&full_path, VALID))
        goto finish;
    /* check to see if we got the correct values from the parser */
    if (!check_return_values (&full_path))
        goto finish;

    /* woot! */
    retval = TRUE;

finish:
    _dbus_string_free (&full_path);

    /* we don't process equiv-config-files as we don't handle <include>
*/
    return retval;
}

#endif /* DBUS_BUILD_TESTS */

```

File = config-parser-trivial.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser-trivial.h XML-library-agnostic configuration file
parser
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef BUS_CONFIG_PARSER_TRIVIAL_H

```

```

#define BUS_CONFIG_PARSER_TRIVIAL_H

#include <dbus/dbus.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-list.h>
#include <dbus/dbus-hash.h>

/* Whatever XML library we're using just pushes data into this API */
typedef struct BusConfigParser BusConfigParser;

BusConfigParser* bus_config_parser_new (const DBusString
*basedir,
                                         dbus_bool_t
is_toplevel,
                                         const BusConfigParser
*parent);

BusConfigParser* bus_config_parser_ref      (BusConfigParser
*parser);
void bus_config_parser_unref      (BusConfigParser
*parser);
dbus_bool_t bus_config_parser_check_doctype (BusConfigParser
*parser,
                                             const char
*doctype,
                                             DBusError
*error);
dbus_bool_t bus_config_parser_start_element (BusConfigParser
*parser,
                                             const char
*element_name,
                                             const char
**attribute_names,
                                             const char
**attribute_values,
                                             DBusError
*error);
dbus_bool_t bus_config_parser_end_element  (BusConfigParser
*parser,
                                             const char
*element_name,
                                             DBusError
*error);
dbus_bool_t bus_config_parser_content     (BusConfigParser
*parser,
                                             const DBusString
*content,
                                             DBusError
*error);
dbus_bool_t bus_config_parser_finished    (BusConfigParser
*parser,

```

```

                                                                    DBusError
*error);

/* Functions for extracting the parse results */
const char* bus_config_parser_get_user      (BusConfigParser
*parser);
const char* bus_config_parser_get_type     (BusConfigParser
*parser);
DBusList** bus_config_parser_get_service_dirs (BusConfigParser
*parser);

/* Loader functions (backended off one of the XML parsers). Returns a
 * finished ConfigParser.
 */
BusConfigParser* bus_config_load (const DBusString      *file,
                                dbus_bool_t           is_toplevel,
                                const BusConfigParser *parent,
                                DBusError             *error);

#endif /* BUS_CONFIG_PARSER_TRIVIAL_H */

```

File = config-parser.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser.c XML-library-agnostic configuration file parser
 *
 * Copyright (C) 2003, 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>

```

```

#include "config-parser-common.h"
#include "config-parser.h"
#include "test.h"
#include "utils.h"
#include "policy.h"
#include "selinux.h"
#include <dbus/dbus-list.h>
#include <dbus/dbus-internals.h>
#include <dbus/dbus-sysdeps.h>
#include <string.h>

typedef enum
{
    /* we ignore policies for unknown groups/users */
    POLICY_IGNORED,

    /* non-ignored */
    POLICY_DEFAULT,
    POLICY_MANDATORY,
    POLICY_USER,
    POLICY_GROUP,
    POLICY_CONSOLE
} PolicyType;

typedef struct
{
    ElementType type;

    unsigned int had_content : 1;

    union
    {
        struct
        {
            unsigned int ignore_missing : 1;
            unsigned int if_selinux_enabled : 1;
            unsigned int selinux_root_relative : 1;
        } include;

        struct
        {
            PolicyType type;
            unsigned long gid_uid_or_at_console;
        } policy;

        struct
        {
            char *name;
            long value;
        } limit;
    } d;
}

```

```

} Element;

/**
 * Parser for bus configuration file.
 */
struct BusConfigParser
{
    int refcount;          /**< Reference count */

    DBusString basedir;   /**< Directory we resolve paths relative to */

    DBusList *stack;      /**< stack of Element */

    char *user;           /**< user to run as */

    char *servicehelper; /**< location of the setuid helper */

    char *bus_type;       /**< Message bus type */

    DBusList *listen_on; /**< List of addresses to listen to */

    DBusList *mechanisms; /**< Auth mechanisms */

    DBusList *service_dirs; /**< Directories to look for session
services in */

    DBusList *conf_dirs; /**< Directories to look for policy
configuration in */

    BusPolicy *policy;    /**< Security policy */

    BusLimits limits;     /**< Limits */

    char *pidfile;        /**< PID file */

    DBusList *included_files; /**< Included files stack */

    DBusHashTable *service_context_table; /**< Map service names to
SELinux contexts */

    unsigned int fork : 1; /**< TRUE to fork into daemon mode */

    unsigned int syslog : 1; /**< TRUE to enable syslog */
    unsigned int keep_umask : 1; /**< TRUE to keep original umask when
forking */

    unsigned int is_toplevel : 1; /**< FALSE if we are a sub-config-file
inside another one */

    unsigned int allow_anonymous : 1; /**< TRUE to allow anonymous
connections */

```

```

};

static Element*
push_element (BusConfigParser *parser,
              ElementType      type)
{
    Element *e;

    _dbus_assert (type != ELEMENT_NONE);

    e = dbus_new0 (Element, 1);
    if (e == NULL)
        return NULL;

    if (!_dbus_list_append (&parser->stack, e))
    {
        dbus_free (e);
        return NULL;
    }

    e->type = type;

    return e;
}

static void
element_free (Element *e)
{
    if (e->type == ELEMENT_LIMIT)
        dbus_free (e->d.limit.name);

    dbus_free (e);
}

static void
pop_element (BusConfigParser *parser)
{
    Element *e;

    e = _dbus_list_pop_last (&parser->stack);

    element_free (e);
}

static Element*
peek_element (BusConfigParser *parser)
{
    Element *e;

    e = _dbus_list_get_last (&parser->stack);

    return e;
}

```

```

}

static ElementType
top_element_type (BusConfigParser *parser)
{
    Element *e;

    e = _dbus_list_get_last (&parser->stack);

    if (e)
        return e->type;
    else
        return ELEMENT_NONE;
}

static dbus_bool_t
merge_service_context_hash (DBusHashTable *dest,
                            DBusHashTable *from)
{
    DBusHashIter iter;
    char *service_copy;
    char *context_copy;

    service_copy = NULL;
    context_copy = NULL;

    _dbus_hash_iter_init (from, &iter);
    while (_dbus_hash_iter_next (&iter))
    {
        const char *service = _dbus_hash_iter_get_string_key (&iter);
        const char *context = _dbus_hash_iter_get_value (&iter);

        service_copy = _dbus_strdup (service);
        if (service_copy == NULL)
            goto fail;
        context_copy = _dbus_strdup (context);
        if (context_copy == NULL)
            goto fail;

        if (!_dbus_hash_table_insert_string (dest, service_copy,
context_copy))
            goto fail;

        service_copy = NULL;
        context_copy = NULL;
    }

    return TRUE;

fail:
    if (service_copy)
        dbus_free (service_copy);
}

```

```

    if (context_copy)
        dbus_free (context_copy);

    return FALSE;
}

static dbus_bool_t
service_dirs_find_dir (DBusList **service_dirs,
                      const char *dir)
{
    DBusList *link;

    _dbus_assert (dir != NULL);

    for (link = *service_dirs; link; link =
         _dbus_list_get_next_link(service_dirs, link))
    {
        const char *link_dir;

        link_dir = (const char *)link->data;
        if (strcmp (dir, link_dir) == 0)
            return TRUE;
    }

    return FALSE;
}

static dbus_bool_t
service_dirs_append_unique_or_free (DBusList **service_dirs,
                                   char *dir)
{
    if (!service_dirs_find_dir (service_dirs, dir))
        return _dbus_list_append (service_dirs, dir);

    dbus_free (dir);
    return TRUE;
}

static void
service_dirs_append_link_unique_or_free (DBusList **service_dirs,
                                       DBusList *dir_link)
{
    if (!service_dirs_find_dir (service_dirs, dir_link->data))
    {
        _dbus_list_append_link (service_dirs, dir_link);
    }
    else
    {
        dbus_free (dir_link->data);
        _dbus_list_free_link (dir_link);
    }
}

```



```

}

static dbus_bool_t
merge_included (BusConfigParser *parser,
                BusConfigParser *included,
                DBusError      *error)
{
    DBusList *link;

    if (!bus_policy_merge (parser->policy,
                          included->policy))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!merge_service_context_hash (parser->service_context_table,
                                     included->service_context_table))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (included->user != NULL)
    {
        dbus_free (parser->user);
        parser->user = included->user;
        included->user = NULL;
    }

    if (included->bus_type != NULL)
    {
        dbus_free (parser->bus_type);
        parser->bus_type = included->bus_type;
        included->bus_type = NULL;
    }

    if (included->fork)
        parser->fork = TRUE;

    if (included->keep_umask)
        parser->keep_umask = TRUE;

    if (included->pidfile != NULL)
    {
        dbus_free (parser->pidfile);
        parser->pidfile = included->pidfile;
        included->pidfile = NULL;
    }

    while ((link = _dbus_list_pop_first_link (&included->listen_on)))
        _dbus_list_append_link (&parser->listen_on, link);
}

```

```

while ((link = _dbus_list_pop_first_link (&included->mechanisms))
       _dbus_list_append_link (&parser->mechanisms, link);

while ((link = _dbus_list_pop_first_link (&included->service_dirs))
       service_dirs_append_link_unique_or_free (&parser->service_dirs,
link);

while ((link = _dbus_list_pop_first_link (&included->conf_dirs))
       _dbus_list_append_link (&parser->conf_dirs, link);

return TRUE;
}

static dbus_bool_t
seen_include (BusConfigParser *parser,
             const DBusString *file)
{
    DBusList *iter;

    iter = parser->included_files;
    while (iter != NULL)
    {
        if (! strcmp (_dbus_string_get_const_data (file), iter->data))
            return TRUE;

        iter = _dbus_list_get_next_link (&parser->included_files, iter);
    }

    return FALSE;
}

BusConfigParser*
bus_config_parser_new (const DBusString *basedir,
                      dbus_bool_t      is_toplevel,
                      const BusConfigParser *parent)
{
    BusConfigParser *parser;

    parser = dbus_new0 (BusConfigParser, 1);
    if (parser == NULL)
        return NULL;

    parser->is_toplevel = !!is_toplevel;

    if (!_dbus_string_init (&parser->basedir))
    {
        dbus_free (parser);
        return NULL;
    }

    if ((parser->policy = bus_policy_new ()) == NULL) ||

```

```

    !_dbus_string_copy (basedir, 0, &parser->basedir, 0) ||
    ((parser->service_context_table = _dbus_hash_table_new
(DBUS_HASH_STRING,
                                dbus_free,
                                dbus_free)) == NULL))
{
    if (parser->policy)
        bus_policy_unref (parser->policy);

    _dbus_string_free (&parser->basedir);

    dbus_free (parser);
    return NULL;
}

if (parent != NULL)
{
    /* Initialize the parser's limits from the parent. */
    parser->limits = parent->limits;

    /* Use the parent's list of included_files to avoid
    circular inclusions. */
    parser->included_files = parent->included_files;
}
else
{
    /* Make up some numbers! woot! */
    parser->limits.max_incoming_bytes = _DBUS_ONE_MEGABYTE * 127;
    parser->limits.max_outgoing_bytes = _DBUS_ONE_MEGABYTE * 127;
    parser->limits.max_message_size = _DBUS_ONE_MEGABYTE * 32;

    /* We set relatively conservative values here since due to the
    way SCM_RIGHTS works we need to preallocate an array for the
    maximum number of file descriptors we can receive. Picking a
    high value here thus translates directly to more memory
    allocation. */
    parser->limits.max_incoming_unix_fds = 1024*4;
    parser->limits.max_outgoing_unix_fds = 1024*4;
    parser->limits.max_message_unix_fds = 1024;

    /* Making this long means the user has to wait longer for an
error
    * message if something screws up, but making it too short means
    * they might see a false failure.
    */
    parser->limits.activation_timeout = 25000; /* 25 seconds */

    /* Making this long risks making a DOS attack easier, but too
short
    * and legitimate auth will fail. If interactive auth (ask user
for

```

```

    * password) is allowed, then potentially it has to be quite
long.
    */
    parser->limits.auth_timeout = 30000; /* 30 seconds */

    parser->limits.max_incomplete_connections = 64;
    parser->limits.max_connections_per_user = 256;

    /* Note that max_completed_connections /
max_connections_per_user
    * is the number of users that would have to work together to
    * DOS all the other users.
    */
    parser->limits.max_completed_connections = 2048;

    parser->limits.max_pending_activations = 512;
    parser->limits.max_services_per_connection = 512;

    /* For this one, keep in mind that it isn't only the memory used
    * by the match rules, but slowdown from linearly walking a big
    * list of them. A client adding more than this is almost
    * certainly a bad idea for that reason, and should change to a
    * smaller number of wider-net match rules - getting every last
    * message to the bus is probably better than having a thousand
    * match rules.
    */
    parser->limits.max_match_rules_per_connection = 512;

    parser->limits.reply_timeout = -1; /* never */

    /* this is effectively a limit on message queue size for
messages
    * that require a reply
    */
    parser->limits.max_replies_per_connection = 1024*8;
}

parser->refcount = 1;

return parser;
}

BusConfigParser *
bus_config_parser_ref (BusConfigParser *parser)
{
    _dbus_assert (parser->refcount > 0);

    parser->refcount += 1;

    return parser;
}

```

```

void
bus_config_parser_unref (BusConfigParser *parser)
{
    _dbus_assert (parser->refcount > 0);

    parser->refcount -= 1;

    if (parser->refcount == 0)
    {
        while (parser->stack != NULL)
            pop_element (parser);

        dbus_free (parser->user);
        dbus_free (parser->servicehelper);
        dbus_free (parser->bus_type);
        dbus_free (parser->pidfile);

        _dbus_list_foreach (&parser->listen_on,
                           (DBusForeachFunction) dbus_free,
                           NULL);

        _dbus_list_clear (&parser->listen_on);

        _dbus_list_foreach (&parser->service_dirs,
                           (DBusForeachFunction) dbus_free,
                           NULL);

        _dbus_list_clear (&parser->service_dirs);

        _dbus_list_foreach (&parser->conf_dirs,
                           (DBusForeachFunction) dbus_free,
                           NULL);

        _dbus_list_clear (&parser->conf_dirs);

        _dbus_list_foreach (&parser->mechanisms,
                           (DBusForeachFunction) dbus_free,
                           NULL);

        _dbus_list_clear (&parser->mechanisms);

        _dbus_string_free (&parser->basedir);

        if (parser->policy)
            bus_policy_unref (parser->policy);

        if (parser->service_context_table)
            _dbus_hash_table_unref (parser->service_context_table);

        dbus_free (parser);
    }
}

```

```

dbus_bool_t
bus_config_parser_check_doctype (BusConfigParser *parser,
                                const char *doctype,
                                DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (strcmp (doctype, "busconfig") != 0)
    {
        dbus_set_error (error,
                        DBUS_ERROR_FAILED,
                        "Configuration file has the wrong document type
%s",
                        doctype);

        return FALSE;
    }
    else
        return TRUE;
}

typedef struct
{
    const char *name;
    const char **retloc;
} LocateAttr;

static dbus_bool_t
locate_attributes (BusConfigParser *parser,
                  const char *element_name,
                  const char **attribute_names,
                  const char **attribute_values,
                  DBusError *error,
                  const char *first_attribute_name,
                  const char **first_attribute_retloc,
                  ...)
{
    va_list args;
    const char *name;
    const char **retloc;
    int n_attrs;
#define MAX_ATTRS 24
    LocateAttr attrs[MAX_ATTRS];
    dbus_bool_t retval;
    int i;

    _dbus_assert (first_attribute_name != NULL);
    _dbus_assert (first_attribute_retloc != NULL);

    retval = TRUE;

    n_attrs = 1;

```

```

attrs[0].name = first_attribute_name;
attrs[0].retloc = first_attribute_retloc;
*first_attribute_retloc = NULL;

va_start (args, first_attribute_retloc);

name = va_arg (args, const char*);
retloc = va_arg (args, const char**);

while (name != NULL)
{
    _dbus_assert (retloc != NULL);
    _dbus_assert (n_attrs < MAX_ATTRS);

    attrs[n_attrs].name = name;
    attrs[n_attrs].retloc = retloc;
    n_attrs += 1;
    *retloc = NULL;

    name = va_arg (args, const char*);
    retloc = va_arg (args, const char**);
}

va_end (args);

i = 0;
while (attribute_names[i])
{
    int j;
    dbus_bool_t found;

    found = FALSE;
    j = 0;
    while (j < n_attrs)
    {
        if (strcmp (attrs[j].name, attribute_names[i]) == 0)
        {
            retloc = attrs[j].retloc;

            if (*retloc != NULL)
            {
                dbus_set_error (error, DBUS_ERROR_FAILED,
                    "Attribute \"%s\" repeated twice on
the same <%s> element",
                    attrs[j].name, element_name);
                retval = FALSE;
                goto out;
            }

            *retloc = attribute_values[i];
            found = TRUE;
        }
        j++;
    }
}

```

```

        ++j;
    }

    if (!found)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Attribute \"%s\" is invalid on <%s> element
in this context",
            attribute_names[i], element_name);
        retval = FALSE;
        goto out;
    }

    ++i;
}

out:
return retval;
}

static dbus_bool_t
check_no_attributes (BusConfigParser *parser,
                    const char *element_name,
                    const char **attribute_names,
                    const char **attribute_values,
                    DBusError *error)
{
    if (attribute_names[0] != NULL)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Attribute \"%s\" is invalid on <%s> element in
this context",
            attribute_names[0], element_name);
        return FALSE;
    }

    return TRUE;
}

static dbus_bool_t
start_busconfig_child (BusConfigParser *parser,
                    const char *element_name,
                    const char **attribute_names,
                    const char **attribute_values,
                    DBusError *error)
{
    ElementType element_type;

    element_type = bus_config_parser_element_name_to_type
(element_name);

```



```

if (element_type == ELEMENT_USER)
{
    if (!check_no_attributes (parser, "user", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_USER) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_CONFIGTYPE)
{
    if (!check_no_attributes (parser, "type", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_CONFIGTYPE) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_FORK)
{
    if (!check_no_attributes (parser, "fork", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_FORK) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    parser->fork = TRUE;

    return TRUE;
}
else if (element_type == ELEMENT_SYSLOG)
{
    if (!check_no_attributes (parser, "syslog", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_SYSLOG) == NULL)
    {

```

```

        BUS_SET_OOM (error);
        return FALSE;
    }

    parser->syslog = TRUE;

    return TRUE;
}
else if (element_type == ELEMENT_KEEP_UMASK)
{
    if (!check_no_attributes (parser, "keep_umask", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_KEEP_UMASK) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    parser->keep_umask = TRUE;

    return TRUE;
}
else if (element_type == ELEMENT_PIDFILE)
{
    if (!check_no_attributes (parser, "pidfile", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_PIDFILE) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_LISTEN)
{
    if (!check_no_attributes (parser, "listen", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_LISTEN) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
}

```

```

else if (element_type == ELEMENT_AUTH)
{
    if (!check_no_attributes (parser, "auth", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_AUTH) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_SERVICEHELPER)
{
    if (!check_no_attributes (parser, "servicehelper",
attribute_names, attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_SERVICEHELPER) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_INCLUDEDIR)
{
    if (!check_no_attributes (parser, "includedir", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_INCLUDEDIR) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_STANDARD_SESSION_SERVICEDIRS)
{
    DBusList *link;
    DBusList *dirs;
    dirs = NULL;

    if (!check_no_attributes (parser,
"standard_session_servicedirs", attribute_names, attribute_values,
error))
        return FALSE;

```

```

    if (push_element (parser, ELEMENT_STANDARD_SESSION_SERVICEDIRS)
== NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!_dbus_get_standard_session_servicedirs (&dirs))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    while ((link = _dbus_list_pop_first_link (&dirs))
        service_dirs_append_link_unique_or_free (&parser-
>service_dirs, link);

        return TRUE;
    }
    else if (element_type == ELEMENT_STANDARD_SYSTEM_SERVICEDIRS)
    {
        DBusList *link;
        DBusList *dirs;
        dirs = NULL;

        if (!check_no_attributes (parser, "standard_system_servicedirs",
attribute_names, attribute_values, error))
            return FALSE;

        if (push_element (parser, ELEMENT_STANDARD_SYSTEM_SERVICEDIRS)
== NULL)
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        if (!_dbus_get_standard_system_servicedirs (&dirs))
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        while ((link = _dbus_list_pop_first_link (&dirs))
            service_dirs_append_link_unique_or_free (&parser-
>service_dirs, link);

            return TRUE;
        }
    else if (element_type == ELEMENT_ALLOW_ANONYMOUS)
    {

```

```

    if (!check_no_attributes (parser, "allow_anonymous",
attribute_names, attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_ALLOW_ANONYMOUS) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    parser->allow_anonymous = TRUE;
    return TRUE;
}
else if (element_type == ELEMENT_SERVICEDIR)
{
    if (!check_no_attributes (parser, "servicedir", attribute_names,
attribute_values, error))
        return FALSE;

    if (push_element (parser, ELEMENT_SERVICEDIR) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_INCLUDE)
{
    Element *e;
    const char *if_selinux_enabled;
    const char *ignore_missing;
    const char *selinux_root_relative;

    if ((e = push_element (parser, ELEMENT_INCLUDE)) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    e->d.include.ignore_missing = FALSE;
    e->d.include.if_selinux_enabled = FALSE;
    e->d.include.selinux_root_relative = FALSE;

    if (!locate_attributes (parser, "include",
        attribute_names,
        attribute_values,
        error,
        "ignore_missing", &ignore_missing,
        "if_selinux_enabled",
&if_selinux_enabled,

```

```

        "selinux_root_relative",
&selinux_root_relative,
        NULL))
    return FALSE;

    if (ignore_missing != NULL)
    {
        if (strcmp (ignore_missing, "yes") == 0)
            e->d.include.ignore_missing = TRUE;
        else if (strcmp (ignore_missing, "no") == 0)
            e->d.include.ignore_missing = FALSE;
        else
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
value \"yes\" or \"no\");
                "ignore_missing attribute must have
            return FALSE;
        }
    }

    if (if_selinux_enabled != NULL)
    {
        if (strcmp (if_selinux_enabled, "yes") == 0)
            e->d.include.if_selinux_enabled = TRUE;
        else if (strcmp (if_selinux_enabled, "no") == 0)
            e->d.include.if_selinux_enabled = FALSE;
        else
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
value"
                "if_selinux_enabled attribute must have
            " \"yes\" or \"no\");
            return FALSE;
        }
    }

    if (selinux_root_relative != NULL)
    {
        if (strcmp (selinux_root_relative, "yes") == 0)
            e->d.include.selinux_root_relative = TRUE;
        else if (strcmp (selinux_root_relative, "no") == 0)
            e->d.include.selinux_root_relative = FALSE;
        else
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
have value"
                "selinux_root_relative attribute must
            " \"yes\" or \"no\");
            return FALSE;
        }
    }
}

```

```

    return TRUE;
}
else if (element_type == ELEMENT_POLICY)
{
    Element *e;
    const char *context;
    const char *user;
    const char *group;
    const char *at_console;

    if ((e = push_element (parser, ELEMENT_POLICY)) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    e->d.policy.type = POLICY_IGNORED;

    if (!locate_attributes (parser, "policy",
                           attribute_names,
                           attribute_values,
                           error,
                           "context", &context,
                           "user", &user,
                           "group", &group,
                           "at_console", &at_console,
                           NULL))

        return FALSE;

    if (((context && user) ||
         (context && group) ||
         (context && at_console)) ||
        ((user && group) ||
         (user && at_console) ||
         (group && at_console) ||
         !(context || user || group || at_console))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                           "<policy> element must have exactly one of
(context|user|group|at_console) attributes");
            return FALSE;
        }

    if (context != NULL)
    {
        if (strcmp (context, "default") == 0)
        {
            e->d.policy.type = POLICY_DEFAULT;
        }
        else if (strcmp (context, "mandatory") == 0)
        {
            e->d.policy.type = POLICY_MANDATORY;
        }
    }
}

```

```

    }
    else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "context attribute on <policy> must have
the value \"default\" or \"mandatory\", not \"%s\"",
            context);
        return FALSE;
    }
}
else if (user != NULL)
{
    DBusString username;
    _dbus_string_init_const (&username, user);

    if (_dbus_parse_unix_user_from_config (&username,
        &e-
>d.policy.gid_uid_or_at_console))
        e->d.policy.type = POLICY_USER;
    else
        _dbus_warn ("Unknown username \"%s\" in message bus
configuration file\n",
            user);
}
else if (group != NULL)
{
    DBusString group_name;
    _dbus_string_init_const (&group_name, group);

    if (_dbus_parse_unix_group_from_config (&group_name,
        &e-
>d.policy.gid_uid_or_at_console))
        e->d.policy.type = POLICY_GROUP;
    else
        _dbus_warn ("Unknown group \"%s\" in message bus
configuration file\n",
            group);
}
else if (at_console != NULL)
{
    dbus_bool_t t;
    t = (strcmp (at_console, "true") == 0);
    if (t || strcmp (at_console, "false") == 0)
    {
        e->d.policy.gid_uid_or_at_console = t;
        e->d.policy.type = POLICY_CONSOLE;
    }
    else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Unknown value \"%s\" for at_console in
message bus configuration file",

```



```

        at_console);

        return FALSE;
    }
}
else
{
    _dbus_assert_not_reached ("all <policy> attributes null and
we didn't set error");
}

return TRUE;
}
else if (element_type == ELEMENT_LIMIT)
{
    Element *e;
    const char *name;

    if ((e = push_element (parser, ELEMENT_LIMIT)) == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!locate_attributes (parser, "limit",
                           attribute_names,
                           attribute_values,
                           error,
                           "name", &name,
                           NULL))

        return FALSE;

    if (name == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
attribute"");
        return FALSE;
    }

    e->d.limit.name = _dbus_strdup (name);
    if (e->d.limit.name == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    return TRUE;
}
else if (element_type == ELEMENT_SELINUX)
{

```

```

        if (!check_no_attributes (parser, "selinux", attribute_names,
attribute_values, error))
            return FALSE;

        if (push_element (parser, ELEMENT_SELINUX) == NULL)
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        return TRUE;
    }
else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
"Element <%s> not allowed inside <%s> in
configuration file",
                        element_name, "busconfig");
        return FALSE;
    }
}

static dbus_bool_t
append_rule_from_element (BusConfigParser *parser,
                        const char *element_name,
                        const char **attribute_names,
                        const char **attribute_values,
                        dbus_bool_t allow,
                        DBusError *error)
{
    const char *log;
    const char *send_interface;
    const char *send_member;
    const char *send_error;
    const char *send_destination;
    const char *send_path;
    const char *send_type;
    const char *receive_interface;
    const char *receive_member;
    const char *receive_error;
    const char *receive_sender;
    const char *receive_path;
    const char *receive_type;
    const char *eavesdrop;
    const char *send_requested_reply;
    const char *receive_requested_reply;
    const char *own;
    const char *own_prefix;
    const char *user;
    const char *group;

    BusPolicyRule *rule;

```

```

if (!locate_attributes (parser, element_name,
                        attribute_names,
                        attribute_values,
                        error,
                        "send_interface", &send_interface,
                        "send_member", &send_member,
                        "send_error", &send_error,
                        "send_destination", &send_destination,
                        "send_path", &send_path,
                        "send_type", &send_type,
                        "receive_interface", &receive_interface,
                        "receive_member", &receive_member,
                        "receive_error", &receive_error,
                        "receive_sender", &receive_sender,
                        "receive_path", &receive_path,
                        "receive_type", &receive_type,
                        "eavesdrop", &eavesdrop,
                        "send_requested_reply",
&send_requested_reply,
                        "receive_requested_reply",
&receive_requested_reply,
                        "own", &own,
                        "own_prefix", &own_prefix,
                        "user", &user,
                        "group", &group,
                        "log", &log,
                        NULL))

    return FALSE;

if (!(send_interface || send_member || send_error ||
send_destination ||
    send_type || send_path ||
    receive_interface || receive_member || receive_error ||
receive_sender ||
    receive_type || receive_path || eavesdrop ||
    send_requested_reply || receive_requested_reply ||
    own || own_prefix || user || group))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                    "Element <%s> must have one or more attributes",
                    element_name);
    return FALSE;
}

if ((send_member && (send_interface == NULL && send_path == NULL))
||
    (receive_member && (receive_interface == NULL && receive_path ==
NULL)))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,

```

```

        "On element <%s>, if you specify a member you
must specify an interface or a path. Keep in mind that not all
messages have an interface field.",
        element_name);
    return FALSE;
}

/* Allowed combinations of elements are:
*
*   base, must be all send or all receive:
*       nothing
*       interface
*       interface + member
*       error
*
*   base send_ can combine with send_destination, send_path,
send_type, send_requested_reply
*   base receive_ with receive_sender, receive_path, receive_type,
receive_requested_reply, eavesdrop
*
*   user, group, own, own_prefix must occur alone
*
* Pretty sure the below stuff is broken, FIXME think about it more.
*/

if ((send_interface && (send_error ||
    receive_interface ||
    receive_member ||
    receive_error ||
    receive_sender ||
    receive_requested_reply ||
    own || own_prefix ||
    user ||
    group)) ||

    (send_member && (send_error ||
    receive_interface ||
    receive_member ||
    receive_error ||
    receive_sender ||
    receive_requested_reply ||
    own || own_prefix ||
    user ||
    group)) ||

    (send_error && (receive_interface ||
    receive_member ||
    receive_error ||
    receive_sender ||
    receive_requested_reply ||
    own || own_prefix ||
    user ||

```

```

        group)) ||

(send_destination && (receive_interface ||
                    receive_member ||
                    receive_error ||
                    receive_sender ||
                    receive_requested_reply ||
                    own || own_prefix ||
                    user ||
                    group)) ||

(send_type && (receive_interface ||
             receive_member ||
             receive_error ||
             receive_sender ||
             receive_requested_reply ||
             own || own_prefix ||
             user ||
             group)) ||

(send_path && (receive_interface ||
             receive_member ||
             receive_error ||
             receive_sender ||
             receive_requested_reply ||
             own || own_prefix ||
             user ||
             group)) ||

(send_requested_reply && (receive_interface ||
                       receive_member ||
                       receive_error ||
                       receive_sender ||
                       receive_requested_reply ||
                       own || own_prefix ||
                       user ||
                       group)) ||

(receive_interface && (receive_error ||
                    own || own_prefix ||
                    user ||
                    group)) ||

(receive_member && (receive_error ||
                  own || own_prefix ||
                  user ||
                  group)) ||

(receive_error && (own || own_prefix ||
                 user ||
                 group)) ||

```

```

(eavesdrop && (own || own_prefix ||
              user ||
              group)) ||

(receive_requested_reply && (own || own_prefix ||
                             user ||
                             group)) ||

(own && (own_prefix || user || group)) ||

(own_prefix && (own || user || group)) ||

(user && group))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                  "Invalid combination of attributes on element
<%s>",
                  element_name);
    return FALSE;
}

rule = NULL;

/* In BusPolicyRule, NULL represents wildcard.
 * In the config file, '*' represents it.
 */
#define IS_WILDCARD(str) ((str) && ((str)[0]) == '*' && ((str)[1]) ==
'\0')

if (send_interface || send_member || send_error || send_destination
||
    send_path || send_type || send_requested_reply)
{
    int message_type;

    if (IS_WILDCARD (send_interface))
        send_interface = NULL;
    if (IS_WILDCARD (send_member))
        send_member = NULL;
    if (IS_WILDCARD (send_error))
        send_error = NULL;
    if (IS_WILDCARD (send_destination))
        send_destination = NULL;
    if (IS_WILDCARD (send_path))
        send_path = NULL;
    if (IS_WILDCARD (send_type))
        send_type = NULL;

    message_type = DBUS_MESSAGE_TYPE_INVALID;
    if (send_type != NULL)
    {
        message_type = dbus_message_type_from_string (send_type);
    }
}

```

```

        if (message_type == DBUS_MESSAGE_TYPE_INVALID)
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "Bad message type \"%s\"",
                send_type);
            return FALSE;
        }
    }

    if (eavesdrop &&
        !(strcmp (eavesdrop, "true") == 0 ||
          strcmp (eavesdrop, "false") == 0))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Bad value \"%s\" for %s attribute, must be
true or false",
            "eavesdrop", eavesdrop);
        return FALSE;
    }

    if (send_requested_reply &&
        !(strcmp (send_requested_reply, "true") == 0 ||
          strcmp (send_requested_reply, "false") == 0))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Bad value \"%s\" for %s attribute, must be
true or false",
            "send_requested_reply",
send_requested_reply);
        return FALSE;
    }

    rule = bus_policy_rule_new (BUS_POLICY_RULE_SEND, allow);
    if (rule == NULL)
        goto nomem;

    if (eavesdrop)
        rule->d.send.eavesdrop = (strcmp (eavesdrop, "true") == 0);

    if (log)
        rule->d.send.log = (strcmp (log, "true") == 0);

    if (send_requested_reply)
        rule->d.send.requested_reply = (strcmp (send_requested_reply,
"true") == 0);

    rule->d.send.message_type = message_type;
    rule->d.send.path = _dbus_strdup (send_path);
    rule->d.send.interface = _dbus_strdup (send_interface);
    rule->d.send.member = _dbus_strdup (send_member);
    rule->d.send.error = _dbus_strdup (send_error);
    rule->d.send.destination = _dbus_strdup (send_destination);

```

```

if (send_path && rule->d.send.path == NULL)
    goto nomem;
if (send_interface && rule->d.send.interface == NULL)
    goto nomem;
if (send_member && rule->d.send.member == NULL)
    goto nomem;
if (send_error && rule->d.send.error == NULL)
    goto nomem;
if (send_destination && rule->d.send.destination == NULL)
    goto nomem;
}
else if (receive_interface || receive_member || receive_error ||
receive_sender ||
        receive_path || receive_type || eavesdrop ||
receive_requested_reply)
{
    int message_type;

    if (IS_WILDCARD (receive_interface))
        receive_interface = NULL;
    if (IS_WILDCARD (receive_member))
        receive_member = NULL;
    if (IS_WILDCARD (receive_error))
        receive_error = NULL;
    if (IS_WILDCARD (receive_sender))
        receive_sender = NULL;
    if (IS_WILDCARD (receive_path))
        receive_path = NULL;
    if (IS_WILDCARD (receive_type))
        receive_type = NULL;

    message_type = DBUS_MESSAGE_TYPE_INVALID;
    if (receive_type != NULL)
    {
        message_type = dbus_message_type_from_string (receive_type);
        if (message_type == DBUS_MESSAGE_TYPE_INVALID)
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "Bad message type \"%s\"",
                receive_type);
            return FALSE;
        }
    }

    if (eavesdrop &&
        !(strcmp (eavesdrop, "true") == 0 ||
        strcmp (eavesdrop, "false") == 0))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Bad value \"%s\" for %s attribute, must be
true or false",

```



```

        "eavesdrop", eavesdrop);
    return FALSE;
}

if (receive_requested_reply &&
    !(strcmp (receive_requested_reply, "true") == 0 ||
      strcmp (receive_requested_reply, "false") == 0))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                   "Bad value \"%s\" for %s attribute, must be
true or false",
                   "receive_requested_reply",
receive_requested_reply);
    return FALSE;
}

rule = bus_policy_rule_new (BUS_POLICY_RULE_RECEIVE, allow);
if (rule == NULL)
    goto nomem;

if (eavesdrop)
    rule->d.receive.eavesdrop = (strcmp (eavesdrop, "true") == 0);

if (receive_requested_reply)
    rule->d.receive.requested_reply = (strcmp
(receive_requested_reply, "true") == 0);

rule->d.receive.message_type = message_type;
rule->d.receive.path = _dbus_strdup (receive_path);
rule->d.receive.interface = _dbus_strdup (receive_interface);
rule->d.receive.member = _dbus_strdup (receive_member);
rule->d.receive.error = _dbus_strdup (receive_error);
rule->d.receive.origin = _dbus_strdup (receive_sender);

if (receive_path && rule->d.receive.path == NULL)
    goto nomem;
if (receive_interface && rule->d.receive.interface == NULL)
    goto nomem;
if (receive_member && rule->d.receive.member == NULL)
    goto nomem;
if (receive_error && rule->d.receive.error == NULL)
    goto nomem;
if (receive_sender && rule->d.receive.origin == NULL)
    goto nomem;
}
else if (own || own_prefix)
{
    rule = bus_policy_rule_new (BUS_POLICY_RULE_OWN, allow);
    if (rule == NULL)
        goto nomem;

    if (own)

```

```

    {
        if (IS_WILDCARD (own))
            own = NULL;

        rule->d.own.prefix = 0;
        rule->d.own.service_name = _dbus_strdup (own);
        if (own && rule->d.own.service_name == NULL)
            goto nomem;
    }
else
    {
        rule->d.own.prefix = 1;
        rule->d.own.service_name = _dbus_strdup (own_prefix);
        if (rule->d.own.service_name == NULL)
            goto nomem;
    }
}
else if (user)
    {
        if (IS_WILDCARD (user))
            {
                rule = bus_policy_rule_new (BUS_POLICY_RULE_USER, allow);
                if (rule == NULL)
                    goto nomem;

                rule->d.user.uid = DBUS_UID_UNSET;
            }
        else
            {
                DBusString username;
                dbus_uid_t uid;

                _dbus_string_init_const (&username, user);

                if (_dbus_parse_unix_user_from_config (&username, &uid))
                    {
                        rule = bus_policy_rule_new (BUS_POLICY_RULE_USER,
allow);

                        if (rule == NULL)
                            goto nomem;

                        rule->d.user.uid = uid;
                    }
                else
                    {
                        _dbus_warn ("Unknown username \"%s\" on element <%s>\n",
user, element_name);
                    }
            }
    }
else if (group)
    {

```

```

if (IS_WILDCARD (group))
{
    rule = bus_policy_rule_new (BUS_POLICY_RULE_GROUP, allow);
    if (rule == NULL)
        goto nomem;

    rule->d.group.gid = DBUS_GID_UNSET;
}
else
{
    DBusString groupname;
    dbus_gid_t gid;

    _dbus_string_init_const (&groupname, group);

    if (_dbus_parse_unix_group_from_config (&groupname, &gid))
    {
        rule = bus_policy_rule_new (BUS_POLICY_RULE_GROUP,
allow);
        if (rule == NULL)
            goto nomem;

        rule->d.group.gid = gid;
    }
    else
    {
        _dbus_warn ("Unknown group \"%s\" on element <%s>\n",
                    group, element_name);
    }
}
}
else
    _dbus_assert_not_reached ("Did not handle some combination of
attributes on <allow> or <deny>");

if (rule != NULL)
{
    Element *pe;

    pe = peek_element (parser);
    _dbus_assert (pe != NULL);
    _dbus_assert (pe->type == ELEMENT_POLICY);

    switch (pe->d.policy.type)
    {
        case POLICY_IGNORED:
            /* drop the rule on the floor */
            break;

        case POLICY_DEFAULT:
            if (!bus_policy_append_default_rule (parser->policy, rule))
                goto nomem;
    }
}

```

```

        break;
    case POLICY_MANDATORY:
        if (!bus_policy_append_mandatory_rule (parser->policy,
rule))
            goto nomem;
        break;
    case POLICY_USER:
        if (!BUS_POLICY_RULE_IS_PER_CLIENT (rule))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "<%s> rule cannot be per-user because it
has bus-global semantics",
                element_name);
            goto failed;
        }

        if (!bus_policy_append_user_rule (parser->policy, pe-
>d.policy.gid_uid_or_at_console,
                rule))
            goto nomem;
        break;
    case POLICY_GROUP:
        if (!BUS_POLICY_RULE_IS_PER_CLIENT (rule))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "<%s> rule cannot be per-group because
it has bus-global semantics",
                element_name);
            goto failed;
        }

        if (!bus_policy_append_group_rule (parser->policy, pe-
>d.policy.gid_uid_or_at_console,
                rule))
            goto nomem;
        break;

    case POLICY_CONSOLE:
        if (!bus_policy_append_console_rule (parser->policy, pe-
>d.policy.gid_uid_or_at_console,
                rule))
            goto nomem;
        break;
    }

    bus_policy_rule_unref (rule);
    rule = NULL;
}

return TRUE;

```

```

nomen:
    BUS_SET_OOM (error);
failed:
    if (rule)
        bus_policy_rule_unref (rule);
    return FALSE;
}

static dbus_bool_t
start_policy_child (BusConfigParser *parser,
                   const char *element_name,
                   const char **attribute_names,
                   const char **attribute_values,
                   DBusError *error)
{
    if (strcmp (element_name, "allow") == 0)
    {
        if (!append_rule_from_element (parser, element_name,
attribute_values,
                                     TRUE, error))
            return FALSE;

        if (push_element (parser, ELEMENT_ALLOW) == NULL)
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        return TRUE;
    }
    else if (strcmp (element_name, "deny") == 0)
    {
        if (!append_rule_from_element (parser, element_name,
attribute_values,
                                     FALSE, error))
            return FALSE;

        if (push_element (parser, ELEMENT_DENY) == NULL)
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        return TRUE;
    }
    else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
"Element <%s> not allowed inside <%s> in
configuration file",

```

```

        element_name, "policy");
    return FALSE;
}
}

static dbus_bool_t
start_selinux_child (BusConfigParser *parser,
                    const char *element_name,
                    const char **attribute_names,
                    const char **attribute_values,
                    DBusError *error)
{
    char *own_copy;
    char *context_copy;

    own_copy = NULL;
    context_copy = NULL;

    if (strcmp (element_name, "associate") == 0)
    {
        const char *own;
        const char *context;

        if (!locate_attributes (parser, "associate",
                                attribute_names,
                                attribute_values,
                                error,
                                "own", &own,
                                "context", &context,
                                NULL))

            return FALSE;

        if (push_element (parser, ELEMENT_ASSOCIATE) == NULL)
        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        if (own == NULL || context == NULL)
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                            "Element <associate> must have attributes
own=\"<servicename>\" and context=\"<selinux context>\"");
            return FALSE;
        }

        own_copy = _dbus_strdup (own);
        if (own_copy == NULL)
            goto oom;
        context_copy = _dbus_strdup (context);
        if (context_copy == NULL)
            goto oom;
    }
}

```

```

        if (!_dbus_hash_table_insert_string (parser-
>service_context_table,
                                own_copy, context_copy))
            goto oom;

        return TRUE;
    }
    else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Element <%s> not allowed inside <%s> in
configuration file",
            element_name, "selinux");
        return FALSE;
    }

oom:
    if (own_copy)
        dbus_free (own_copy);

    if (context_copy)
        dbus_free (context_copy);

    BUS_SET_OOM (error);
    return FALSE;
}

dbus_bool_t
bus_config_parser_start_element (BusConfigParser    *parser,
                                const char          *element_name,
                                const char          **attribute_names,
                                const char          **attribute_values,
                                DBusError          *error)
{
    ElementType t;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    /* printf ("START: %s\n", element_name); */

    t = top_element_type (parser);

    if (t == ELEMENT_NONE)
    {
        if (strcmp (element_name, "busconfig") == 0)
        {
            if (!check_no_attributes (parser, "busconfig",
attribute_names, attribute_values, error))
                return FALSE;

            if (push_element (parser, ELEMENT_BUSCONFIG) == NULL)

```

```

        {
            BUS_SET_OOM (error);
            return FALSE;
        }

        return TRUE;
    }
    else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "Unknown element <%s> at root of
configuration file",
            element_name);
        return FALSE;
    }
}
else if (t == ELEMENT_BUSCONFIG)
{
    return start_busconfig_child (parser, element_name,
        attribute_names, attribute_values,
        error);
}
else if (t == ELEMENT_POLICY)
{
    return start_policy_child (parser, element_name,
        attribute_names, attribute_values,
        error);
}
else if (t == ELEMENT_SELINUX)
{
    return start_selinux_child (parser, element_name,
        attribute_names, attribute_values,
        error);
}
else
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
        "Element <%s> is not allowed in this context",
        element_name);
    return FALSE;
}
}

static dbus_bool_t
set_limit (BusConfigParser *parser,
           const char      *name,
           long             value,
           DBusError        *error)
{
    dbus_bool_t must_be_positive;
    dbus_bool_t must_be_int;

```



```
must_be_int = FALSE;
must_be_positive = FALSE;

if (strcmp (name, "max_incoming_bytes") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_incoming_bytes = value;
}
else if (strcmp (name, "max_incoming_unix_fds") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_incoming_unix_fds = value;
}
else if (strcmp (name, "max_outgoing_bytes") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_outgoing_bytes = value;
}
else if (strcmp (name, "max_outgoing_unix_fds") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_outgoing_unix_fds = value;
}
else if (strcmp (name, "max_message_size") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_message_size = value;
}
else if (strcmp (name, "max_message_unix_fds") == 0)
{
    must_be_positive = TRUE;
    parser->limits.max_message_unix_fds = value;
}
else if (strcmp (name, "service_start_timeout") == 0)
{
    must_be_positive = TRUE;
    must_be_int = TRUE;
    parser->limits.activation_timeout = value;
}
else if (strcmp (name, "auth_timeout") == 0)
{
    must_be_positive = TRUE;
    must_be_int = TRUE;
    parser->limits.auth_timeout = value;
}
else if (strcmp (name, "reply_timeout") == 0)
{
    must_be_positive = TRUE;
    must_be_int = TRUE;
    parser->limits.reply_timeout = value;
}
else if (strcmp (name, "max_completed_connections") == 0)
```

```

    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_completed_connections = value;
    }
else if (strcmp (name, "max_incomplete_connections") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_incomplete_connections = value;
    }
else if (strcmp (name, "max_connections_per_user") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_connections_per_user = value;
    }
else if (strcmp (name, "max_pending_service_starts") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_pending_activations = value;
    }
else if (strcmp (name, "max_names_per_connection") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_services_per_connection = value;
    }
else if (strcmp (name, "max_match_rules_per_connection") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_match_rules_per_connection = value;
    }
else if (strcmp (name, "max_replies_per_connection") == 0)
    {
        must_be_positive = TRUE;
        must_be_int = TRUE;
        parser->limits.max_replies_per_connection = value;
    }
else
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "There is no limit called \"%s\"\n",
                       name);
        return FALSE;
    }

if (must_be_positive && value < 0)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,

```

```

        "<limit name=\"%s\"> must be a positive
number\n",
        name);
    return FALSE;
}

if (must_be_int &&
    (value < _DBUS_INT_MIN || value > _DBUS_INT_MAX))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
        "<limit name=\"%s\"> value is too large\n",
        name);
    return FALSE;
}

return TRUE;
}

dbus_bool_t
bus_config_parser_end_element (BusConfigParser *parser,
                               const char *element_name,
                               DBusError *error)
{
    ElementType t;
    const char *n;
    Element *e;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    /* printf ("END: %s\n", element_name); */

    t = top_element_type (parser);

    if (t == ELEMENT_NONE)
    {
        /* should probably be an assertion failure but
         * being paranoid about XML parsers
         */
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "XML parser ended element with no element on the
stack");
        return FALSE;
    }

    n = bus_config_parser_element_type_to_name (t);
    _dbus_assert (n != NULL);
    if (strcmp (n, element_name) != 0)
    {
        /* should probably be an assertion failure but
         * being paranoid about XML parsers
         */
        dbus_set_error (error, DBUS_ERROR_FAILED,

```

```

        "XML element <%s> ended but topmost element on
the stack was <%s>",
        element_name, n);
    return FALSE;
}

e = peek_element (parser);
_dbus_assert (e != NULL);

switch (e->type)
{
case ELEMENT_NONE:
    _dbus_assert_not_reached ("element in stack has no type");
    break;

case ELEMENT_INCLUDE:
case ELEMENT_USER:
case ELEMENT_CONFIGTYPE:
case ELEMENT_LISTEN:
case ELEMENT_PIDFILE:
case ELEMENT_AUTH:
case ELEMENT_SERVICEDIR:
case ELEMENT_SERVICEHELPER:
case ELEMENT_INCLUDEDIR:
case ELEMENT_LIMIT:
    if (!e->had_content)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
content inside it",
        bus_config_parser_element_type_to_name (e-
>type));
        return FALSE;
    }

    if (e->type == ELEMENT_LIMIT)
    {
        if (!set_limit (parser, e->d.limit.name, e->d.limit.value,
error))
            return FALSE;
    }
    break;

case ELEMENT_BUSCONFIG:
case ELEMENT_POLICY:
case ELEMENT_ALLOW:
case ELEMENT_DENY:
case ELEMENT_FORK:
case ELEMENT_SYSLOG:
case ELEMENT_KEEP_UMASK:
case ELEMENT_SELINUX:
case ELEMENT_ASSOCIATE:

```

```

        case ELEMENT_STANDARD_SESSION_SERVICEDIRS:
        case ELEMENT_STANDARD_SYSTEM_SERVICEDIRS:
        case ELEMENT_ALLOW_ANONYMOUS:
            break;
    }

    pop_element (parser);

    return TRUE;
}

static dbus_bool_t
all_whitespace (const DBusString *str)
{
    int i;

    _dbus_string_skip_white (str, 0, &i);

    return i == _dbus_string_get_length (str);
}

static dbus_bool_t
make_full_path (const DBusString *basedir,
                const DBusString *filename,
                DBusString *full_path)
{
    if (_dbus_path_is_absolute (filename))
    {
        return _dbus_string_copy (filename, 0, full_path, 0);
    }
    else
    {
        if (!_dbus_string_copy (basedir, 0, full_path, 0))
            return FALSE;

        if (!_dbus_concat_dir_and_file (full_path, filename))
            return FALSE;

        return TRUE;
    }
}

static dbus_bool_t
include_file (BusConfigParser *parser,
             const DBusString *filename,
             dbus_bool_t ignore_missing,
             DBusError *error)
{
    /* FIXME good test case for this would load each config file in the
     * test suite both alone, and as an include, and check
     * that the result is the same
     */

```

```

BusConfigParser *included;
const char *filename_str;
DBusError tmp_error;

dbus_error_init (&tmp_error);

filename_str = _dbus_string_get_const_data (filename);

/* Check to make sure this file hasn't already been included. */
if (seen_include (parser, filename))
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                   "Circular inclusion of file '%s'",
                   filename_str);
    return FALSE;
}

if (! _dbus_list_append (&parser->included_files, (void *)
filename_str))
{
    BUS_SET_OOM (error);
    return FALSE;
}

/* Since parser is passed in as the parent, included
   inherits parser's limits. */
included = bus_config_load (filename, FALSE, parser, &tmp_error);

_dbus_list_pop_last (&parser->included_files);

if (included == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);

    if (dbus_error_has_name (&tmp_error, DBUS_ERROR_FILE_NOT_FOUND)
&&
        ignore_missing)
    {
        dbus_error_free (&tmp_error);
        return TRUE;
    }
    else
    {
        dbus_move_error (&tmp_error, error);
        return FALSE;
    }
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);

    if (!merge_included (parser, included, error))

```

```

        {
            bus_config_parser_unref (included);
            return FALSE;
        }

        /* Copy included's limits back to parser. */
        parser->limits = included->limits;

        bus_config_parser_unref (included);
        return TRUE;
    }
}

static dbus_bool_t
servicehelper_path (BusConfigParser *parser,
                   const DBusString *filename,
                   DBusError *error)
{
    const char *filename_str;
    char *servicehelper;

    filename_str = _dbus_string_get_const_data (filename);

    /* copy to avoid overwriting with NULL on OOM */
    servicehelper = _dbus_strdup (filename_str);

    /* check for OOM */
    if (servicehelper == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    /* save the latest servicehelper only if not OOM */
    dbus_free (parser->servicehelper);
    parser->servicehelper = servicehelper;

    /* We don't check whether the helper exists; instead we
     * would just fail to ever activate anything if it doesn't.
     * This allows an admin to fix the problem if it doesn't exist.
     * It also allows the parser test suite to successfully parse
     * test cases without installing the helper. ;-)
     */

    return TRUE;
}

static dbus_bool_t
include_dir (BusConfigParser *parser,
            const DBusString *dirname,
            DBusError *error)
{

```

```

DBusString filename;
dbus_bool_t retval;
DBusError tmp_error;
DBusDirIter *dir;
char *s;

if (!_dbus_string_init (&filename))
{
    BUS_SET_OOM (error);
    return FALSE;
}

retval = FALSE;

dir = _dbus_directory_open (dirname, error);

if (dir == NULL)
    goto failed;

dbus_error_init (&tmp_error);
while (_dbus_directory_get_next_file (dir, &filename, &tmp_error))
{
    DBusString full_path;

    if (!_dbus_string_init (&full_path))
    {
        BUS_SET_OOM (error);
        goto failed;
    }

    if (!_dbus_string_copy (dirname, 0, &full_path, 0))
    {
        BUS_SET_OOM (error);
        _dbus_string_free (&full_path);
        goto failed;
    }

    if (!_dbus_concat_dir_and_file (&full_path, &filename))
    {
        BUS_SET_OOM (error);
        _dbus_string_free (&full_path);
        goto failed;
    }

    if (_dbus_string_ends_with_c_str (&full_path, ".conf"))
    {
        if (!include_file (parser, &full_path, TRUE, error))
        {
            if (dbus_error_is_set (error))
            {
                /* We log to syslog unconditionally here, because
this is

```



```

        * the configuration parser, so we don't yet know
whether
        * this bus is going to want to write to syslog!
(There's
        * also some layer inversion going on, if we want to
use
        * the bus context.) */
        _dbus_system_log (DBUS_SYSTEM_LOG_INFO,
                        "Encountered error '%s' while
parsing '%s'\n",
                        error->message,
                        _dbus_string_get_const_data
(&full_path));
        dbus_error_free (error);
    }
}

    _dbus_string_free (&full_path);
}

if (dbus_error_is_set (&tmp_error))
{
    dbus_move_error (&tmp_error, error);
    goto failed;
}

if (!_dbus_string_copy_data (dirname, &s))
{
    BUS_SET_OOM (error);
    goto failed;
}

if (!_dbus_list_append (&parser->conf_dirs, s))
{
    dbus_free (s);
    BUS_SET_OOM (error);
    goto failed;
}

retval = TRUE;

failed:
    _dbus_string_free (&filename);

    if (dir)
        _dbus_directory_close (dir);

    return retval;
}

```

```

dbus_bool_t
bus_config_parser_content (BusConfigParser *parser,
                           const DBusString *content,
                           DBusError *error)
{
    Element *e;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

#ifdef 0
    {
        const char *c_str;

        _dbus_string_get_const_data (content, &c_str);

        printf ("CONTENT %d bytes: %s\n", _dbus_string_get_length
                (content), c_str);
    }
#endif

    e = peek_element (parser);
    if (e == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Text content outside of any XML element in
configuration file");
        return FALSE;
    }
    else if (e->had_content)
    {
        _dbus_assert_not_reached ("Element had multiple content
blocks");
        return FALSE;
    }

    switch (top_element_type (parser))
    {
        {
        case ELEMENT_NONE:
            _dbus_assert_not_reached ("element at top of stack has no
type");
            return FALSE;

        case ELEMENT_BUSCONFIG:
        case ELEMENT_POLICY:
        case ELEMENT_ALLOW:
        case ELEMENT_DENY:
        case ELEMENT_FORK:
        case ELEMENT_SYSLOG:
        case ELEMENT_KEEP_UMASK:
        case ELEMENT_STANDARD_SESSION_SERVICEDIRS:
        case ELEMENT_STANDARD_SYSTEM_SERVICEDIRS:
        case ELEMENT_ALLOW_ANONYMOUS:

```

```

case ELEMENT_SELINUX:
case ELEMENT_ASSOCIATE:
    if (all_whitespace (content))
        return TRUE;
    else
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "No text content expected inside XML element
%s in configuration file",
                bus_config_parser_element_type_to_name
(top_element_type (parser)));
            return FALSE;
        }

case ELEMENT_PIDFILE:
    {
        char *s;

        e->had_content = TRUE;

        if (!_dbus_string_copy_data (content, &s))
            goto nomem;

        dbus_free (parser->pidfile);
        parser->pidfile = s;
    }
    break;

case ELEMENT_INCLUDE:
    {
        DBusString full_path, selinux_policy_root;

        e->had_content = TRUE;

        if (e->d.include.if_selinux_enabled
            && !bus_selinux_enabled ())
            break;

        if (!_dbus_string_init (&full_path))
            goto nomem;

        if (e->d.include.selinux_root_relative)
        {
            if (!bus_selinux_get_policy_root ())
            {
                dbus_set_error (error, DBUS_ERROR_FAILED,
                    "Could not determine SELinux policy root for
relative inclusion");
                _dbus_string_free (&full_path);
                return FALSE;
            }
            _dbus_string_init_const (&selinux_policy_root,

```

```

        bus_selinux_get_policy_root ());
    if (!make_full_path (&selinux_policy_root, content,
&full_path))
    {
        _dbus_string_free (&full_path);
        goto nomem;
    }
    else if (!make_full_path (&parser->basedir, content,
&full_path))
    {
        _dbus_string_free (&full_path);
        goto nomem;
    }

    if (!include_file (parser, &full_path,
        e->d.include.ignore_missing, error))
    {
        _dbus_string_free (&full_path);
        return FALSE;
    }

    _dbus_string_free (&full_path);
}
break;

case ELEMENT_SERVICEHELPER:
{
    DBusString full_path;

    e->had_content = TRUE;

    if (!_dbus_string_init (&full_path))
        goto nomem;

    if (!make_full_path (&parser->basedir, content, &full_path))
    {
        _dbus_string_free (&full_path);
        goto nomem;
    }

    if (!servicehelper_path (parser, &full_path, error))
    {
        _dbus_string_free (&full_path);
        return FALSE;
    }

    _dbus_string_free (&full_path);
}
break;

case ELEMENT_INCLUDEDIR:

```

```

{
    DBusString full_path;

    e->had_content = TRUE;

    if (!_dbus_string_init (&full_path))
        goto nomem;

    if (!make_full_path (&parser->basedir, content, &full_path))
    {
        _dbus_string_free (&full_path);
        goto nomem;
    }

    if (!include_dir (parser, &full_path, error))
    {
        _dbus_string_free (&full_path);
        return FALSE;
    }

    _dbus_string_free (&full_path);
}
break;

case ELEMENT_USER:
{
    char *s;

    e->had_content = TRUE;

    if (!_dbus_string_copy_data (content, &s))
        goto nomem;

    dbus_free (parser->user);
    parser->user = s;
}
break;

case ELEMENT_CONFIGTYPE:
{
    char *s;

    e->had_content = TRUE;

    if (!_dbus_string_copy_data (content, &s))
        goto nomem;

    dbus_free (parser->bus_type);
    parser->bus_type = s;
}
break;

```

```

case ELEMENT_LISTEN:
{
    char *s;

    e->had_content = TRUE;

    if (!_dbus_string_copy_data (content, &s))
        goto nomem;

    if (!_dbus_list_append (&parser->listen_on,
                            s))
        {
            dbus_free (s);
            goto nomem;
        }
}
break;

case ELEMENT_AUTH:
{
    char *s;

    e->had_content = TRUE;

    if (!_dbus_string_copy_data (content, &s))
        goto nomem;

    if (!_dbus_list_append (&parser->mechanisms,
                            s))
        {
            dbus_free (s);
            goto nomem;
        }
}
break;

case ELEMENT_SERVICEDIR:
{
    char *s;
    DBusString full_path;

    e->had_content = TRUE;

    if (!_dbus_string_init (&full_path))
        goto nomem;

    if (!make_full_path (&parser->basedir, content, &full_path))
        {
            _dbus_string_free (&full_path);
            goto nomem;
        }
}

```

```

    if (!_dbus_string_copy_data (&full_path, &s))
    {
        _dbus_string_free (&full_path);
        goto nomem;
    }

    /* _only_ extra session directories can be specified */
    if (!service_dirs_append_unique_or_free (&parser->service_dirs, s))
    {
        _dbus_string_free (&full_path);
        dbus_free (s);
        goto nomem;
    }

    _dbus_string_free (&full_path);
}
break;

case ELEMENT_LIMIT:
{
    long val;

    e->had_content = TRUE;

    val = 0;
    if (!_dbus_string_parse_int (content, 0, &val, NULL))
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
            "<limit name=\"%s\"> element has invalid
value (could not parse as integer)",
            e->d.limit.name);
        return FALSE;
    }

    e->d.limit.value = val;

    _dbus_verbose ("Loaded value %ld for limit %s\n",
        e->d.limit.value,
        e->d.limit.name);
}
break;
}

_DBUS_ASSERT_ERROR_IS_CLEAR (error);
return TRUE;

nomem:
BUS_SET_OOM (error);
return FALSE;
}

```

```

dbus_bool_t
bus_config_parser_finished (BusConfigParser *parser,
                           DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (parser->stack != NULL)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Element <%s> was not closed in configuration
file",
                       bus_config_parser_element_type_to_name
(top_element_type (parser)));

        return FALSE;
    }

    if (parser->is_toplevel && parser->listen_on == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Configuration file needs one or more <listen>
elements giving addresses");
        return FALSE;
    }

    return TRUE;
}

const char*
bus_config_parser_get_user (BusConfigParser *parser)
{
    return parser->user;
}

const char*
bus_config_parser_get_type (BusConfigParser *parser)
{
    return parser->bus_type;
}

DBusList**
bus_config_parser_get_addresses (BusConfigParser *parser)
{
    return &parser->listen_on;
}

DBusList**
bus_config_parser_get_mechanisms (BusConfigParser *parser)
{
    return &parser->mechanisms;
}

```



```

DBusList**
bus_config_parser_get_service_dirs (BusConfigParser *parser)
{
    return &parser->service_dirs;
}

DBusList**
bus_config_parser_get_conf_dirs (BusConfigParser *parser)
{
    return &parser->conf_dirs;
}

dbus_bool_t
bus_config_parser_get_fork (BusConfigParser *parser)
{
    return parser->fork;
}

dbus_bool_t
bus_config_parser_get_syslog (BusConfigParser *parser)
{
    return parser->syslog;
}

dbus_bool_t
bus_config_parser_get_keep_umask (BusConfigParser *parser)
{
    return parser->keep_umask;
}

dbus_bool_t
bus_config_parser_get_allow_anonymous (BusConfigParser *parser)
{
    return parser->allow_anonymous;
}

const char *
bus_config_parser_get_pidfile (BusConfigParser *parser)
{
    return parser->pidfile;
}

const char *
bus_config_parser_get_servicehelper (BusConfigParser *parser)
{
    return parser->servicehelper;
}

BusPolicy*
bus_config_parser_steal_policy (BusConfigParser *parser)
{
    BusPolicy *policy;
}

```

```

    _dbus_assert (parser->policy != NULL); /* can only steal the policy
1 time */

    policy = parser->policy;

    parser->policy = NULL;

    return policy;
}

/* Overwrite any limits that were set in the configuration file */
void
bus_config_parser_get_limits (BusConfigParser *parser,
                             BusLimits      *limits)
{
    *limits = parser->limits;
}

DBusHashTable*
bus_config_parser_steal_service_context_table (BusConfigParser
*parser)
{
    DBusHashTable *table;

    _dbus_assert (parser->service_context_table != NULL); /* can only
steal once */

    table = parser->service_context_table;

    parser->service_context_table = NULL;

    return table;
}

#ifdef DBUS_BUILD_TESTS
#include <stdio.h>

typedef enum
{
    VALID,
    INVALID,
    UNKNOWN
} Validity;

static dbus_bool_t
do_check_own_rules (BusPolicy *policy)
{
    const struct {
        char *name;
        dbus_bool_t allowed;
    } checks[] = {

```

```

{"org.freedesktop", FALSE},
{"org.freedesktop.ManySystem", FALSE},
{"org.freedesktop.ManySystems", TRUE},
{"org.freedesktop.ManySystems.foo", TRUE},
{"org.freedesktop.ManySystems.foo.bar", TRUE},
{"org.freedesktop.ManySystems2", FALSE},
{"org.freedesktop.ManySystems2.foo", FALSE},
{"org.freedesktop.ManySystems2.foo.bar", FALSE},
{NULL, FALSE}
};
int i = 0;

while (checks[i].name)
{
    DBusString service_name;
    dbus_bool_t ret;

    if (!_dbus_string_init (&service_name))
        _dbus_assert_not_reached ("couldn't init string");
    if (!_dbus_string_append (&service_name, checks[i].name))
        _dbus_assert_not_reached ("couldn't append string");

    ret = bus_policy_check_can_own (policy, &service_name);
    printf ("          Check name %s: %s\n", checks[i].name,
           ret ? "allowed" : "not allowed");
    if (checks[i].allowed && !ret)
    {
        _dbus_warn ("Cannot own %s\n", checks[i].name);
        return FALSE;
    }
    if (!checks[i].allowed && ret)
    {
        _dbus_warn ("Can own %s\n", checks[i].name);
        return FALSE;
    }
    _dbus_string_free (&service_name);

    i++;
}

return TRUE;
}

static dbus_bool_t
do_load (const DBusString *full_path,
         Validity          validity,
         dbus_bool_t       oom_possible,
         dbus_bool_t       check_own_rules)
{
    BusConfigParser *parser;
    DBusError error;

```

```

dbus_error_init (&error);

parser = bus_config_load (full_path, TRUE, NULL, &error);
if (parser == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (&error);

    if (oom_possible &&
        dbus_error_has_name (&error, DBUS_ERROR_NO_MEMORY))
    {
        _dbus_verbose ("Failed to load valid file due to OOM\n");
        dbus_error_free (&error);
        return TRUE;
    }
    else if (validity == VALID)
    {
        _dbus_warn ("Failed to load valid file but still had memory:
%s\n",
                    error.message);

        dbus_error_free (&error);
        return FALSE;
    }
    else
    {
        dbus_error_free (&error);
        return TRUE;
    }
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (&error);

    if (check_own_rules && do_check_own_rules (parser->policy) ==
FALSE)
    {
        return FALSE;
    }

    bus_config_parser_unref (parser);

    if (validity == INVALID)
    {
        _dbus_warn ("Accepted invalid file\n");
        return FALSE;
    }

    return TRUE;
}
}

typedef struct

```

```

{
    const DBusString *full_path;
    Validity          validity;
    dbus_bool_t      check_own_rules;
} LoaderOomData;

static dbus_bool_t
check_loader_oom_func (void *data)
{
    LoaderOomData *d = data;

    return do_load (d->full_path, d->validity, TRUE, d-
>check_own_rules);
}

static dbus_bool_t
process_test_valid_subdir (const DBusString *test_base_dir,
                           const char      *subdir,
                           Validity        validity)
{
    DBusString test_directory;
    DBusString filename;
    DBusDirIter *dir;
    dbus_bool_t retval;
    DBusError error;

    retval = FALSE;
    dir = NULL;

    if (!_dbus_string_init (&test_directory))
        _dbus_assert_not_reached ("didn't allocate test_directory\n");

    _dbus_string_init_const (&filename, subdir);

    if (!_dbus_string_copy (test_base_dir, 0,
                           &test_directory, 0))
        _dbus_assert_not_reached ("couldn't copy test_base_dir to
test_directory");

    if (!_dbus_concat_dir_and_file (&test_directory, &filename))
        _dbus_assert_not_reached ("couldn't allocate full path");

    _dbus_string_free (&filename);
    if (!_dbus_string_init (&filename))
        _dbus_assert_not_reached ("didn't allocate filename string\n");

    dbus_error_init (&error);
    dir = _dbus_directory_open (&test_directory, &error);
    if (dir == NULL)
    {
        _dbus_warn ("Could not open %s: %s\n",
                   _dbus_string_get_const_data (&test_directory),

```

```

        error.message);
    dbus_error_free (&error);
    goto failed;
}

if (validity == VALID)
    printf ("Testing valid files:\n");
else if (validity == INVALID)
    printf ("Testing invalid files:\n");
else
    printf ("Testing unknown files:\n");

next:
while (_dbus_directory_get_next_file (dir, &filename, &error))
{
    DBusString full_path;
    LoaderOomData d;

    if (!_dbus_string_init (&full_path))
        _dbus_assert_not_reached ("couldn't init string");

    if (!_dbus_string_copy (&test_directory, 0, &full_path, 0))
        _dbus_assert_not_reached ("couldn't copy dir to full_path");

    if (!_dbus_concat_dir_and_file (&full_path, &filename))
        _dbus_assert_not_reached ("couldn't concat file to dir");

    if (!_dbus_string_ends_with_c_str (&full_path, ".conf"))
    {
        _dbus_verbose ("Skipping non-.conf file %s\n",
            _dbus_string_get_const_data (&filename));
        _dbus_string_free (&full_path);
        goto next;
    }

    printf ("    %s\n", _dbus_string_get_const_data (&filename));

    _dbus_verbose ("    expecting %s\n",
        validity == VALID ? "valid" :
        (validity == INVALID ? "invalid" :
        (validity == UNKNOWN ? "unknown" : "???")));

    d.full_path = &full_path;
    d.validity = validity;
    d.check_own_rules = _dbus_string_ends_with_c_str (&full_path,
        "check-own-rules.conf");

    /* FIXME hackaround for an expat problem, see
     * https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=124747
     * http://freedesktop.org/pipermail/dbus/2004-May/001153.html
     */
}

```

```

        /* if (!_dbus_test_oom_handling ("config-loader",
check_loader_oom_func, &d)) */
        if (!check_loader_oom_func (&d))
            _dbus_assert_not_reached ("test failed");

        _dbus_string_free (&full_path);
    }

    if (dbus_error_is_set (&error))
    {
        _dbus_warn ("Could not get next file in %s: %s\n",
            _dbus_string_get_const_data (&test_directory),
            error.message);
        dbus_error_free (&error);
        goto failed;
    }

    retval = TRUE;

failed:

    if (dir)
        _dbus_directory_close (dir);
    _dbus_string_free (&test_directory);
    _dbus_string_free (&filename);

    return retval;
}

static dbus_bool_t
bools_equal (dbus_bool_t a,
            dbus_bool_t b)
{
    return a ? b : !b;
}

static dbus_bool_t
strings_equal_or_both_null (const char *a,
                           const char *b)
{
    if (a == NULL || b == NULL)
        return a == b;
    else
        return !strcmp (a, b);
}

static dbus_bool_t
elements_equal (const Element *a,
               const Element *b)
{
    if (a->type != b->type)
        return FALSE;
}

```

```

if (!bools_equal (a->had_content, b->had_content))
    return FALSE;

switch (a->type)
{

case ELEMENT_INCLUDE:
    if (!bools_equal (a->d.include.ignore_missing,
                    b->d.include.ignore_missing))
        return FALSE;
    break;

case ELEMENT_POLICY:
    if (a->d.policy.type != b->d.policy.type)
        return FALSE;
    if (a->d.policy.gid_uid_or_at_console != b-
>d.policy.gid_uid_or_at_console)
        return FALSE;
    break;

case ELEMENT_LIMIT:
    if (strcmp (a->d.limit.name, b->d.limit.name))
        return FALSE;
    if (a->d.limit.value != b->d.limit.value)
        return FALSE;
    break;

default:
    /* do nothing */
    break;
}

return TRUE;

}

static dbus_bool_t
lists_of_elements_equal (DBusList *a,
                        DBusList *b)
{
    DBusList *ia;
    DBusList *ib;

    ia = a;
    ib = b;

    while (ia != NULL && ib != NULL)
    {
        if (elements_equal (ia->data, ib->data))
            return FALSE;
        ia = _dbus_list_get_next_link (&a, ia);
    }
}

```



```

        ib = _dbus_list_get_next_link (&b, ib);
    }

    return ia == NULL && ib == NULL;
}

static dbus_bool_t
lists_of_c_strings_equal (DBusList *a,
                          DBusList *b)
{
    DBusList *ia;
    DBusList *ib;

    ia = a;
    ib = b;

    while (ia != NULL && ib != NULL)
    {
        if (strcmp (ia->data, ib->data))
            return FALSE;
        ia = _dbus_list_get_next_link (&a, ia);
        ib = _dbus_list_get_next_link (&b, ib);
    }

    return ia == NULL && ib == NULL;
}

static dbus_bool_t
limits_equal (const BusLimits *a,
             const BusLimits *b)
{
    return
        (a->max_incoming_bytes == b->max_incoming_bytes
         || a->max_incoming_unix_fds == b->max_incoming_unix_fds
         || a->max_outgoing_bytes == b->max_outgoing_bytes
         || a->max_outgoing_unix_fds == b->max_outgoing_unix_fds
         || a->max_message_size == b->max_message_size
         || a->max_message_unix_fds == b->max_message_unix_fds
         || a->activation_timeout == b->activation_timeout
         || a->auth_timeout == b->auth_timeout
         || a->max_completed_connections == b->max_completed_connections
         || a->max_incomplete_connections == b->max_incomplete_connections
         || a->max_connections_per_user == b->max_connections_per_user
         || a->max_pending_activations == b->max_pending_activations
         || a->max_services_per_connection == b-
>max_services_per_connection
         || a->max_match_rules_per_connection == b-
>max_match_rules_per_connection
         || a->max_replies_per_connection == b->max_replies_per_connection
         || a->reply_timeout == b->reply_timeout);
}

```

```

static dbus_bool_t
config_parsers_equal (const BusConfigParser *a,
                      const BusConfigParser *b)
{
    if (!_dbus_string_equal (&a->basedir, &b->basedir))
        return FALSE;

    if (!_lists_of_elements_equal (a->stack, b->stack))
        return FALSE;

    if (!_strings_equal_or_both_null (a->user, b->user))
        return FALSE;

    if (!_lists_of_c_strings_equal (a->listen_on, b->listen_on))
        return FALSE;

    if (!_lists_of_c_strings_equal (a->mechanisms, b->mechanisms))
        return FALSE;

    if (!_lists_of_c_strings_equal (a->service_dirs, b->service_dirs))
        return FALSE;

    /* FIXME: compare policy */

    /* FIXME: compare service selinux ID table */

    if (!limits_equal (&a->limits, &b->limits))
        return FALSE;

    if (!_strings_equal_or_both_null (a->pidfile, b->pidfile))
        return FALSE;

    if (!bools_equal (a->fork, b->fork))
        return FALSE;

    if (!bools_equal (a->keep_umask, b->keep_umask))
        return FALSE;

    if (!bools_equal (a->is_toplevel, b->is_toplevel))
        return FALSE;

    return TRUE;
}

```

```

static dbus_bool_t
all_are_equiv (const DBusString *target_directory)
{
    DBusString filename;
    DBusDirIter *dir;
    BusConfigParser *first_parser;
    BusConfigParser *parser;
    DBusError error;

```

```

dbus_bool_t equal;
dbus_bool_t retval;

dir = NULL;
first_parser = NULL;
parser = NULL;
retval = FALSE;

if (!_dbus_string_init (&filename))
    _dbus_assert_not_reached ("didn't allocate filename string");

dbus_error_init (&error);
dir = _dbus_directory_open (target_directory, &error);
if (dir == NULL)
{
    _dbus_warn ("Could not open %s: %s\n",
                _dbus_string_get_const_data (target_directory),
                error.message);
    dbus_error_free (&error);
    goto finished;
}

printf ("Comparing equivalent files:\n");

next:
while (_dbus_directory_get_next_file (dir, &filename, &error))
{
    DBusString full_path;

    if (!_dbus_string_init (&full_path))
        _dbus_assert_not_reached ("couldn't init string");

    if (!_dbus_string_copy (target_directory, 0, &full_path, 0))
        _dbus_assert_not_reached ("couldn't copy dir to full_path");

    if (!_dbus_concat_dir_and_file (&full_path, &filename))
        _dbus_assert_not_reached ("couldn't concat file to dir");

    if (!_dbus_string_ends_with_c_str (&full_path, ".conf"))
    {
        _dbus_verbose ("Skipping non-.conf file %s\n",
                        _dbus_string_get_const_data (&filename));
        _dbus_string_free (&full_path);
        goto next;
    }

    printf ("    %s\n", _dbus_string_get_const_data (&filename));

    parser = bus_config_load (&full_path, TRUE, NULL, &error);

    if (parser == NULL)
    {

```

```

        _dbus_warn ("Could not load file %s: %s\n",
                    _dbus_string_get_const_data (&full_path),
                    error.message);
        _dbus_string_free (&full_path);
        dbus_error_free (&error);
        goto finished;
    }
    else if (first_parser == NULL)
    {
        _dbus_string_free (&full_path);
        first_parser = parser;
    }
    else
    {
        _dbus_string_free (&full_path);
        equal = config_parsers_equal (first_parser, parser);
        bus_config_parser_unref (parser);
        if (! equal)
            goto finished;
    }
}

retval = TRUE;

finished:
_dbus_string_free (&filename);
if (first_parser)
    bus_config_parser_unref (first_parser);
if (dir)
    _dbus_directory_close (dir);

return retval;
}

static dbus_bool_t
process_test_equiv_subdir (const DBusString *test_base_dir,
                          const char      *subdir)
{
    DBusString test_directory;
    DBusString filename;
    DBusDirIter *dir;
    DBusError error;
    dbus_bool_t equal;
    dbus_bool_t retval;

    dir = NULL;
    retval = FALSE;

    if (!_dbus_string_init (&test_directory))
        _dbus_assert_not_reached ("didn't allocate test_directory");

```

```

_dbus_string_init_const (&filename, subdir);

if (!_dbus_string_copy (test_base_dir, 0,
                       &test_directory, 0))
    _dbus_assert_not_reached ("couldn't copy test_base_dir to
test_directory");

if (!_dbus_concat_dir_and_file (&test_directory, &filename))
    _dbus_assert_not_reached ("couldn't allocate full path");

_dbus_string_free (&filename);
if (!_dbus_string_init (&filename))
    _dbus_assert_not_reached ("didn't allocate filename string");

dbus_error_init (&error);
dir = _dbus_directory_open (&test_directory, &error);
if (dir == NULL)
{
    _dbus_warn ("Could not open %s: %s\n",
               _dbus_string_get_const_data (&test_directory),
               error.message);
    dbus_error_free (&error);
    goto finished;
}

while (_dbus_directory_get_next_file (dir, &filename, &error))
{
    DBusString full_path;

    /* Skip CVS's magic directories! */
    if (_dbus_string_equal_c_str (&filename, "CVS"))
        continue;

    if (!_dbus_string_init (&full_path))
        _dbus_assert_not_reached ("couldn't init string");

    if (!_dbus_string_copy (&test_directory, 0, &full_path, 0))
        _dbus_assert_not_reached ("couldn't copy dir to full_path");

    if (!_dbus_concat_dir_and_file (&full_path, &filename))
        _dbus_assert_not_reached ("couldn't concat file to dir");

    equal = all_are_equiv (&full_path);
    _dbus_string_free (&full_path);

    if (!equal)
        goto finished;
}

retval = TRUE;

finished:

```

```

    _dbus_string_free (&test_directory);
    _dbus_string_free (&filename);
    if (dir)
        _dbus_directory_close (dir);

    return retval;
}

static const char *test_session_service_dir_matches[] =
{
#ifdef DBUS_UNIX
    "/testhome/foo/.testlocal/testshare/dbus-1/services",
    "/testusr/testlocal/testshare/dbus-1/services",
    "/testusr/testshare/dbus-1/services",
    DBUS_DATADIR"/dbus-1/services",
#endif
/* will be filled in test_default_session_servicedirs() */
#ifdef DBUS_WIN
    NULL,
    NULL,
#endif
    NULL
};

static dbus_bool_t
test_default_session_servicedirs (void)
{
    DBusList *dirs;
    DBusList *link;
    DBusString progs;
    int i;

#ifdef DBUS_WIN
    const char *common_progs;
    char buffer[1024];

    if (_dbus_get_install_root(buffer, sizeof(buffer)))
    {
        strcat(buffer, DBUS_DATADIR);
        strcat(buffer, "/dbus-1/services");
        test_session_service_dir_matches[0] = buffer;
    }
#endif

    /* On Unix we don't actually use this variable, but it's easier to
       handle the
       * deallocation if we always allocate it, whether needed or not */
    if (!_dbus_string_init (&progs))
        _dbus_assert_not_reached ("OOM allocating progs");

#ifdef DBUS_UNIX

```

```

common_progs = _dbus_getenv ("CommonProgramFiles");

if (common_progs)
{
    if (!_dbus_string_append (&progs, common_progs))
    {
        _dbus_string_free (&progs);
        return FALSE;
    }

    if (!_dbus_string_append (&progs, "/dbus-1/services"))
    {
        _dbus_string_free (&progs);
        return FALSE;
    }
    test_session_service_dir_matches[1] =
_dbus_string_get_const_data(&progs);
}
#endif
dirs = NULL;

printf ("Testing retrieving the default session service
directories\n");
if (!_dbus_get_standard_session_servicedirs (&dirs))
    _dbus_assert_not_reached ("couldn't get standard dirs");

/* make sure our defaults end with share/dbus-1/service */
while ((link = _dbus_list_pop_first_link (&dirs)))
{
    DBusString path;

    printf ("    default service dir: %s\n", (char *)link->data);
    _dbus_string_init_const (&path, (char *)link->data);
    if (!_dbus_string_ends_with_c_str (&path, "dbus-1/services"))
    {
        printf ("error with default session service directories\n");
        dbus_free (link->data);
        _dbus_list_free_link (link);
        _dbus_string_free (&progs);
        return FALSE;
    }

    dbus_free (link->data);
    _dbus_list_free_link (link);
}

#ifdef DBUS_UNIX
    if (!_dbus_setenv ("XDG_DATA_HOME",
"/testhome/foo/.testlocal/testshare"))
        _dbus_assert_not_reached ("couldn't setenv XDG_DATA_HOME");
#endif

```

```

    if (!_dbus_setenv ("XDG_DATA_DIRS", ":/testusr/testlocal/testshare:
:/testusr/testshare:"))
        _dbus_assert_not_reached ("couldn't setenv XDG_DATA_DIRS");
#endif
    if (!_dbus_get_standard_session_servicedirs (&dirs))
        _dbus_assert_not_reached ("couldn't get standard dirs");

    /* make sure we read and parse the env variable correctly */
    i = 0;
    while ((link = _dbus_list_pop_first_link (&dirs))
        {
            printf ("    test service dir: %s\n", (char *)link->data);
            if (test_session_service_dir_matches[i] == NULL)
                {
                    printf ("more directories parsed than in match set\n");
                    dbus_free (link->data);
                    _dbus_list_free_link (link);
                    _dbus_string_free (&progs);
                    return FALSE;
                }

            if (strcmp (test_session_service_dir_matches[i],
                (char *)link->data) != 0)
                {
                    printf ("%s directory does not match %s in the match set\n",
                        (char *)link->data,
                        test_session_service_dir_matches[i]);
                    dbus_free (link->data);
                    _dbus_list_free_link (link);
                    _dbus_string_free (&progs);
                    return FALSE;
                }

            ++i;

            dbus_free (link->data);
            _dbus_list_free_link (link);
        }

    if (test_session_service_dir_matches[i] != NULL)
        {
            printf ("extra data %s in the match set was not matched\n",
                test_session_service_dir_matches[i]);

            _dbus_string_free (&progs);
            return FALSE;
        }

    _dbus_string_free (&progs);
    return TRUE;
}

```



```

static const char *test_system_service_dir_matches[] =
{
#ifdef DBUS_UNIX
    "/usr/local/share/dbus-1/system-services",
    "/usr/share/dbus-1/system-services",
#endif
    DBUS_DATADIR"/dbus-1/system-services",
#ifdef DBUS_UNIX
    "/lib/dbus-1/system-services",
#endif

#ifdef DBUS_WIN
    NULL,
#endif
    NULL
};

static dbus_bool_t
test_default_system_servicedirs (void)
{
    DBusList *dirs;
    DBusList *link;
    DBusString progs;
#ifdef DBUS_UNIX
    const char *common_progs;
#endif
    int i;

    /* On Unix we don't actually use this variable, but it's easier to
       handle the
       * deallocation if we always allocate it, whether needed or not */
    if (!_dbus_string_init (&progs))
        _dbus_assert_not_reached ("OOM allocating progs");

#ifdef DBUS_UNIX
    common_progs = _dbus_getenv ("CommonProgramFiles");

    if (common_progs)
    {
        if (!_dbus_string_append (&progs, common_progs))
        {
            _dbus_string_free (&progs);
            return FALSE;
        }

        if (!_dbus_string_append (&progs, "/dbus-1/system-services"))
        {
            _dbus_string_free (&progs);
            return FALSE;
        }

        test_system_service_dir_matches[1] =
            _dbus_string_get_const_data(&progs);
    }

```

```

    }
#endif
    dirs = NULL;

    printf ("Testing retrieving the default system service
directories\n");
    if (!_dbus_get_standard_system_servicedirs (&dirs))
        _dbus_assert_not_reached ("couldn't get stardard dirs");

    /* make sure our defaults end with share/dbus-1/system-service */
    while ((link = _dbus_list_pop_first_link (&dirs))
    {
        DBusString path;

        printf ("    default service dir: %s\n", (char *)link->data);
        _dbus_string_init_const (&path, (char *)link->data);
        if (!_dbus_string_ends_with_c_str (&path, "dbus-1/system-
services"))
        {
            printf ("error with default system service directories\n");
            dbus_free (link->data);
            _dbus_list_free_link (link);
            _dbus_string_free (&progs);
            return FALSE;
        }

        dbus_free (link->data);
        _dbus_list_free_link (link);
    }

#ifdef DBUS_UNIX
    if (!_dbus_setenv ("XDG_DATA_HOME",
"/testhome/foo/.testlocal/testshare"))
        _dbus_assert_not_reached ("couldn't setenv XDG_DATA_HOME");

    if (!_dbus_setenv ("XDG_DATA_DIRS", ":/testusr/testlocal/testshare:
:/testusr/testshare:"))
        _dbus_assert_not_reached ("couldn't setenv XDG_DATA_DIRS");
#endif
    if (!_dbus_get_standard_system_servicedirs (&dirs))
        _dbus_assert_not_reached ("couldn't get stardard dirs");

    /* make sure we read and parse the env variable correctly */
    i = 0;
    while ((link = _dbus_list_pop_first_link (&dirs))
    {
        printf ("    test service dir: %s\n", (char *)link->data);
        if (test_system_service_dir_matches[i] == NULL)
        {
            printf ("more directories parsed than in match set\n");
            dbus_free (link->data);
            _dbus_list_free_link (link);
        }
    }
}

```

```

        _dbus_string_free (&progs);
        return FALSE;
    }

    if (strcmp (test_system_service_dir_matches[i],
                (char *)link->data) != 0)
    {
        printf ("%s directory does not match %s in the match set\n",
                (char *)link->data,
                test_system_service_dir_matches[i]);
        dbus_free (link->data);
        _dbus_list_free_link (link);
        _dbus_string_free (&progs);
        return FALSE;
    }

    ++i;

    dbus_free (link->data);
    _dbus_list_free_link (link);
}

if (test_system_service_dir_matches[i] != NULL)
{
    printf ("extra data %s in the match set was not matched\n",
            test_system_service_dir_matches[i]);

    _dbus_string_free (&progs);
    return FALSE;
}

_dbus_string_free (&progs);
return TRUE;
}

dbus_bool_t
bus_config_parser_test (const DBusString *test_data_dir)
{
    if (test_data_dir == NULL ||
        _dbus_string_get_length (test_data_dir) == 0)
    {
        printf ("No test data\n");
        return TRUE;
    }

    if (!test_default_session_servicedirs())
        return FALSE;

#ifdef DBUS_WIN
    printf("default system service dir skipped\n");
#else
    if (!test_default_system_servicedirs())

```

```

        return FALSE;
#endif

    if (!process_test_valid_subdir (test_data_dir, "valid-config-files",
VALID))
        return FALSE;

    if (!process_test_valid_subdir (test_data_dir, "invalid-config-
files", INVALID))
        return FALSE;

    if (!process_test_equiv_subdir (test_data_dir, "equiv-config-
files"))
        return FALSE;

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = config-parser.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* config-parser.h XML-library-agnostic configuration file parser
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

```

```

#ifndef BUS_CONFIG_PARSER_H

```



```

dbus_bool_t      bus_config_parser_finished      (BusConfigParser
*parser,
                                                    DBusError
*error);

/* Functions for extracting the parse results */
const char* bus_config_parser_get_user          (BusConfigParser
*parser);
const char* bus_config_parser_get_type          (BusConfigParser
*parser);
DBusList** bus_config_parser_get_addresses      (BusConfigParser
*parser);
DBusList** bus_config_parser_get_mechanisms     (BusConfigParser
*parser);
dbus_bool_t bus_config_parser_get_fork          (BusConfigParser
*parser);
dbus_bool_t bus_config_parser_get_allow_anonymous (BusConfigParser
*parser);
dbus_bool_t bus_config_parser_get_syslog        (BusConfigParser
*parser);
dbus_bool_t bus_config_parser_get_keep_umask    (BusConfigParser
*parser);
const char* bus_config_parser_get_pidfile       (BusConfigParser
*parser);
const char* bus_config_parser_get_servicehelper (BusConfigParser
*parser);
DBusList** bus_config_parser_get_service_dirs  (BusConfigParser
*parser);
DBusList** bus_config_parser_get_conf_dirs     (BusConfigParser
*parser);
BusPolicy* bus_config_parser_steal_policy       (BusConfigParser
*parser);
void        bus_config_parser_get_limits        (BusConfigParser
*parser,
                                                    BusLimits
*limits);

DBusHashTable* bus_config_parser_steal_service_context_table
(BusConfigParser *parser);

/* Loader functions (backended off one of the XML parsers). Returns a
 * finished ConfigParser.
 */
BusConfigParser* bus_config_load (const DBusString      *file,
                                dbus_bool_t            is_toplevel,
                                const BusConfigParser *parent,
                                DBusError              *error);

#endif /* BUS_CONFIG_PARSER_H */

```

File = config.guess

```
#!/bin/sh
# Attempt to guess a canonical system name.
# Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
# 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
# 2011, 2012 Free Software Foundation, Inc.

timestamp='2012-08-14'

# This file is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, see <http://www.gnu.org/licenses/>.
#
# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program that contains a
# configuration script generated by Autoconf, you may include it under
# the same distribution terms that you use for the rest of that
# program.

# Originally written by Per Bothner. Please send patches (context
# diff format) to <config-patches@gnu.org> and include a ChangeLog
# entry.
#
# This script attempts to guess a canonical system name similar to
# config.sub. If it succeeds, it prints the system name on stdout,
# and
# exits with 0. Otherwise, it exits with 1.
#
# You can get the latest version of this script from:
#
http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config.guess;hb=HEAD

me=`echo "$0" | sed -e 's,./,,'`

usage="\
Usage: $0 [OPTION]

Output the configuration name of the system \"$me\" is run on.

Operation modes:
```

```
-h, --help          print this help, then exit
-t, --time-stamp   print date of last modification, then exit
-v, --version      print version number, then exit
```

Report bugs and patches to <config-patches@gnu.org>."

```
version="\
GNU config.guess ($timestamp)
```

Originally written by Per Bothner.
Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012
Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE."

```
help="
Try \`$me --help' for more information."
```

```
# Parse command line
while test $# -gt 0 ; do
  case $1 in
    --time-stamp | --time* | -t )
      echo "$timestamp" ; exit ;;
    --version | -v )
      echo "$version" ; exit ;;
    --help | --h* | -h )
      echo "$usage"; exit ;;
    -- )          # Stop option processing
      shift; break ;;
    - )          # Use stdin as input.
      break ;;
    -* )
      echo "$me: invalid option $1$help" >&2
      exit 1 ;;
    * )
      break ;;
  esac
done

if test $# != 0; then
  echo "$me: too many arguments$help" >&2
  exit 1
fi
```

```
trap 'exit 1' 1 2 15
```

```
# CC_FOR_BUILD -- compiler used by this script. Note that the use of a
# compiler to aid in system detection is discouraged as it requires
```



```

# temporary files to be created and, as you can see below, it is a
# headache to deal with in a portable fashion.

# Historically, `CC_FOR_BUILD' used to be named `HOST_CC'. We still
# use `HOST_CC' if defined, but it is deprecated.

# Portable tmp directory creation inspired by the Autoconf team.

set_cc_for_build='
trap "exitcode=\$?; (rm -f \$tmpfiles 2>/dev/null; rmdir \$tmp
2>/dev/null) && exit \$exitcode" 0 ;
trap "rm -f \$tmpfiles 2>/dev/null; rmdir \$tmp 2>/dev/null; exit 1" 1
2 13 15 ;
: ${TMPDIR=/tmp} ;
{ tmp=`(umask 077 && mktemp -d "$TMPDIR/cgXXXXXX") 2>/dev/null` &&
test -n "$tmp" && test -d "$tmp" ; } ||
{ test -n "$RANDOM" && tmp=$TMPDIR/cg$$-$RANDOM && (umask 077 &&
mkdir $tmp) ; } ||
{ tmp=$TMPDIR/cg-$$ && (umask 077 && mkdir $tmp) && echo "Warning:
creating insecure temp directory" >&2 ; } ||
{ echo "$me: cannot create a temporary directory in $TMPDIR" >&2 ;
exit 1 ; } ;
dummy=$tmp/dummy ;
tmpfiles="$dummy.c $dummy.o $dummy.rel $dummy" ;
case $CC_FOR_BUILD,$HOST_CC,$CC in
,,)    echo "int x;" > $dummy.c ;
      for c in cc gcc c89 c99 ; do
        if ($c -c -o $dummy.o $dummy.c) >/dev/null 2>&1 ; then
          CC_FOR_BUILD="$c"; break ;
        fi ;
      done ;
      if test x"$CC_FOR_BUILD" = x ; then
        CC_FOR_BUILD=no_compiler_found ;
      fi
      ;;
,,*)   CC_FOR_BUILD=$CC ;;
*,*)   CC_FOR_BUILD=$HOST_CC ;;
esac ; set_cc_for_build= ;'

# This is needed to find uname on a Pyramid OSx when run in the BSD
universe.
# (ghazi@noc.rutgers.edu 1994-08-24)
if (test -f /.attbin/uname) >/dev/null 2>&1 ; then
  PATH=$PATH:/.attbin ; export PATH
fi

UNAME_MACHINE=`(uname -m) 2>/dev/null` || UNAME_MACHINE=unknown
UNAME_RELEASE=`(uname -r) 2>/dev/null` || UNAME_RELEASE=unknown
UNAME_SYSTEM=`(uname -s) 2>/dev/null` || UNAME_SYSTEM=unknown
UNAME_VERSION=`(uname -v) 2>/dev/null` || UNAME_VERSION=unknown

# Detect uclibc systems.

```

```

LIBC="gnu"
if [ -f /usr/include/bits/uClibc_config.h ]
then
    LIBC=uclibc
    if [ -n `grep "#define __UCLIBC_CONFIG_VERSION__"
/usr/include/bits/uClibc_config.h` ]
    then
        UCLIBC_SUBVER=`sed -n "/#define __UCLIBC_CONFIG_VERSION__
/s///p" /usr/include/bits/uClibc_config.h`
        LIBC=$LIBC$UCLIBC_SUBVER
    fi
fi

# Note: order is significant - the case branches are not exclusive.

case
"${UNAME_MACHINE}:${UNAME_SYSTEM}:${UNAME_RELEASE}:${UNAME_VERSION}"
in
*:NetBSD:*:*)
    # NetBSD (nbsd) targets should (where applicable) match one or
    # more of the tuples: *-*-netbsdelf*, *-*-netbsdout*,
    # *-*-netbsdcoeff* and *-*-netbsd*. For targets that recently
    # switched to ELF, *-*-netbsd* would select the old
    # object file format. This provides both forward
    # compatibility and a consistent mechanism for selecting the
    # object file format.
    #
    # Note: NetBSD doesn't particularly care about the vendor
    # portion of the name. We always set it to "unknown".
    sysctl="sysctl -n hw.machine_arch"
    UNAME_MACHINE_ARCH=`(/sbin/$sysctl 2>/dev/null || \
    /usr/sbin/$sysctl 2>/dev/null || echo unknown)`
    case "${UNAME_MACHINE_ARCH}" in
        armeb) machine=armeb-unknown ;;
        arm*) machine=arm-unknown ;;
        sh3el) machine=shl-unknown ;;
        sh3eb) machine=sh-unknown ;;
        sh5el) machine=sh5le-unknown ;;
        *) machine=${UNAME_MACHINE_ARCH}-unknown ;;
    esac
    # The Operating System including object format, if it has
switched
    # to ELF recently, or will in the future.
    case "${UNAME_MACHINE_ARCH}" in
        arm*|i386|m68k|ns32k|sh3*|sparc|vax)
            eval $set_cc_for_build
            if echo __ELF__ | $CC_FOR_BUILD -E - 2>/dev/null \
            | grep -q __ELF__
            then
                # Once all utilities can be ECOFF (netbsdcoeff) or
a.out (netbsdout).

```

```

        # Return netbsd for either.  FIX?
        os=netbsd
    else
        os=netbsdelf
    fi
    ;;
*)
    os=netbsd
    ;;
esac
# The OS release
# Debian GNU/NetBSD machines have a different userland, and
# thus, need a distinct triplet. However, they do not need
# kernel version information, so it can be replaced with a
# suitable tag, in the style of linux-gnu.
case "${UNAME_VERSION}" in
    Debian*)
        release='-gnu'
        ;;
*)
        release=`echo ${UNAME_RELEASE}|sed -e 's/[-_].*/\./'`
        ;;
esac
# Since CPU_TYPE-MANUFACTURER-KERNEL-OPERATING_SYSTEM:
# contains redundant information, the shorter form:
# CPU_TYPE-MANUFACTURER-OPERATING_SYSTEM is used.
echo "${machine}-${os}${release}"
exit ;;
*:Bitrig:*)
    UNAME_MACHINE_ARCH=`arch | sed 's/Bitrig.//'`
    echo ${UNAME_MACHINE_ARCH}-unknown-bitrig${UNAME_RELEASE}
    exit ;;
*:OpenBSD:*)
    UNAME_MACHINE_ARCH=`arch | sed 's/OpenBSD.//'`
    echo ${UNAME_MACHINE_ARCH}-unknown-openbsd${UNAME_RELEASE}
    exit ;;
*:ekkoBSD:*)
    echo ${UNAME_MACHINE}-unknown-ekkobsd${UNAME_RELEASE}
    exit ;;
*:SolidBSD:*)
    echo ${UNAME_MACHINE}-unknown-solidbsd${UNAME_RELEASE}
    exit ;;
macppc:MirBSD:*)
    echo powerpc-unknown-mirbsd${UNAME_RELEASE}
    exit ;;
*:MirBSD:*)
    echo ${UNAME_MACHINE}-unknown-mirbsd${UNAME_RELEASE}
    exit ;;
alpha:OSF1:*)
    case $UNAME_RELEASE in
        *4.0)
            UNAME_RELEASE=`/usr/sbin/sizer -v | awk '{print $3}'`

```

```

        ;;
*5.*)
    UNAME_RELEASE=`/usr/sbin/sizer -v | awk '{print $4}'`
    ;;
esac
# According to Compaq, /usr/sbin/psrinfo has been available on
# OSF/1 and Tru64 systems produced since 1995. I hope that
# covers most systems running today. This code pipes the CPU
# types through head -n 1, so we only detect the type of CPU 0.
ALPHA_CPU_TYPE=`/usr/sbin/psrinfo -v | sed -n -e 's/^ The alpha
\(.*\) processor.*$/\1/p' | head -n 1`
case "$ALPHA_CPU_TYPE" in
    "EV4 (21064)")
        UNAME_MACHINE="alpha" ;;
    "EV4.5 (21064)")
        UNAME_MACHINE="alpha" ;;
    "LCA4 (21066/21068)")
        UNAME_MACHINE="alpha" ;;
    "EV5 (21164)")
        UNAME_MACHINE="alphaev5" ;;
    "EV5.6 (21164A)")
        UNAME_MACHINE="alphaev56" ;;
    "EV5.6 (21164PC)")
        UNAME_MACHINE="alphapca56" ;;
    "EV5.7 (21164PC)")
        UNAME_MACHINE="alphapca57" ;;
    "EV6 (21264)")
        UNAME_MACHINE="alphaev6" ;;
    "EV6.7 (21264A)")
        UNAME_MACHINE="alphaev67" ;;
    "EV6.8CB (21264C)")
        UNAME_MACHINE="alphaev68" ;;
    "EV6.8AL (21264B)")
        UNAME_MACHINE="alphaev68" ;;
    "EV6.8CX (21264D)")
        UNAME_MACHINE="alphaev68" ;;
    "EV6.9A (21264/EV69A)")
        UNAME_MACHINE="alphaev69" ;;
    "EV7 (21364)")
        UNAME_MACHINE="alphaev7" ;;
    "EV7.9 (21364A)")
        UNAME_MACHINE="alphaev79" ;;
esac
# A Pn.n version is a patched version.
# A Vn.n version is a released version.
# A Tn.n version is a released field test version.
# A Xn.n version is an unreleased experimental baselevel.
# 1.2 uses "1.2" for uname -r.
echo ${UNAME_MACHINE}-dec-osf`echo ${UNAME_RELEASE} | sed -e
's/^[PVTX]//' | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxyz'`

```

```

# Reset EXIT trap before exiting to avoid spurious non-zero exit
code.
exitcode=$?
trap '' 0
exit $exitcode ;;
Alpha\ *:Windows_NT*:*)
# How do we know it's Interix rather than the generic POSIX
subsystem?
# Should we change UNAME_MACHINE based on the output of uname
instead
# of the specific Alpha model?
echo alpha-pc-interix
exit ;;
21064:Windows_NT:50:3)
echo alpha-dec-winnt3.5
exit ;;
Amiga*:UNIX_System_V:4.0:*)
echo m68k-unknown-sysv4
exit ;;
*:[Aa]miga[Oo][Ss]:*:*)
echo ${UNAME_MACHINE}-unknown-amigaos
exit ;;
*:[Mm]orph[Oo][Ss]:*:*)
echo ${UNAME_MACHINE}-unknown-morphos
exit ;;
*:OS/390:*:*)
echo i370-ibm-openedition
exit ;;
*:z/VM:*:*)
echo s390-ibm-zvmoe
exit ;;
*:OS400:*:*)
echo powerpc-ibm-os400
exit ;;
arm:RISC*:1.[012]*:*|arm:riscix:1.[012]*:*)
echo arm-acorn-riscix${UNAME_RELEASE}
exit ;;
arm:riscos:*:*|arm:RISCOS:*:*)
echo arm-unknown-riscos
exit ;;
SR2?01:HI-UX/MPP:*:* | SR8000:HI-UX/MPP:*:*)
echo hppa1.1-hitachi-hiuxmpp
exit ;;
Pyramid*:OSx*:*:* | MIS*:OSx*:*:* | MIS*:SMP_DC-OSx*:*:*)
# akee@wpdis03.wpafb.af.mil (Earle F. Ake) contributed MIS and
NILE.
if test "`(/bin/universe) 2>/dev/null`" = att ; then
echo pyramid-pyramid-sysv3
else
echo pyramid-pyramid-bsd
fi
exit ;;

```

```

NILE*:*:*:dcosx)
    echo pyramid-pyramid-svr4
    exit ;;
DRS?6000:unix:4.0:6*)
    echo sparc-icl-nx6
    exit ;;
DRS?6000:UNIX_SV:4.2*:7* | DRS?6000:isis:4.2*:7*)
    case `usr/bin/uname -p` in
        sparc) echo sparc-icl-nx7; exit ;;
    esac ;;
s390x:SunOS:*:*)
    echo ${UNAME_MACHINE}-ibm-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^.]*//`
    exit ;;
sun4H:SunOS:5.*:*)
    echo sparc-hal-solaris2`echo ${UNAME_RELEASE}|sed -e 's/[^.]*//`
    exit ;;
sun4*:SunOS:5.*:* | tadpole*:SunOS:5.*:*)
    echo sparc-sun-solaris2`echo ${UNAME_RELEASE}|sed -e 's/[^.]*//`
    exit ;;
i86pc:AuroraUX:5.*:* | i86xen:AuroraUX:5.*:*)
    echo i386-pc-auroraux${UNAME_RELEASE}
    exit ;;
i86pc:SunOS:5.*:* | i86xen:SunOS:5.*:*)
    eval `set_cc_for_build
    SUN_ARCH="i386"
    # If there is a compiler, see if it is configured for 64-bit
objects.
    # Note that the Sun cc does not turn __LP64__ into 1 like gcc
does.
    # This test works for both compilers.
    if [ "$CC_FOR_BUILD" != 'no_compiler_found' ]; then
        if (echo '#ifdef __amd64'; echo IS_64BIT_ARCH; echo '#endif')
| \
            (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) | \
            grep IS_64BIT_ARCH >/dev/null
        then
            SUN_ARCH="x86_64"
        fi
    fi
    echo ${SUN_ARCH}-pc-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^.]*//`
    exit ;;
sun4*:SunOS:6.*:*)
    # According to config.sub, this is the proper way to canonicalize
    # SunOS6. Hard to guess exactly what SunOS6 will be like, but
    # it's likely to be more like Solaris than SunOS4.
    echo sparc-sun-solaris3`echo ${UNAME_RELEASE}|sed -e 's/[^.]*//`
    exit ;;
sun4*:SunOS:*:*)
    case "`usr/bin/arch -k`" in
        Series*|S4*)

```

```

        UNAME_RELEASE=`uname -v`
        ;;
    esac
    # Japanese Language versions have a version number like `4.1.3-
JL'.
    echo sparc-sun-sunos`echo ${UNAME_RELEASE}|sed -e 's/-/_/'`
    exit ;;
sun3*:SunOS:*)
    echo m68k-sun-sunos${UNAME_RELEASE}
    exit ;;
sun*:*:4.2BSD:*)
    UNAME_RELEASE=`(sed lq /etc/motd | awk '{print substr($5,1,3)}')
2>/dev/null`
    test "x${UNAME_RELEASE}" = "x" && UNAME_RELEASE=3
    case "`bin/arch`" in
        sun3)
            echo m68k-sun-sunos${UNAME_RELEASE}
            ;;
        sun4)
            echo sparc-sun-sunos${UNAME_RELEASE}
            ;;
    esac
    exit ;;
aushp:SunOS:*)
    echo sparc-auspex-sunos${UNAME_RELEASE}
    exit ;;
# The situation for MiNT is a little confusing.  The machine name
# can be virtually everything (everything which is not
# "atarist" or "atariste" at least should have a processor
# > m68000).  The system name ranges from "MiNT" over "FreeMiNT"
# to the lowercase version "mint" (or "freemint").  Finally
# the system name "TOS" denotes a system which is actually not
# MiNT.  But MiNT is downward compatible to TOS, so this should
# be no problem.
atarist[e]:*MiNT:*:* | atarist[e]:*mint:*:* | atarist[e]:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
atari*:*MiNT:*:* | atari*:*mint:*:* | atarist[e]:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
*falcon*:*MiNT:*:* | *falcon*:*mint:*:* | *falcon*:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
milan*:*MiNT:*:* | milan*:*mint:*:* | *milan*:*TOS:*:*)
    echo m68k-milan-mint${UNAME_RELEASE}
    exit ;;
hades*:*MiNT:*:* | hades*:*mint:*:* | *hades*:*TOS:*:*)
    echo m68k-hades-mint${UNAME_RELEASE}
    exit ;;
*:*MiNT:*:* | *:*mint:*:* | *:*TOS:*:*)
    echo m68k-unknown-mint${UNAME_RELEASE}
    exit ;;

```

```

m68k:machten:*:*)
    echo m68k-apple-machten${UNAME_RELEASE}
    exit ;;
powerpc:machten:*:*)
    echo powerpc-apple-machten${UNAME_RELEASE}
    exit ;;
RISC*:Mach:*:*)
    echo mips-dec-mach_bsd4.3
    exit ;;
RISC*:ULTRIX:*:*)
    echo mips-dec-ultrix${UNAME_RELEASE}
    exit ;;
VAX*:ULTRIX*:*:*)
    echo vax-dec-ultrix${UNAME_RELEASE}
    exit ;;
2020:CLIX:*:* | 2430:CLIX:*:*)
    echo clipper-intergraph-clix${UNAME_RELEASE}
    exit ;;
mips:*:*:UMIPS | mips:*:*:RISCos)
    eval $set_cc_for_build
    sed 's/^ // ' << EOF >$dummy.c
#ifdef __cplusplus
#include <stdio.h> /* for printf() prototype */
    int main (int argc, char *argv[]) {
#else
    int main (argc, argv) int argc; char *argv[]; {
#endif
    #if defined (host_mips) && defined (MIPSEB)
    #if defined (SYSTYPE_SYSV)
        printf ("mips-mips-riscos%ssysv\n", argv[1]); exit (0);
    #endif
    #if defined (SYSTYPE_SVR4)
        printf ("mips-mips-riscos%ssvr4\n", argv[1]); exit (0);
    #endif
    #if defined (SYSTYPE_BSD43) || defined(SYSTYPE_BSD)
        printf ("mips-mips-riscos%bsbsd\n", argv[1]); exit (0);
    #endif
    #endif
        exit (-1);
    }
EOF
    $CC_FOR_BUILD -o $dummy $dummy.c &&
    dummyarg=`echo "${UNAME_RELEASE}" | sed -n 's/\([0-9]*\).*\/\1/p'` &&
    SYSTEM_NAME=`$dummy $dummyarg` &&
    { echo "$SYSTEM_NAME"; exit; }
    echo mips-mips-riscos${UNAME_RELEASE}
    exit ;;
Motorola:PowerMAX_OS:*:*)
    echo powerpc-motorola-powermax
    exit ;;
Motorola:*:4.3:PL8-*)

```



```

    echo powerpc-harris-powermax
    exit ;;
Night_Hawk:*:*:PowerMAX_OS | Synergy:PowerMAX_OS:*:*)
    echo powerpc-harris-powermax
    exit ;;
Night_Hawk:Power_UNIX:*:*)
    echo powerpc-harris-powerunix
    exit ;;
m88k:CX/UX:7:*:*)
    echo m88k-harris-cxux7
    exit ;;
m88k:*:4*:R4*)
    echo m88k-motorola-sysv4
    exit ;;
m88k:*:3*:R3*)
    echo m88k-motorola-sysv3
    exit ;;
AViiON:dgux:*:*)
    # DG/UX returns AViiON for all architectures
    UNAME_PROCESSOR=`/usr/bin/uname -p`
    if [ $UNAME_PROCESSOR = mc88100 ] || [ $UNAME_PROCESSOR = mc88110
]
    then
        if [ ${TARGET_BINARY_INTERFACE}x = m88kdguxelfx ] || \
            [ ${TARGET_BINARY_INTERFACE}x = x ]
        then
            echo m88k-dg-dgux${UNAME_RELEASE}
        else
            echo m88k-dg-dguxbcs${UNAME_RELEASE}
        fi
    else
        echo i586-dg-dgux${UNAME_RELEASE}
    fi
    exit ;;
M88*:DolphinOS:*:*) # DolphinOS (SVR3)
    echo m88k-dolphin-sysv3
    exit ;;
M88*:*:R3*:*:*)
    # Delta 88k system running SVR3
    echo m88k-motorola-sysv3
    exit ;;
XD88*:*:*:*:*) # Tektronix XD88 system running UTekV (SVR3)
    echo m88k-tektronix-sysv3
    exit ;;
Tek43[0-9][0-9]:UTek:*:*) # Tektronix 4300 system running UTek
(BSD)
    echo m68k-tektronix-bsd
    exit ;;
*:IRIX*:*:*:*)
    echo mips-sgi-irix`echo ${UNAME_RELEASE}|sed -e 's/-/_/g'`
    exit ;;
?????????:AIX?:[12].1:2) # AIX 2.2.1 or AIX 2.1.1 is RT/PC AIX.

```

```

    echo romp-ibm-aix      # uname -m gives an 8 hex-code CPU id
    exit ;;               # Note that: echo "`uname -s`" gives 'AIX
,
i*86:AIX:*:*)
    echo i386-ibm-aix
    exit ;;
ia64:AIX:*:*)
    if [ -x /usr/bin/oslevel ] ; then
        IBM_REV=`/usr/bin/oslevel`
    else
        IBM_REV=${UNAME_VERSION}.${UNAME_RELEASE}
    fi
    echo ${UNAME_MACHINE}-ibm-aix${IBM_REV}
    exit ;;
*:AIX:2:3)
    if grep bos325 /usr/include/stdio.h >/dev/null 2>&1; then
        eval $set_cc_for_build
        sed 's/^        //' << EOF >$dummy.c
        #include <sys/systemcfg.h>

        main()
        {
            if (!__power_pc())
                exit(1);
            puts("powerpc-ibm-aix3.2.5");
            exit(0);
        }
EOF
        if $CC_FOR_BUILD -o $dummy $dummy.c && SYSTEM_NAME=`$dummy`
        then
            echo "$SYSTEM_NAME"
        else
            echo rs6000-ibm-aix3.2.5
        fi
    elif grep bos324 /usr/include/stdio.h >/dev/null 2>&1; then
        echo rs6000-ibm-aix3.2.4
    else
        echo rs6000-ibm-aix3.2
    fi
    exit ;;
*:AIX:*: [4567])
    IBM_CPU_ID=`/usr/sbin/lsdev -C -c processor -S available | sed 1q
| awk '{ print $1 }'`
    if /usr/sbin/lsattr -El ${IBM_CPU_ID} | grep ' POWER' >/dev/null
2>&1; then
        IBM_ARCH=rs6000
    else
        IBM_ARCH=powerpc
    fi
    if [ -x /usr/bin/oslevel ] ; then
        IBM_REV=`/usr/bin/oslevel`
    else

```

```

        IBM_REV=${UNAME_VERSION}.${UNAME_RELEASE}
    fi
    echo ${IBM_ARCH}-ibm-aix${IBM_REV}
    exit ;;
*:AIX:*:*)
    echo rs6000-ibm-aix
    exit ;;
ibmrt:4.4BSD:*|romp-ibm:BSD:*)
    echo romp-ibm-bsd4.4
    exit ;;
ibmrt:*BSD:*|romp-ibm:BSD:*)
    echo romp-ibm-bsd${UNAME_RELEASE} # covers RT/PC BSD and
                                        # 4.3 with uname added to
                                        # report: romp-ibm BSD 4.3
*:BOSX:*:*)
    echo rs6000-bull-bosx
    exit ;;
DPX/2?00:B.O.S.:*:*:*)
    echo m68k-bull-sysv3
    exit ;;
9000/[34]??:4.3bsd:1.*:*)
    echo m68k-hp-bsd
    exit ;;
hp300:4.4BSD:*:* | 9000/[34]??:4.3bsd:2.*:*)
    echo m68k-hp-bsd4.4
    exit ;;
9000/[34678]?:HP-UX:*:*:*)
    HPUX_REV=`echo ${UNAME_RELEASE}|sed -e 's/[^.]*. [0B]*//'\`
    case "${UNAME_MACHINE}" in
        9000/31? )           HP_ARCH=m68000 ;;
        9000/[34]?? )      HP_ARCH=m68k ;;
        9000/[678][0-9][0-9])
            if [ -x /usr/bin/getconf ]; then
                sc_cpu_version=`/usr/bin/getconf SC_CPU_VERSION
2>/dev/null`
                sc_kernel_bits=`/usr/bin/getconf SC_KERNEL_BITS
2>/dev/null`
                case "${sc_cpu_version}" in
                    523) HP_ARCH="hppa1.0" ;; # CPU_PA_RISC1_0
                    528) HP_ARCH="hppa1.1" ;; # CPU_PA_RISC1_1
                    532)           # CPU_PA_RISC2_0
                case "${sc_kernel_bits}" in
                    32) HP_ARCH="hppa2.0n" ;;
                    64) HP_ARCH="hppa2.0w" ;;
                    ') HP_ARCH="hppa2.0" ;; # HP-UX 10.20
                esac ;;
            esac
    esac
    fi
    if [ "${HP_ARCH}" = "" ]; then
        eval `set_cc_for_build
        sed 's/^ // ' << EOF >$dummy.c

#define _HPUX_SOURCE

```

```

#include <stdlib.h>
#include <unistd.h>

int main ()
{
#ifdef _SC_KERNEL_BITS
    long bits = sysconf(_SC_KERNEL_BITS);
#endif
    long cpu = sysconf (_SC_CPU_VERSION);

    switch (cpu)
    {
        case CPU_PA_RISC1_0: puts ("hppa1.0"); break;
        case CPU_PA_RISC1_1: puts ("hppa1.1"); break;
        case CPU_PA_RISC2_0:
#ifdef _SC_KERNEL_BITS
            switch (bits)
            {
                case 64: puts ("hppa2.0w"); break;
                case 32: puts ("hppa2.0n"); break;
                default: puts ("hppa2.0"); break;
            } break;
        #else /* !defined(_SC_KERNEL_BITS) */
            puts ("hppa2.0"); break;
        #endif
        default: puts ("hppa1.0"); break;
    }
    exit (0);
}

EOF
        (CCOPTS= $CC_FOR_BUILD -o $dummy $dummy.c 2>/dev/null)
&& HP_ARCH=`$dummy`
        test -z "$HP_ARCH" && HP_ARCH=hppa
    fi ;;
esac
if [ ${HP_ARCH} = "hppa2.0w" ]
then
    eval $set_cc_for_build

    # hppa2.0w-hp-hpux* has a 64-bit kernel and a compiler
generating
    # 32-bit code. hppa64-hp-hpux* has the same kernel and a
compiler
    # generating 64-bit code. GNU and HP use different
nomenclature:
    #
    # $ CC_FOR_BUILD=cc ./config.guess
    # => hppa2.0w-hp-hpux11.23
    # $ CC_FOR_BUILD="cc +DA2.0w" ./config.guess
    # => hppa64-hp-hpux11.23

    if echo __LP64__ | (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) |

```

```

        grep -q __LP64__
    then
        HP_ARCH="hppa2.0w"
    else
        HP_ARCH="hppa64"
    fi
fi
echo ${HP_ARCH}-hp-hpux${HPUX_REV}
exit ;;
ia64:HP-UX:*:*)
    HPUX_REV=`echo ${UNAME_RELEASE}|sed -e 's/[^.]*.[0B]*//'\`
    echo ia64-hp-hpux${HPUX_REV}
    exit ;;
3050*:HI-UX:*:*)
    eval $set_cc_for_build
    sed 's/^ // ' << EOF >$dummy.c
    #include <unistd.h>
    int
    main ()
    {
        long cpu = sysconf (_SC_CPU_VERSION);
        /* The order matters, because CPU_IS_HP_MC68K erroneously
returns
        true for CPU_PA_RISC1_0. CPU_IS_PA_RISC returns correct
        results, however. */
        if (CPU_IS_PA_RISC (cpu))
        {
            switch (cpu)
            {
                case CPU_PA_RISC1_0: puts ("hppa1.0-hitachi-hiuxwe2");
break;
                case CPU_PA_RISC1_1: puts ("hppa1.1-hitachi-hiuxwe2");
break;
                case CPU_PA_RISC2_0: puts ("hppa2.0-hitachi-hiuxwe2");
break;
                default: puts ("hppa-hitachi-hiuxwe2"); break;
            }
        }
        else if (CPU_IS_HP_MC68K (cpu))
            puts ("m68k-hitachi-hiuxwe2");
        else puts ("unknown-hitachi-hiuxwe2");
        exit (0);
    }
EOF
    $CC_FOR_BUILD -o $dummy $dummy.c && SYSTEM_NAME=`$dummy` &&
        { echo "$SYSTEM_NAME"; exit; }
    echo unknown-hitachi-hiuxwe2
    exit ;;
9000/7???:4.3bsd:*:* | 9000/8?[79]:4.3bsd:*:* )
    echo hppa1.1-hp-bsd
    exit ;;
9000/8???:4.3bsd:*:*)

```

```

    echo hppa1.0-hp-bsd
    exit ;;
*9???:MPE/iX:*:* | *3000*:MPE/iX:*:*)
    echo hppa1.0-hp-mpeix
    exit ;;
hp7?:OSF1:*:* | hp8?[79]:OSF1:*:* )
    echo hppa1.1-hp-osf
    exit ;;
hp8?:OSF1:*:*)
    echo hppa1.0-hp-osf
    exit ;;
i*86:OSF1:*:*)
    if [ -x /usr/sbin/sysversion ] ; then
        echo ${UNAME_MACHINE}-unknown-osf1mk
    else
        echo ${UNAME_MACHINE}-unknown-osf1
    fi
    exit ;;
parisc*:Lites*:*:*)
    echo hppa1.1-hp-lites
    exit ;;
C1*:ConvexOS:*:* | convex:ConvexOS:C1*:* )
    echo c1-convex-bsd
    exit ;;
C2*:ConvexOS:*:* | convex:ConvexOS:C2*:* )
    if getsysinfo -f scalar_acc
    then echo c32-convex-bsd
    else echo c2-convex-bsd
    fi
    exit ;;
C34*:ConvexOS:*:* | convex:ConvexOS:C34*:* )
    echo c34-convex-bsd
    exit ;;
C38*:ConvexOS:*:* | convex:ConvexOS:C38*:* )
    echo c38-convex-bsd
    exit ;;
C4*:ConvexOS:*:* | convex:ConvexOS:C4*:* )
    echo c4-convex-bsd
    exit ;;
CRAY*Y-MP:*:*:*)
    echo ymp-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
    exit ;;
CRAY*[A-Z]90:*:*:*)
    echo ${UNAME_MACHINE}-cray-unicos${UNAME_RELEASE} \
    | sed -e 's/CRAY.*\([A-Z]90\)/\1/' \
        -e y/ABCDEFGHIJKLMNPOQRSTUVWXYZ/abcdefghijklmnopqrstuvwxy/
    -e 's/\.[^.]*/.X/'
    exit ;;
CRAY*TS:*:*:*)
    echo t90-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
    exit ;;

```

```

    CRAY*T3E:*:*:*)
        echo alphaev5-cray-unicosmk${UNAME_RELEASE} | sed -e
's/\.[^.]*/.X/'
        exit ;;
    CRAY*SV1:*:*:*)
        echo svl-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
        exit ;;
    *:UNICOS/mp:*:*:*)
        echo craynv-cray-unicosmp${UNAME_RELEASE} | sed -e
's/\.[^.]*/.X/'
        exit ;;
    F30[01]:UNIX_System_V:*:* | F700:UNIX_System_V:*:*:*)
        FUJITSU_PROC=`uname -m | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxy'`
        FUJITSU_SYS=`uname -p | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxy' | sed -e 's/\/\///'`
        FUJITSU_REL=`echo ${UNAME_RELEASE} | sed -e 's/\/\///'`
        echo "${FUJITSU_PROC}-fujitsu-${FUJITSU_SYS}${FUJITSU_REL}"
        exit ;;
    5000:UNIX_System_V:4.*:*:*)
        FUJITSU_SYS=`uname -p | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxy' | sed -e 's/\/\///'`
        FUJITSU_REL=`echo ${UNAME_RELEASE} | tr
'ABCDEFGHIJKLMNOPQRSTUVWXYZ' 'abcdefghijklmnopqrstuvwxy' | sed -e 's/
\/\///'`
        echo "sparc-fujitsu-${FUJITSU_SYS}${FUJITSU_REL}"
        exit ;;
    i*86:BSD/386:*:* | i*86:BSD/OS:*:* | *:Ascend\ Embedded/OS:*:*:*)
        echo ${UNAME_MACHINE}-pc-bsdi${UNAME_RELEASE}
        exit ;;
    sparc*:BSD/OS:*:*:*)
        echo sparc-unknown-bsdi${UNAME_RELEASE}
        exit ;;
    *:BSD/OS:*:*:*)
        echo ${UNAME_MACHINE}-unknown-bsdi${UNAME_RELEASE}
        exit ;;
    *:FreeBSD:*:*:*)
        UNAME_PROCESSOR=`/usr/bin/uname -p`
        case ${UNAME_PROCESSOR} in
            amd64)
                echo x86_64-unknown-freebsd`echo ${UNAME_RELEASE}|sed -e
's/[-(].*//'` ;;
            *)
                echo ${UNAME_PROCESSOR}-unknown-freebsd`echo
${UNAME_RELEASE}|sed -e 's/[-(].*//'` ;;
        esac
        exit ;;
    i*:CYGWIN:*:*:*)
        echo ${UNAME_MACHINE}-pc-cygwin
        exit ;;
    *:MINGW64:*:*:*)
        echo ${UNAME_MACHINE}-pc-mingw64

```

```

    exit ;;
*:MINGW*:*)
    echo ${UNAME_MACHINE}-pc-mingw32
    exit ;;
i*:MSYS*:*)
    echo ${UNAME_MACHINE}-pc-msys
    exit ;;
i*:windows32*:*)
    # uname -m includes "-pc" on this system.
    echo ${UNAME_MACHINE}-mingw32
    exit ;;
i*:PW*:*)
    echo ${UNAME_MACHINE}-pc-pw32
    exit ;;
*:Interix*:*)
    case ${UNAME_MACHINE} in
        x86)
            echo i586-pc-interix${UNAME_RELEASE}
            exit ;;
        authenticamd | genuineintel | EM64T)
            echo x86_64-unknown-interix${UNAME_RELEASE}
            exit ;;
        IA64)
            echo ia64-unknown-interix${UNAME_RELEASE}
            exit ;;
    esac ;;
[345]86:Windows_95:* | [345]86:Windows_98:* |
[345]86:Windows_NT:*)
    echo i${UNAME_MACHINE}-pc-mks
    exit ;;
8664:Windows_NT:*)
    echo x86_64-pc-mks
    exit ;;
i*:Windows_NT*:*) | Pentium*:Windows_NT*:*)
    # How do we know it's Interix rather than the generic POSIX
subsystem?
    # It also conflicts with pre-2.0 versions of AT&T UWIN. Should we
    # UNAME_MACHINE based on the output of uname instead of i386?
    echo i586-pc-interix
    exit ;;
i*:UWIN*:*)
    echo ${UNAME_MACHINE}-pc-uwin
    exit ;;
amd64:CYGWIN*:*) | x86_64:CYGWIN*:*)
    echo x86_64-unknown-cygwin
    exit ;;
p*:CYGWIN*:*)
    echo powerpcle-unknown-cygwin
    exit ;;
prep*:SunOS:5.*:*)
    echo powerpcle-unknown-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^\.]*//`

```



```

    exit ;;
*:GNU:*:*)
    # the GNU system
    echo `echo ${UNAME_MACHINE}|sed -e 's,[-/].*$,,`-unknown-
gnu`echo ${UNAME_RELEASE}|sed -e 's,/.*$,,`
    exit ;;
*:GNU/*:*:*)
    # other systems with GNU libc and userland
    echo ${UNAME_MACHINE}-unknown-`echo ${UNAME_SYSTEM} | sed
's,^[^/]*/,,' | tr '[A-Z]' '[a-z]` `echo ${UNAME_RELEASE}|sed -e 's/[-
(].*//`-gnu
    exit ;;
i*86:Minix:*:*)
    echo ${UNAME_MACHINE}-pc-minix
    exit ;;
aarch64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-gnu
    exit ;;
aarch64_be:Linux:*:*)
    UNAME_MACHINE=aarch64_be
    echo ${UNAME_MACHINE}-unknown-linux-gnu
    exit ;;
alpha:Linux:*:*)
    case `sed -n '/^cpu model/s/^.*: \(.*)/\1/p' < /proc/cpuinfo` in
        EV5)    UNAME_MACHINE=alphaev5 ;;
        EV56)   UNAME_MACHINE=alphaev56 ;;
        PCA56)  UNAME_MACHINE=alphapca56 ;;
        PCA57)  UNAME_MACHINE=alphapca56 ;;
        EV6)    UNAME_MACHINE=alphaev6 ;;
        EV67)   UNAME_MACHINE=alphaev67 ;;
        EV68*)  UNAME_MACHINE=alphaev68 ;;
    esac
    objdump --private-headers /bin/sh | grep -q ld.so.1
    if test "$?" = 0 ; then LIBC="gnulibc1" ; else LIBC="" ; fi
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
arm*:Linux:*:*)
    eval `set_cc_for_build
    if echo __ARM_EABI__ | $CC_FOR_BUILD -E - 2>/dev/null \
        | grep -q __ARM_EABI__
    then
        echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    else
        if echo __ARM_PCS_VFP | $CC_FOR_BUILD -E - 2>/dev/null \
            | grep -q __ARM_PCS_VFP
        then
            echo ${UNAME_MACHINE}-unknown-linux-gnueabi
        else
            echo ${UNAME_MACHINE}-unknown-linux-gnueabihf
        fi
    fi
    fi
    exit ;;

```

```

avr32*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
cris:Linux:*:*)
    echo ${UNAME_MACHINE}-axis-linux-${LIBC}
    exit ;;
crisv32:Linux:*:*)
    echo ${UNAME_MACHINE}-axis-linux-${LIBC}
    exit ;;
frv:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
hexagon:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
i*86:Linux:*:*)
    LIBC=gnu
    eval `set_cc_for_build`
    sed 's/^_ //' << EOF >$dummy.c
    #ifdef __dietlibc__
    LIBC=dietlibc
    #endif
EOF
    eval `set_cc_for_build` -E $dummy.c 2>/dev/null | grep '^LIBC'
    echo "${UNAME_MACHINE}-pc-linux-${LIBC}"
    exit ;;
ia64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
m32r*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
m68*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
mips:Linux:*:* | mips64:Linux:*:*)
    eval `set_cc_for_build`
    sed 's/^_ //' << EOF >$dummy.c
    #undef CPU
    #undef ${UNAME_MACHINE}
    #undef ${UNAME_MACHINE}el
    #if defined(__MIPSEL__) || defined(__MIPSEL) || defined(_MIPSEL)
|| defined(MIPSEL)
        CPU=${UNAME_MACHINE}el
    #else
    #if defined(__MIPSEB__) || defined(__MIPSEB) || defined(_MIPSEB)
|| defined(MIPSEB)
        CPU=${UNAME_MACHINE}
    #else
        CPU=
    #endif
    #endif
    #endif

```

```

EOF
    eval `\$CC_FOR_BUILD -E $dummy.c 2>/dev/null | grep '^CPU'`
    test x"${CPU}" != x && { echo "${CPU}-unknown-linux-${LIBC}";
exit; }
;;
or32:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
padre:Linux:*:*)
    echo sparc-unknown-linux-${LIBC}
    exit ;;
parisc64:Linux:*:* | hppa64:Linux:*:*)
    echo hppa64-unknown-linux-${LIBC}
    exit ;;
parisc:Linux:*:* | hppa:Linux:*:*)
    # Look for CPU level
    case `grep '^cpu[^a-z]*:' /proc/cpuinfo 2>/dev/null | cut -d' ' -
f2` in
        PA7*) echo hppa1.1-unknown-linux-${LIBC} ;;
        PA8*) echo hppa2.0-unknown-linux-${LIBC} ;;
        *)    echo hppa-unknown-linux-${LIBC} ;;
    esac
    exit ;;
ppc64:Linux:*:*)
    echo powerpc64-unknown-linux-${LIBC}
    exit ;;
ppc:Linux:*:*)
    echo powerpc-unknown-linux-${LIBC}
    exit ;;
s390:Linux:*:* | s390x:Linux:*:*)
    echo ${UNAME_MACHINE}-ibm-linux
    exit ;;
sh64*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
sh*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
sparc:Linux:*:* | sparc64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
tile*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
vax:Linux:*:*)
    echo ${UNAME_MACHINE}-dec-linux-${LIBC}
    exit ;;
x86_64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
xtensa*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}

```

```

exit ;;
i*86:DYNIX/ptx:4*:**)
# ptx 4.0 does uname -s correctly, with DYNIX/ptx in there.
# earlier versions are messed up and put the nodename in both
# sysname and nodename.
echo i386-sequent-sysv4
exit ;;
i*86:UNIX_SV:4.2MP:2.*)
# Unixware is an offshoot of SVR4, but it has its own version
# number series starting with 2...
# I am not positive that other SVR4 systems won't match this,
# I just have to hope. -- rms.
# Use sysv4.2uw... so that sysv4* matches it.
echo ${UNAME_MACHINE}-pc-sysv4.2uw${UNAME_VERSION}
exit ;;
i*86:OS/2*:**)
# If we were able to find `uname', then EMX Unix compatibility
# is probably installed.
echo ${UNAME_MACHINE}-pc-os2-emx
exit ;;
i*86:XTS-300*:*:STOP)
echo ${UNAME_MACHINE}-unknown-stop
exit ;;
i*86:atheos*:*:*)
echo ${UNAME_MACHINE}-unknown-atheos
exit ;;
i*86:syllable*:*:*)
echo ${UNAME_MACHINE}-pc-syllable
exit ;;
i*86:LynxOS:2.*:* | i*86:LynxOS:3.[01]*:* | i*86:LynxOS:4.[02]*:**)
echo i386-unknown-lynxos${UNAME_RELEASE}
exit ;;
i*86:*DOS*:*:*)
echo ${UNAME_MACHINE}-pc-msdosdjgpp
exit ;;
i*86:*:4.*:* | i*86:SYSTEM_V:4.*:**)
UNAME_REL=`echo ${UNAME_RELEASE} | sed 's/\\MP$//'\`
if grep Novell /usr/include/link.h >/dev/null 2>/dev/null; then
    echo ${UNAME_MACHINE}-univel-sysv${UNAME_REL}
else
    echo ${UNAME_MACHINE}-pc-sysv${UNAME_REL}
fi
exit ;;
i*86:*:5:[678]*)
# UnixWare 7.x, OpenUNIX and OpenServer 6.
case ` /bin/uname -X | grep "^Machine" ` in
    *486*)          UNAME_MACHINE=i486 ;;
    *Pentium)      UNAME_MACHINE=i586 ;;
    *Pent*|*Celeron) UNAME_MACHINE=i686 ;;
esac
echo ${UNAME_MACHINE}-unknown-
sysv${UNAME_RELEASE}${UNAME_SYSTEM}${UNAME_VERSION}

```

```

    exit ;;
i*86*:3.2:*)
    if test -f /usr/options/cb.name; then
        UNAME_REL=`sed -n 's/.*Version //p' </usr/options/cb.name`
        echo ${UNAME_MACHINE}-pc-isc$UNAME_REL
    elif /bin/uname -X 2>/dev/null >/dev/null ; then
        UNAME_REL=`(/bin/uname -X|grep Release|sed -e 's/.*= //')`
        (/bin/uname -X|grep i80486 >/dev/null) &&
UNAME_MACHINE=i486
        (/bin/uname -X|grep '^Machine.*Pentium' >/dev/null) \
            && UNAME_MACHINE=i586
        (/bin/uname -X|grep '^Machine.*Pent *II' >/dev/null) \
            && UNAME_MACHINE=i686
        (/bin/uname -X|grep '^Machine.*Pentium Pro' >/dev/null) \
            && UNAME_MACHINE=i686
        echo ${UNAME_MACHINE}-pc-sco$UNAME_REL
    else
        echo ${UNAME_MACHINE}-pc-sysv32
    fi
    exit ;;
pc*:3.2:*)
    # Left here for compatibility:
    # uname -m prints for DJGPP always 'pc', but it prints nothing
about
    # the processor, so we play safe by assuming i586.
    # Note: whatever this is, it MUST be the same as what config.sub
    # prints for the "djgpp" host, or else GDB configury will decide
that
    # this is a cross-build.
    echo i586-pc-msdosdjgpp
    exit ;;
Intel:Mach:3*:*)
    echo i386-pc-mach3
    exit ;;
paragon*:3.2:*)
    echo i860-intel-osf1
    exit ;;
i860*:4.2:*) # i860-SVR4
    if grep Stardent /usr/include/sys/uadmin.h >/dev/null 2>&1 ; then
SVR4
        echo i860-stardent-sysv${UNAME_RELEASE} # Stardent Vistra i860-
    else # Add other i860-SVR4 vendors below as they are discovered.
        echo i860-unknown-sysv${UNAME_RELEASE} # Unknown i860-SVR4
    fi
    exit ;;
mini*:CTIX:SYS*5:*)
    # "miniframe"
    echo m68010-convergent-sysv
    exit ;;
mc68k:UNIX:SYSTEM5:3.51m)
    echo m68k-convergent-sysv
    exit ;;

```

```

M680?0:D-NIX:5.3:*)
    echo m68k-diab-dnix
    exit ;;
M68*:*:R3V[5678]*:*)
    test -r /sysV68 && { echo 'm68k-motorola-sysv'; exit; } ;;
3[345]??:*:4.0:3.0 | 3[34]??A:*:4.0:3.0 | 3[34]??,*:*:4.0:3.0 |
3[34]??/*:*:4.0:3.0 | 4400:*:4.0:3.0 | 4850:*:4.0:3.0 |
SKA40:*:4.0:3.0 | SDS2:*:4.0:3.0 | SHG2:*:4.0:3.0 | S7501*:*:4.0:3.0)
    OS_REL=''
    test -r /etc/.releid \
    && OS_REL=`sed -n 's/[^ ]* [^ ]* \([0-9][0-9]\)\.*\/\1/p' <
/etc/.releid`
    /bin/uname -p 2>/dev/null | grep 86 >/dev/null \
    && { echo i486-ncr-sysv4.3${OS_REL}; exit; }
    /bin/uname -p 2>/dev/null | /bin/grep entium >/dev/null \
    && { echo i586-ncr-sysv4.3${OS_REL}; exit; } ;;
3[34]??:*:4.0:* | 3[34]??,*:*:4.0:*)
    /bin/uname -p 2>/dev/null | grep 86 >/dev/null \
    && { echo i486-ncr-sysv4; exit; } ;;
NCR*:*:4.2:* | MPRAS*:*:4.2:*)
    OS_REL='.3'
    test -r /etc/.releid \
    && OS_REL=`sed -n 's/[^ ]* [^ ]* \([0-9][0-9]\)\.*\/\1/p' <
/etc/.releid`
    /bin/uname -p 2>/dev/null | grep 86 >/dev/null \
    && { echo i486-ncr-sysv4.3${OS_REL}; exit; }
    /bin/uname -p 2>/dev/null | /bin/grep entium >/dev/null \
    && { echo i586-ncr-sysv4.3${OS_REL}; exit; }
    /bin/uname -p 2>/dev/null | /bin/grep pteron >/dev/null \
    && { echo i586-ncr-sysv4.3${OS_REL}; exit; } ;;
m68*:LynxOS:2.*:* | m68*:LynxOS:3.0*:**)
    echo m68k-unknown-lynxos${UNAME_RELEASE}
    exit ;;
mc68030:UNIX_System_V:4.*:**)
    echo m68k-atari-sysv4
    exit ;;
TSUNAMI:LynxOS:2.*:**)
    echo sparc-unknown-lynxos${UNAME_RELEASE}
    exit ;;
rs6000:LynxOS:2.*:**)
    echo rs6000-unknown-lynxos${UNAME_RELEASE}
    exit ;;
PowerPC:LynxOS:2.*:* | PowerPC:LynxOS:3.[01]*:* |
PowerPC:LynxOS:4.[02]*:**)
    echo powerpc-unknown-lynxos${UNAME_RELEASE}
    exit ;;
SM[BE]S:UNIX_SV:*:**)
    echo mips-dde-sysv${UNAME_RELEASE}
    exit ;;
RM*:ReliantUNIX-*:*:**)
    echo mips-sni-sysv4
    exit ;;

```

```

RM*:SINIX-*:*:*)
    echo mips-sni-sysv4
    exit ;;
*:SINIX-*:*:*)
    if uname -p 2>/dev/null >/dev/null ; then
        UNAME_MACHINE=`(uname -p) 2>/dev/null`
        echo ${UNAME_MACHINE}-sni-sysv4
    else
        echo ns32k-sni-sysv
    fi
    exit ;;
PENTIUM*:4.0*:*) # Unisys `ClearPath HMP IX 4000' SVR4/MP effort
    # says <Richard.M.Bartel@ccMail.Census.GOV>
    echo i586-unisys-sysv4
    exit ;;
*:UNIX_System_V:4*:FTX*)
    # From Gerald Hewes <hewes@openmarket.com>.
    # How about differentiating between stratus architectures? -djm
    echo hppal.1-stratus-sysv4
    exit ;;
*:*:*:FTX*)
    # From seanf@swdc.stratus.com.
    echo i860-stratus-sysv4
    exit ;;
i*86:VOS:*:*)
    # From Paul.Green@stratus.com.
    echo ${UNAME_MACHINE}-stratus-vos
    exit ;;
*:VOS:*:*)
    # From Paul.Green@stratus.com.
    echo hppal.1-stratus-vos
    exit ;;
mc68*:A/UX:*:*)
    echo m68k-apple-aux${UNAME_RELEASE}
    exit ;;
news*:NEWS-OS:6*:*)
    echo mips-sony-newsos6
    exit ;;
R[34]000:*System_V*:*:* | R4000:UNIX_SYSV*:*:* | R*000:UNIX_SV*:*:*)
    if [ -d /usr/nec ]; then
        echo mips-nec-sysv${UNAME_RELEASE}
    else
        echo mips-unknown-sysv${UNAME_RELEASE}
    fi
    exit ;;
BeBox:BeOS:*:*) # BeOS running on hardware made by Be, PPC only.
    echo powerpc-be-beos
    exit ;;
BeMac:BeOS:*:*) # BeOS running on Mac or Mac clone, PPC only.
    echo powerpc-apple-beos
    exit ;;
BePC:BeOS:*:*) # BeOS running on Intel PC compatible.

```

```

    echo i586-pc-beos
    exit ;;
BePC:Haiku:*:*)    # Haiku running on Intel PC compatible.
    echo i586-pc-haiku
    exit ;;
x86_64:Haiku:*:*)
    echo x86_64-unknown-haiku
    exit ;;
SX-4:SUPER-UX:*:*)
    echo sx4-nec-superux${UNAME_RELEASE}
    exit ;;
SX-5:SUPER-UX:*:*)
    echo sx5-nec-superux${UNAME_RELEASE}
    exit ;;
SX-6:SUPER-UX:*:*)
    echo sx6-nec-superux${UNAME_RELEASE}
    exit ;;
SX-7:SUPER-UX:*:*)
    echo sx7-nec-superux${UNAME_RELEASE}
    exit ;;
SX-8:SUPER-UX:*:*)
    echo sx8-nec-superux${UNAME_RELEASE}
    exit ;;
SX-8R:SUPER-UX:*:*)
    echo sx8r-nec-superux${UNAME_RELEASE}
    exit ;;
Power*:Rhapsody:*:*)
    echo powerpc-apple-rhapsody${UNAME_RELEASE}
    exit ;;
*:Rhapsody:*:*)
    echo ${UNAME_MACHINE}-apple-rhapsody${UNAME_RELEASE}
    exit ;;
*:Darwin:*:*)
    UNAME_PROCESSOR=`uname -p` || UNAME_PROCESSOR=unknown
    case $UNAME_PROCESSOR in
        i386)
            eval $set_cc_for_build
            if [ "$CC_FOR_BUILD" != 'no_compiler_found' ]; then
                if (echo '#ifdef __LP64__'; echo IS_64BIT_ARCH; echo
'#endif') | \
                    (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) | \
                    grep IS_64BIT_ARCH >/dev/null
                then
                    UNAME_PROCESSOR="x86_64"
                fi
            fi
            ;;
        unknown) UNAME_PROCESSOR=powerpc ;;
    esac
    echo ${UNAME_PROCESSOR}-apple-darwin${UNAME_RELEASE}
    exit ;;
*:procnto*:*:* | *:QNX:[0123456789]*:*)
    UNAME_PROCESSOR=`uname -p`

```



```

if test "$UNAME_PROCESSOR" = "x86"; then
    UNAME_PROCESSOR=i386
    UNAME_MACHINE=pc
fi
echo ${UNAME_PROCESSOR}-${UNAME_MACHINE}-nto-qnx${UNAME_RELEASE}
exit ;;
*:QNX*:4*)
    echo i386-pc-qnx
    exit ;;
NEO-?:NONSTOP_KERNEL:*:*)
    echo neo-tandem-nsk${UNAME_RELEASE}
    exit ;;
NSE-*:NONSTOP_KERNEL:*:*)
    echo nse-tandem-nsk${UNAME_RELEASE}
    exit ;;
NSR-?:NONSTOP_KERNEL:*:*)
    echo nsr-tandem-nsk${UNAME_RELEASE}
    exit ;;
*:NonStop-UX:*:*)
    echo mips-compaq-nonstopux
    exit ;;
BS2000:POSIX*:*:*)
    echo bs2000-siemens-sysv
    exit ;;
DS/*:UNIX_System_V:*:*)
    echo ${UNAME_MACHINE}-${UNAME_SYSTEM}-${UNAME_RELEASE}
    exit ;;
*:Plan9:*:*)
    # "uname -m" is not consistent, so use $cputype instead. 386
    # is converted to i386 for consistency with other x86
    # operating systems.
    if test "$cputype" = "386"; then
        UNAME_MACHINE=i386
    else
        UNAME_MACHINE="$cputype"
    fi
    echo ${UNAME_MACHINE}-unknown-plan9
    exit ;;
*:TOPS-10:*:*)
    echo pdp10-unknown-tops10
    exit ;;
*:TENEX:*:*)
    echo pdp10-unknown-tenex
    exit ;;
KS10:TOPS-20:*:* | KL10:TOPS-20:*:* | TYPE4:TOPS-20:*:*)
    echo pdp10-dec-tops20
    exit ;;
XKL-1:TOPS-20:*:* | TYPE5:TOPS-20:*:*)
    echo pdp10-xkl-tops20
    exit ;;
*:TOPS-20:*:*)
    echo pdp10-unknown-tops20

```

```

    exit ;;
*:ITS:*:*)
    echo pdp10-unknown-its
    exit ;;
SEI:*:*:SEIUX)
    echo mips-sei-seiux${UNAME_RELEASE}
    exit ;;
*:DragonFly:*:*)
    echo ${UNAME_MACHINE}-unknown-dragonfly`echo ${UNAME_RELEASE}|sed
-e 's/[-(].*//'`
    exit ;;
*:*VMS:*:*)
    UNAME_MACHINE=`(uname -p) 2>/dev/null`
    case "${UNAME_MACHINE}" in
        A*) echo alpha-dec-vms ; exit ;;
        I*) echo ia64-dec-vms ; exit ;;
        V*) echo vax-dec-vms ; exit ;;
    esac ;;
*:XENIX:*:SysV)
    echo i386-pc-xenix
    exit ;;
i*86:skyos:*:*)
    echo ${UNAME_MACHINE}-pc-skyos`echo ${UNAME_RELEASE}` | sed -e
's/ .*$//'`
    exit ;;
i*86:rdos:*:*)
    echo ${UNAME_MACHINE}-pc-rdos
    exit ;;
i*86:AROS:*:*)
    echo ${UNAME_MACHINE}-pc-aros
    exit ;;
x86_64:VMkernel:*:*)
    echo ${UNAME_MACHINE}-unknown-esx
    exit ;;
esac

eval $set_cc_for_build
cat >$dummy.c <<EOF
#ifdef _SEQUENT_
#include <sys/types.h>
#include <sys/utsname.h>
#endif
main ()
{
#ifdef (sony)
#ifdef (MIPSEB)
    /* BFD wants "bsd" instead of "newsos". Perhaps BFD should be
    changed,
    I don't know.... */
    printf ("mips-sony-bsd\n"); exit (0);
#else
#include <sys/param.h>

```

```

    printf ("m68k-sony-newsos%s\n",
#ifdef NEWSOS4
    "4"
#else
    ""
#endif
    ); exit (0);
#endif
#endif

#ifdef defined (__arm) && defined (__acorn) && defined (__unix)
    printf ("arm-acorn-riscix\n"); exit (0);
#endif

#ifdef defined (hp300) && !defined (hpux)
    printf ("m68k-hp-bsd\n"); exit (0);
#endif

#ifdef defined (NeXT)
#ifdef !defined (__ARCHITECTURE__)
#define __ARCHITECTURE__ "m68k"
#endif
    int version;
    version=`(hostinfo | sed -n 's/. *NeXT Mach \([0-9]*\) .*/\1/p')
2>/dev/null`;
    if (version < 4)
        printf ("%s-next-nextstep%d\n", __ARCHITECTURE__, version);
    else
        printf ("%s-next-openstep%d\n", __ARCHITECTURE__, version);
    exit (0);
#endif

#ifdef defined (MULTIMAX) || defined (n16)
#ifdef defined (UMAXV)
    printf ("ns32k-encore-sysv\n"); exit (0);
#else
#ifdef defined (CMU)
    printf ("ns32k-encore-mach\n"); exit (0);
#else
    printf ("ns32k-encore-bsd\n"); exit (0);
#endif
#endif
#endif

#ifdef defined (__386BSD__)
    printf ("i386-pc-bsd\n"); exit (0);
#endif

#ifdef defined (sequent)
#ifdef defined (i386)
    printf ("i386-sequent-dynix\n"); exit (0);
#endif

```

```

#if defined (ns32000)
    printf ("ns32k-sequent-dynix\n"); exit (0);
#endif
#endif

#if defined (_SEQUENT_)
    struct utsname un;

    uname(&un);

    if (strncmp(un.version, "V2", 2) == 0) {
        printf ("i386-sequent-ptx2\n"); exit (0);
    }
    if (strncmp(un.version, "V1", 2) == 0) { /* XXX is V1 correct? */
        printf ("i386-sequent-ptx1\n"); exit (0);
    }
    printf ("i386-sequent-ptx\n"); exit (0);

#endif

#if defined (vax)
# if !defined (ultrix)
# include <sys/param.h>
# if defined (BSD)
#   if BSD == 43
        printf ("vax-dec-bsd4.3\n"); exit (0);
#   else
#   if BSD == 199006
        printf ("vax-dec-bsd4.3reno\n"); exit (0);
#   else
        printf ("vax-dec-bsd\n"); exit (0);
#   endif
#   endif
# else
        printf ("vax-dec-bsd\n"); exit (0);
# endif
# else
        printf ("vax-dec-ultrix\n"); exit (0);
# endif
#endif

#if defined (alliant) && defined (i860)
    printf ("i860-alliant-bsd\n"); exit (0);
#endif

    exit (1);
}
EOF

$CC_FOR_BUILD -o $dummy $dummy.c 2>/dev/null && SYSTEM_NAME=`$dummy`
&&
    { echo "$SYSTEM_NAME"; exit; }

```

```

# Apollos put the system type in the environment.

test -d /usr/apollo && { echo ${ISP}-apollo-${SYSTYPE}; exit; }

# Convex versions that predate uname can use getsysinfo(1)

if [ -x /usr/convex/getsysinfo ]
then
  case `getsysinfo -f cpu_type` in
    c1*)
      echo c1-convex-bsd
      exit ;;
    c2*)
      if getsysinfo -f scalar_acc
      then echo c32-convex-bsd
      else echo c2-convex-bsd
      fi
      exit ;;
    c34*)
      echo c34-convex-bsd
      exit ;;
    c38*)
      echo c38-convex-bsd
      exit ;;
    c4*)
      echo c4-convex-bsd
      exit ;;
  esac
fi

cat >&2 <<EOF
$0: unable to guess system type

This script, last modified $timestamp, has failed to recognize
the operating system you are using. It is advised that you
download the most up to date version of the config scripts from

http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config.guess;hb=HEAD
and

http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config.sub;hb=HEAD

If the version you run ($0) is already up to date, please
send the following data and any information you think might be
pertinent to <config-patches@gnu.org> in order to provide the needed
information to handle your system.

config.guess timestamp = $timestamp

```

```
uname -m = `(uname -m) 2>/dev/null || echo unknown`
uname -r = `(uname -r) 2>/dev/null || echo unknown`
uname -s = `(uname -s) 2>/dev/null || echo unknown`
uname -v = `(uname -v) 2>/dev/null || echo unknown`
```

```
/usr/bin/uname -p = `(/usr/bin/uname -p) 2>/dev/null`
/bin/uname -X      = `(/bin/uname -X) 2>/dev/null`
```

```
hostinfo          = `(hostinfo) 2>/dev/null`
/bin/universe     = `(/bin/universe) 2>/dev/null`
/usr/bin/arch -k  = `(/usr/bin/arch -k) 2>/dev/null`
/bin/arch         = `(/bin/arch) 2>/dev/null`
/usr/bin/oslevel  = `(/usr/bin/oslevel) 2>/dev/null`
/usr/convex/getsysinfo = `(/usr/convex/getsysinfo) 2>/dev/null`
```

```
UNAME_MACHINE = ${UNAME_MACHINE}
UNAME_RELEASE = ${UNAME_RELEASE}
UNAME_SYSTEM   = ${UNAME_SYSTEM}
UNAME_VERSION  = ${UNAME_VERSION}
EOF
```

```
exit 1
```

```
# Local variables:
# eval: (add-hook 'write-file-hooks 'time-stamp)
# time-stamp-start: "timestamp='"
# time-stamp-format: "%:y-%02m-%02d"
# time-stamp-end: "'"
# End:
```

```
File = config.guess.~1~
```

```
#!/bin/sh
# Attempt to guess a canonical system name.
# Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
# 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
# 2011, 2012 Free Software Foundation, Inc.
```

```
timestamp='2012-08-14'
```

```
# This file is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.
```

```
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, see <http://www.gnu.org/licenses/>.
#
# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program that contains a
# configuration script generated by Autoconf, you may include it under
# the same distribution terms that you use for the rest of that
program.
```

```
# Originally written by Per Bothner. Please send patches (context
# diff format) to <config-patches@gnu.org> and include a ChangeLog
# entry.
```

```
#
# This script attempts to guess a canonical system name similar to
# config.sub. If it succeeds, it prints the system name on stdout,
and
# exits with 0. Otherwise, it exits with 1.
```

```
#
# You can get the latest version of this script from:
```

```
#
http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config.guess;hb=HEAD
```

```
me=`echo "$0" | sed -e 's,./,,'`
```

```
usage="\
Usage: $0 [OPTION]
```

```
Output the configuration name of the system \"$me' is run on.
```

```
Operation modes:
```

```
-h, --help           print this help, then exit
-t, --time-stamp    print date of last modification, then exit
-v, --version       print version number, then exit
```

```
Report bugs and patches to <config-patches@gnu.org>."
```

```
version="\
GNU config.guess ($timestamp)
```

```
Originally written by Per Bothner.
Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012
Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE."
```

```

help="
Try \`$me --help' for more information."

# Parse command line
while test $# -gt 0 ; do
  case $1 in
    --time-stamp | --time* | -t )
      echo "$timestamp" ; exit ;;
    --version | -v )
      echo "$version" ; exit ;;
    --help | --h* | -h )
      echo "$usage"; exit ;;
    -- )
      # Stop option processing
      shift; break ;;
    - )
      # Use stdin as input.
      break ;;
    -* )
      echo "$me: invalid option $1$help" >&2
      exit 1 ;;
    * )
      break ;;
  esac
done

if test $# != 0; then
  echo "$me: too many arguments$help" >&2
  exit 1
fi

trap 'exit 1' 1 2 15

# CC_FOR_BUILD -- compiler used by this script. Note that the use of a
# compiler to aid in system detection is discouraged as it requires
# temporary files to be created and, as you can see below, it is a
# headache to deal with in a portable fashion.

# Historically, `CC_FOR_BUILD' used to be named `HOST_CC'. We still
# use `HOST_CC' if defined, but it is deprecated.

# Portable tmp directory creation inspired by the Autoconf team.

set_cc_for_build='
trap "exitcode=\$?; (rm -f \${tmpfiles} 2>/dev/null; rmdir \${tmp}
2>/dev/null) && exit \${exitcode}" 0 ;
trap "rm -f \${tmpfiles} 2>/dev/null; rmdir \${tmp} 2>/dev/null; exit 1" 1
2 13 15 ;
: ${TMPDIR=/tmp} ;
{ tmp=`(umask 077 && mktemp -d "$TMPDIR/cgXXXXXX") 2>/dev/null` &&
test -n "$tmp" && test -d "$tmp" ; } ||
{ test -n "$RANDOM" && tmp=$TMPDIR/cg$$-$RANDOM && (umask 077 &&
mkdir $tmp) ; } ||

```



```

{ tmp=$TMPDIR/cg-$$ && (umask 077 && mkdir $tmp) && echo "Warning:
creating insecure temp directory" >&2 ; } ||
{ echo "$me: cannot create a temporary directory in $TMPDIR" >&2 ;
exit 1 ; } ;
dummy=$tmp/dummy ;
tmpfiles="$dummy.c $dummy.o $dummy.rel $dummy" ;
case $CC_FOR_BUILD,$HOST_CC,$CC in
,,)    echo "int x;" > $dummy.c ;
      for c in cc gcc c89 c99 ; do
        if ($c -c -o $dummy.o $dummy.c) >/dev/null 2>&1 ; then
          CC_FOR_BUILD="$c" ; break ;
        fi ;
      done ;
      if test x"$CC_FOR_BUILD" = x ; then
        CC_FOR_BUILD=no_compiler_found ;
      fi
      ;;
,*,*)  CC_FOR_BUILD=$CC ;;
,*,*)  CC_FOR_BUILD=$HOST_CC ;;
esac ; set_cc_for_build= ;'

```

```

# This is needed to find uname on a Pyramid OSx when run in the BSD
universe.

```

```

# (ghazi@noc.rutgers.edu 1994-08-24)
if (test -f /.attbin/uname) >/dev/null 2>&1 ; then
  PATH=$PATH:/.attbin ; export PATH
fi

```

```

UNAME_MACHINE=`(uname -m) 2>/dev/null` || UNAME_MACHINE=unknown
UNAME_RELEASE=`(uname -r) 2>/dev/null` || UNAME_RELEASE=unknown
UNAME_SYSTEM=`(uname -s) 2>/dev/null` || UNAME_SYSTEM=unknown
UNAME_VERSION=`(uname -v) 2>/dev/null` || UNAME_VERSION=unknown

```

```

# Detect uclibc systems.

```

```

LIBC="gnu"
if [ -f /usr/include/bits/uClibc_config.h ]
then
  LIBC=uclibc
  if [ -n `grep "#define __UCLIBC_CONFIG_VERSION__"
/usr/include/bits/uClibc_config.h` ]
  then
    UCLIBC_SUBVER=`sed -n "/#define __UCLIBC_CONFIG_VERSION__
/s///p" /usr/include/bits/uClibc_config.h`
    LIBC=$LIBC$UCLIBC_SUBVER
  fi
fi

```

```

# Note: order is significant - the case branches are not exclusive.

```

```

case
"${UNAME_MACHINE}:${UNAME_SYSTEM}:${UNAME_RELEASE}:${UNAME_VERSION}"
in
*:NetBSD:*:*)
# NetBSD (nbsd) targets should (where applicable) match one or
# more of the tuples: *-*-netbsdelf*, *-*-netbsdout*,
# *-*-netbsdcoeff* and *-*-netbsd*. For targets that recently
# switched to ELF, *-*-netbsd* would select the old
# object file format. This provides both forward
# compatibility and a consistent mechanism for selecting the
# object file format.
#
# Note: NetBSD doesn't particularly care about the vendor
# portion of the name. We always set it to "unknown".
sysctl="sysctl -n hw.machine_arch"
UNAME_MACHINE_ARCH=`(/sbin/$sysctl 2>/dev/null || \
/usr/sbin/$sysctl 2>/dev/null || echo unknown)`
case "${UNAME_MACHINE_ARCH}" in
armeb) machine=armeb-unknown ;;
arm*) machine=arm-unknown ;;
sh3el) machine=sh1-unknown ;;
sh3eb) machine=sh-unknown ;;
sh5el) machine=sh5le-unknown ;;
*) machine=${UNAME_MACHINE_ARCH}-unknown ;;
esac
# The Operating System including object format, if it has
switched
# to ELF recently, or will in the future.
case "${UNAME_MACHINE_ARCH}" in
arm*|i386|m68k|ns32k|sh3*|sparc|vax)
eval $set_cc_for_build
if echo __ELF__ | $CC_FOR_BUILD -E - 2>/dev/null \
| grep -q __ELF__
then
# Once all utilities can be ECOFF (netbsdcoeff) or
a.out (netbsdout).
# Return netbsd for either. FIX?
os=netbsd
else
os=netbsdelf
fi
;;
*)
os=netbsd
;;
esac
# The OS release
# Debian GNU/NetBSD machines have a different userland, and
# thus, need a distinct triplet. However, they do not need
# kernel version information, so it can be replaced with a
# suitable tag, in the style of linux-gnu.
case "${UNAME_VERSION}" in

```

```

    Debian*)
        release='-gnu'
        ;;
    *)
        release=`echo ${UNAME_RELEASE}|sed -e 's/[-_].*/\./'`
        ;;
esac
# Since CPU_TYPE-MANUFACTURER-KERNEL-OPERATING_SYSTEM:
# contains redundant information, the shorter form:
# CPU_TYPE-MANUFACTURER-OPERATING_SYSTEM is used.
echo "${machine}-${os}${release}"
exit ;;
*:Bitrig:*)
    UNAME_MACHINE_ARCH=`arch | sed 's/Bitrig.//'`
    echo ${UNAME_MACHINE_ARCH}-unknown-bitrig${UNAME_RELEASE}
    exit ;;
*:OpenBSD:*)
    UNAME_MACHINE_ARCH=`arch | sed 's/OpenBSD.//'`
    echo ${UNAME_MACHINE_ARCH}-unknown-openbsd${UNAME_RELEASE}
    exit ;;
*:ekkoBSD:*)
    echo ${UNAME_MACHINE}-unknown-ekkobsd${UNAME_RELEASE}
    exit ;;
*:SolidBSD:*)
    echo ${UNAME_MACHINE}-unknown-solidbsd${UNAME_RELEASE}
    exit ;;
macppc:MirBSD:*)
    echo powerpc-unknown-mirbsd${UNAME_RELEASE}
    exit ;;
*:MirBSD:*)
    echo ${UNAME_MACHINE}-unknown-mirbsd${UNAME_RELEASE}
    exit ;;
alpha:OSF1:*)
    case $UNAME_RELEASE in
        *4.0)
            UNAME_RELEASE=`/usr/sbin/sizer -v | awk '{print $3}'`
            ;;
        *5.*)
            UNAME_RELEASE=`/usr/sbin/sizer -v | awk '{print $4}'`
            ;;
    esac
# According to Compaq, /usr/sbin/psrinfo has been available on
# OSF/1 and Tru64 systems produced since 1995. I hope that
# covers most systems running today. This code pipes the CPU
# types through head -n 1, so we only detect the type of CPU 0.
ALPHA_CPU_TYPE=`/usr/sbin/psrinfo -v | sed -n -e 's/^ The alpha
\(.*\) processor.*$/\1/p' | head -n 1`
case "$ALPHA_CPU_TYPE" in
    "EV4 (21064)")
        UNAME_MACHINE="alpha" ;;
    "EV4.5 (21064)")
        UNAME_MACHINE="alpha" ;;

```

```

"LCA4 (21066/21068)")
  UNAME_MACHINE="alpha" ;;
"EV5 (21164)")
  UNAME_MACHINE="alphaev5" ;;
"EV5.6 (21164A)")
  UNAME_MACHINE="alphaev56" ;;
"EV5.6 (21164PC)")
  UNAME_MACHINE="alphapca56" ;;
"EV5.7 (21164PC)")
  UNAME_MACHINE="alphapca57" ;;
"EV6 (21264)")
  UNAME_MACHINE="alphaev6" ;;
"EV6.7 (21264A)")
  UNAME_MACHINE="alphaev67" ;;
"EV6.8CB (21264C)")
  UNAME_MACHINE="alphaev68" ;;
"EV6.8AL (21264B)")
  UNAME_MACHINE="alphaev68" ;;
"EV6.8CX (21264D)")
  UNAME_MACHINE="alphaev68" ;;
"EV6.9A (21264/EV69A)")
  UNAME_MACHINE="alphaev69" ;;
"EV7 (21364)")
  UNAME_MACHINE="alphaev7" ;;
"EV7.9 (21364A)")
  UNAME_MACHINE="alphaev79" ;;
esac
# A Pn.n version is a patched version.
# A Vn.n version is a released version.
# A Tn.n version is a released field test version.
# A Xn.n version is an unreleased experimental baselevel.
# 1.2 uses "1.2" for uname -r.
echo ${UNAME_MACHINE}-dec-osf`echo ${UNAME_RELEASE} | sed -e
's/^[PVTX]//' | tr 'ABCDEFGHijklmnopqrstuvwxyz'
'abcdefghijklmnopqrstuvwxyx`
# Reset EXIT trap before exiting to avoid spurious non-zero exit
code.
exitcode=$?
trap '' 0
exit $exitcode ;;
Alpha\ *:Windows_NT*:*)
# How do we know it's Interix rather than the generic POSIX
subsystem?
# Should we change UNAME_MACHINE based on the output of uname
instead
# of the specific Alpha model?
echo alpha-pc-interix
exit ;;
21064:Windows_NT:50:3)
echo alpha-dec-winnt3.5
exit ;;
Amiga*:UNIX_System_V:4.0:*)

```

```

    echo m68k-unknown-sysv4
    exit ;;
*: [Aa]miga [Oo] [Ss]:*:*)
    echo ${UNAME_MACHINE}-unknown-amigaos
    exit ;;
*: [Mm]orph [Oo] [Ss]:*:*)
    echo ${UNAME_MACHINE}-unknown-morphos
    exit ;;
*: OS/390:*:*)
    echo i370-ibm-openedition
    exit ;;
*: z/VM:*:*)
    echo s390-ibm-zvmoe
    exit ;;
*: OS400:*:*)
    echo powerpc-ibm-os400
    exit ;;
arm:RISC*:1.[012]*:* | arm:riscix:1.[012]*:*)
    echo arm-acorn-riscix${UNAME_RELEASE}
    exit ;;
arm:riscos:*:* | arm:RISCOS:*:*)
    echo arm-unknown-riscos
    exit ;;
SR2?01:HI-UX/MPP:*:* | SR8000:HI-UX/MPP:*:*)
    echo hppa1.1-hitachi-hiuxmpp
    exit ;;
Pyramid*:OSx*:*:* | MIS*:OSx*:*:* | MIS*:SMP_DC-OSx*:*:*)
    # akee@wpdis03.wpafb.af.mil (Earle F. Ake) contributed MIS and
NILE.
    if test "`(/bin/universe) 2>/dev/null`" = att ; then
        echo pyramid-pyramid-sysv3
    else
        echo pyramid-pyramid-bsd
    fi
    exit ;;
NILE*:*:*:dcosx)
    echo pyramid-pyramid-svr4
    exit ;;
DRS?6000:unix:4.0:6*)
    echo sparc-icl-nx6
    exit ;;
DRS?6000:UNIX_SV:4.2*:7* | DRS?6000:isis:4.2*:7*)
    case `/usr/bin/uname -p` in
        sparc) echo sparc-icl-nx7; exit ;;
    esac ;;
s390x:SunOS:*:*)
    echo ${UNAME_MACHINE}-ibm-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^.]*//`
    exit ;;
sun4H:SunOS:5.*:*)
    echo sparc-hal-solaris2`echo ${UNAME_RELEASE}|sed -e 's/[^.]*//`
    exit ;;

```

```

sun4*:SunOS:5.*:* | tadpole*:SunOS:5.*:*)
    echo sparc-sun-solaris2`echo ${UNAME_RELEASE}|sed -e 's/[^.]*/'`
    exit ;;
i86pc:AuroraUX:5.*:* | i86xen:AuroraUX:5.*:*)
    echo i386-pc-auroraux${UNAME_RELEASE}
    exit ;;
i86pc:SunOS:5.*:* | i86xen:SunOS:5.*:*)
    eval $set_cc_for_build
    SUN_ARCH="i386"
    # If there is a compiler, see if it is configured for 64-bit
objects.
    # Note that the Sun cc does not turn __LP64__ into 1 like gcc
does.
    # This test works for both compilers.
    if [ "$CC_FOR_BUILD" != 'no_compiler_found' ]; then
        if (echo '#ifdef __amd64'; echo IS_64BIT_ARCH; echo '#endif')
| \
            (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) | \
            grep IS_64BIT_ARCH >/dev/null
        then
            SUN_ARCH="x86_64"
        fi
    fi
    echo ${SUN_ARCH}-pc-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^.]*/'`
    exit ;;
sun4*:SunOS:6.*:*)
    # According to config.sub, this is the proper way to canonicalize
    # SunOS6. Hard to guess exactly what SunOS6 will be like, but
    # it's likely to be more like Solaris than SunOS4.
    echo sparc-sun-solaris3`echo ${UNAME_RELEASE}|sed -e 's/[^.]*/'`
    exit ;;
sun4*:SunOS:*:*)
    case "`/usr/bin/arch -k`" in
        Series*|S4*)
            UNAME_RELEASE=`uname -v`
            ;;
    esac
    # Japanese Language versions have a version number like `4.1.3-
JL'.
    echo sparc-sun-sunos`echo ${UNAME_RELEASE}|sed -e 's/-/_/'`
    exit ;;
sun3*:SunOS:*:*)
    echo m68k-sun-sunos${UNAME_RELEASE}
    exit ;;
sun*:*:4.2BSD:*)
    UNAME_RELEASE=`(sed 1q /etc/motd | awk '{print substr($5,1,3)}')
2>/dev/null`
    test "x${UNAME_RELEASE}" = "x" && UNAME_RELEASE=3
    case "`/bin/arch`" in
        sun3)
            echo m68k-sun-sunos${UNAME_RELEASE}

```

```

        ;;
        sun4)
            echo sparc-sun-sunos${UNAME_RELEASE}
        ;;
    esac
    exit ;;
aushp:SunOS:*:*)
    echo sparc-auspex-sunos${UNAME_RELEASE}
    exit ;;
# The situation for MiNT is a little confusing.  The machine name
# can be virtually everything (everything which is not
# "atarist" or "atariste" at least should have a processor
# > m68000).  The system name ranges from "MiNT" over "FreeMiNT"
# to the lowercase version "mint" (or "freemint").  Finally
# the system name "TOS" denotes a system which is actually not
# MiNT.  But MiNT is downward compatible to TOS, so this should
# be no problem.
atarist[e]:*MiNT:*:* | atarist[e]:*mint:*:* | atarist[e]:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
atari*:*MiNT:*:* | atari*:*mint:*:* | atarist[e]:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
*falcon*:*MiNT:*:* | *falcon*:*mint:*:* | *falcon*:*TOS:*:*)
    echo m68k-atari-mint${UNAME_RELEASE}
    exit ;;
milan*:*MiNT:*:* | milan*:*mint:*:* | *milan*:*TOS:*:*)
    echo m68k-milan-mint${UNAME_RELEASE}
    exit ;;
hades*:*MiNT:*:* | hades*:*mint:*:* | *hades*:*TOS:*:*)
    echo m68k-hades-mint${UNAME_RELEASE}
    exit ;;
*:*MiNT:*:* | *:*mint:*:* | *:*TOS:*:*)
    echo m68k-unknown-mint${UNAME_RELEASE}
    exit ;;
m68k:machten:*:*)
    echo m68k-apple-machten${UNAME_RELEASE}
    exit ;;
powerpc:machten:*:*)
    echo powerpc-apple-machten${UNAME_RELEASE}
    exit ;;
RISC*:Mach:*:*)
    echo mips-dec-mach_bsd4.3
    exit ;;
RISC*:ULTRIX:*:*)
    echo mips-dec-ultrix${UNAME_RELEASE}
    exit ;;
VAX*:ULTRIX*:*:*)
    echo vax-dec-ultrix${UNAME_RELEASE}
    exit ;;
2020:CLIX:*:* | 2430:CLIX:*:*)
    echo clipper-intergraph-clix${UNAME_RELEASE}

```

```

        exit ;;
mips:*:*:UMIPS | mips:*:*:RISCos)
    eval $set_cc_for_build
    sed 's/^_ //' << EOF >$dummy.c
#ifdef __cplusplus
#include <stdio.h> /* for printf() prototype */
    int main (int argc, char *argv[]) {
#else
    int main (argc, argv) int argc; char *argv[]; {
#endif
    #if defined (host_mips) && defined (MIPSEB)
    #if defined (SYSTYPE_SYSV)
        printf ("mips-mips-riscos%ssysv\n", argv[1]); exit (0);
    #endif
    #if defined (SYSTYPE_SVR4)
        printf ("mips-mips-riscos%ssvr4\n", argv[1]); exit (0);
    #endif
    #if defined (SYSTYPE_BSD43) || defined(SYSTYPE_BSD)
        printf ("mips-mips-riscos%bsbd\n", argv[1]); exit (0);
    #endif
    #endif
        exit (-1);
    }
EOF
    $CC_FOR_BUILD -o $dummy $dummy.c &&
    dummyarg=`echo "${UNAME_RELEASE}" | sed -n 's/\([0-9]*\).*/\1/p'` &&
    SYSTEM_NAME=`$dummy $dummyarg` &&
    { echo "$SYSTEM_NAME"; exit; }
    echo mips-mips-riscos${UNAME_RELEASE}
    exit ;;
Motorola:PowerMAX_OS:*:*)
    echo powerpc-motorola-powermax
    exit ;;
Motorola:*:4.3:PL8-*)
    echo powerpc-harris-powermax
    exit ;;
Night_Hawk:*:*:PowerMAX_OS | Synergy:PowerMAX_OS:*:*)
    echo powerpc-harris-powermax
    exit ;;
Night_Hawk:Power_UNIX:*:*)
    echo powerpc-harris-powerunix
    exit ;;
m88k:CX/UX:7:*:*)
    echo m88k-harris-cxux7
    exit ;;
m88k:*:4*:R4*)
    echo m88k-motorola-sysv4
    exit ;;
m88k:*:3*:R3*)
    echo m88k-motorola-sysv3
    exit ;;

```



```

AViION:dgux:*:*)
# DG/UX returns AViION for all architectures
UNAME_PROCESSOR=`/usr/bin/uname -p`
if [ $UNAME_PROCESSOR = mc88100 ] || [ $UNAME_PROCESSOR = mc88110
]
then
    if [ ${TARGET_BINARY_INTERFACE}x = m88kdguxelfx ] || \
        [ ${TARGET_BINARY_INTERFACE}x = x ]
    then
        echo m88k-dg-dgux${UNAME_RELEASE}
    else
        echo m88k-dg-dguxbcs${UNAME_RELEASE}
    fi
else
    echo i586-dg-dgux${UNAME_RELEASE}
fi
exit ;;
M88*:DolphinOS:*:*) # DolphinOS (SVR3)
echo m88k-dolphin-sysv3
exit ;;
M88*:*:R3:*:*)
# Delta 88k system running SVR3
echo m88k-motorola-sysv3
exit ;;
XD88*:*:*:*) # Tektronix XD88 system running UTekV (SVR3)
echo m88k-tektronix-sysv3
exit ;;
Tek43[0-9][0-9]:UTek:*:*) # Tektronix 4300 system running UTek
(BSD)
echo m68k-tektronix-bsd
exit ;;
*:IRIX*:*:*)
echo mips-sgi-irix`echo ${UNAME_RELEASE}|sed -e 's/-/_/g'`
exit ;;
?????????:AIX?:[12].1:2) # AIX 2.2.1 or AIX 2.1.1 is RT/PC AIX.
echo romp-ibm-aix # uname -m gives an 8 hex-code CPU id
exit ;; # Note that: echo "`uname -s`" gives 'AIX

i*86:AIX:*:*)
echo i386-ibm-aix
exit ;;
ia64:AIX:*:*)
if [ -x /usr/bin/oslevel ] ; then
    IBM_REV=`/usr/bin/oslevel`
else
    IBM_REV=${UNAME_VERSION}.${UNAME_RELEASE}
fi
echo ${UNAME_MACHINE}-ibm-aix${IBM_REV}
exit ;;
*:AIX:2:3)
if grep bos325 /usr/include/stdio.h >/dev/null 2>&1; then
    eval `set_cc_for_build

```

```

sed 's/^          //' << EOF >$dummy.c
#include <sys/systemcfg.h>

main()
{
    if (!__power_pc())
        exit(1);
    puts("powerpc-ibm-aix3.2.5");
    exit(0);
}
EOF

if $CC_FOR_BUILD -o $dummy $dummy.c && SYSTEM_NAME=`$dummy`
then
    echo "$SYSTEM_NAME"
else
    echo rs6000-ibm-aix3.2.5
fi
elif grep bos324 /usr/include/stdio.h >/dev/null 2>&1; then
    echo rs6000-ibm-aix3.2.4
else
    echo rs6000-ibm-aix3.2
fi
exit ;;
*:AIX:*: [4567])
    IBM_CPU_ID=`/usr/sbin/lsdev -C -c processor -S available | sed 1q
| awk '{ print $1 }'`
    if /usr/sbin/lsattr -El ${IBM_CPU_ID} | grep ' POWER' >/dev/null
2>&1; then
        IBM_ARCH=rs6000
    else
        IBM_ARCH=powerpc
    fi
    if [ -x /usr/bin/oslevel ] ; then
        IBM_REV=`/usr/bin/oslevel`
    else
        IBM_REV=${UNAME_VERSION}.${UNAME_RELEASE}
    fi
    echo ${IBM_ARCH}-ibm-aix${IBM_REV}
    exit ;;
*:AIX:*:*)
    echo rs6000-ibm-aix
    exit ;;
ibmrt:4.4BSD:*|romp-ibm:BSD:*)
    echo romp-ibm-bsd4.4
    exit ;;
ibmrt:*BSD:*|romp-ibm:BSD:*)
    echo romp-ibm-bsd${UNAME_RELEASE} # covers RT/PC BSD and
# 4.3 with uname added to
# report: romp-ibm BSD 4.3
    exit ;;
*:BOSX:*:*)
    echo rs6000-bull-bosx
    exit ;;
DPX/2?00:B.O.S.:*:*)

```

```

    echo m68k-bull-sysv3
    exit ;;
9000/[34]?:4.3bsd:1.*:*)
    echo m68k-hp-bsd
    exit ;;
hp300:4.4BSD:*:* | 9000/[34]?:4.3bsd:2.*:*)
    echo m68k-hp-bsd4.4
    exit ;;
9000/[34678]?:HP-UX:*:*)
HPUX_REV=`echo ${UNAME_RELEASE}|sed -e 's/[^.]*.[0B]*//'\`
case "${UNAME_MACHINE}" in
    9000/31? )           HP_ARCH=m68000 ;;
    9000/[34]?? )      HP_ARCH=m68k ;;
    9000/[678][0-9][0-9])
        if [ -x /usr/bin/getconf ]; then
            sc_cpu_version=`/usr/bin/getconf SC_CPU_VERSION
2>/dev/null`
            sc_kernel_bits=`/usr/bin/getconf SC_KERNEL_BITS
2>/dev/null`
            case "${sc_cpu_version}" in
                523) HP_ARCH="hppa1.0" ;; # CPU_PA_RISC1_0
                528) HP_ARCH="hppa1.1" ;; # CPU_PA_RISC1_1
                532)           # CPU_PA_RISC2_0
            case "${sc_kernel_bits}" in
                32) HP_ARCH="hppa2.0n" ;;
                64) HP_ARCH="hppa2.0w" ;;
                ') HP_ARCH="hppa2.0" ;; # HP-UX 10.20
            esac ;;
        esac
    esac
fi
if [ "${HP_ARCH}" = "" ]; then
    eval `set_cc_for_build
sed 's/^          //' << EOF >$dummy.c

#define _HPUX_SOURCE
#include <stdlib.h>
#include <unistd.h>

int main ()
{
    #if defined(_SC_KERNEL_BITS)
        long bits = sysconf(_SC_KERNEL_BITS);
    #endif
        long cpu = sysconf (_SC_CPU_VERSION);

    switch (cpu)
    {
        case CPU_PA_RISC1_0: puts ("hppa1.0"); break;
        case CPU_PA_RISC1_1: puts ("hppa1.1"); break;
        case CPU_PA_RISC2_0:
    #if defined(_SC_KERNEL_BITS)
        switch (bits)

```

```

        {
            case 64: puts ("hppa2.0w"); break;
            case 32: puts ("hppa2.0n"); break;
            default: puts ("hppa2.0"); break;
        } break;
    #else /* !defined(_SC_KERNEL_BITS) */
        puts ("hppa2.0"); break;
    #endif
        default: puts ("hppa1.0"); break;
    }
    exit (0);
}
EOF
        (CCOPTS= $CC_FOR_BUILD -o $dummy $dummy.c 2>/dev/null)
&& HP_ARCH=`$dummy`
        test -z "$HP_ARCH" && HP_ARCH=hppa
        fi ;;
esac
if [ ${HP_ARCH} = "hppa2.0w" ]
then
    eval $set_cc_for_build

    # hppa2.0w-hp-hpux* has a 64-bit kernel and a compiler
generating
    # 32-bit code.  hppa64-hp-hpux* has the same kernel and a
compiler
    # generating 64-bit code.  GNU and HP use different
nomenclature:
    #
    # $ CC_FOR_BUILD=cc ./config.guess
    # => hppa2.0w-hp-hpux11.23
    # $ CC_FOR_BUILD="cc +DA2.0w" ./config.guess
    # => hppa64-hp-hpux11.23

    if echo __LP64__ | (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) |
        grep -q __LP64__
    then
        HP_ARCH="hppa2.0w"
    else
        HP_ARCH="hppa64"
    fi
fi
echo ${HP_ARCH}-hp-hpux${HPUX_REV}
exit ;;
ia64:HP-UX:*:*)
    HPUX_REV=`echo ${UNAME_RELEASE}|sed -e 's/[^.]*. [0B]*//`
    echo ia64-hp-hpux${HPUX_REV}
    exit ;;
3050*:HI-UX:*:*)
    eval $set_cc_for_build
    sed 's/^ // ' << EOF >$dummy.c
    #include <unistd.h>

```

```

int
main ()
{
    long cpu = sysconf (_SC_CPU_VERSION);
    /* The order matters, because CPU_IS_HP_MC68K erroneously
returns
    true for CPU_PA_RISC1_0. CPU_IS_PA_RISC returns correct
    results, however. */
    if (CPU_IS_PA_RISC (cpu))
    {
        switch (cpu)
        {
            case CPU_PA_RISC1_0: puts ("hppa1.0-hitachi-hiuxwe2");
break;
            case CPU_PA_RISC1_1: puts ("hppa1.1-hitachi-hiuxwe2");
break;
            case CPU_PA_RISC2_0: puts ("hppa2.0-hitachi-hiuxwe2");
break;
            default: puts ("hppa-hitachi-hiuxwe2"); break;
        }
    }
    else if (CPU_IS_HP_MC68K (cpu))
        puts ("m68k-hitachi-hiuxwe2");
    else puts ("unknown-hitachi-hiuxwe2");
    exit (0);
}
EOF
$CC_FOR_BUILD -o $dummy $dummy.c && SYSTEM_NAME=`$dummy` &&
    { echo "$SYSTEM_NAME"; exit; }
echo unknown-hitachi-hiuxwe2
exit ;;
9000/7???:4.3bsd:*:* | 9000/8?[79]:4.3bsd:*:* )
    echo hppa1.1-hp-bsd
    exit ;;
9000/8???:4.3bsd:*:*)
    echo hppa1.0-hp-bsd
    exit ;;
*9???:MPE/iX:*:* | *3000*:MPE/iX:*:*)
    echo hppa1.0-hp-mpeix
    exit ;;
hp7???:OSF1:*:* | hp8?[79]:OSF1:*:* )
    echo hppa1.1-hp-osf
    exit ;;
hp8???:OSF1:*:*)
    echo hppa1.0-hp-osf
    exit ;;
i*86:OSF1:*:*)
    if [ -x /usr/sbin/sysversion ] ; then
        echo ${UNAME_MACHINE}-unknown-osf1mk
    else
        echo ${UNAME_MACHINE}-unknown-osf1
    fi

```

```

    exit ;;
parisc*:Lites*:*:*)
    echo hppal.1-hp-lites
    exit ;;
C1*:ConvexOS:*:* | convex:ConvexOS:C1*:**)
    echo c1-convex-bsd
    exit ;;
C2*:ConvexOS:*:* | convex:ConvexOS:C2*:**)
    if getsysinfo -f scalar_acc
    then echo c32-convex-bsd
    else echo c2-convex-bsd
    fi
    exit ;;
C34*:ConvexOS:*:* | convex:ConvexOS:C34*:**)
    echo c34-convex-bsd
    exit ;;
C38*:ConvexOS:*:* | convex:ConvexOS:C38*:**)
    echo c38-convex-bsd
    exit ;;
C4*:ConvexOS:*:* | convex:ConvexOS:C4*:**)
    echo c4-convex-bsd
    exit ;;
CRAY*Y-MP:*:*:*)
    echo ymp-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
    exit ;;
CRAY*[A-Z]90:*:*:*)
    echo ${UNAME_MACHINE}-cray-unicos${UNAME_RELEASE} \
    | sed -e 's/CRAY.*\([A-Z]90\)/\1/' \
        -e y/ABCDEFGHIJKLMNOPQRSTUVWXYZ/abcdefghijklmnopqrstuvwxyzyz/
\
        -e 's/\.[^.]*/.X/'
    exit ;;
CRAY*T90:*:*:*)
    echo t90-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
    exit ;;
CRAY*T3E:*:*:*)
    echo alphaev5-cray-unicosmk${UNAME_RELEASE} | sed -e
's/\.[^.]*/.X/'
    exit ;;
CRAY*SV1:*:*:*)
    echo sv1-cray-unicos${UNAME_RELEASE} | sed -e 's/\.[^.]*/.X/'
    exit ;;
*:UNICOS/mp:*:*:*)
    echo craynv-cray-unicosmp${UNAME_RELEASE} | sed -e
's/\.[^.]*/.X/'
    exit ;;
F30[01]:UNIX_System_V:*:* | F700:UNIX_System_V:*:**)
    FUJITSU_PROC=`uname -m | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxyzyz'`
    FUJITSU_SYS=`uname -p | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxyzyz'` | sed -e 's/\/\///'`
    FUJITSU_REL=`echo ${UNAME_RELEASE} | sed -e 's/ /_/'`

```

```

    echo "${FUJITSU_PROC}-fujitsu-${FUJITSU_SYS}${FUJITSU_REL}"
    exit ;;
5000:UNIX_System_V:4.*:*)
    FUJITSU_SYS=`uname -p | tr 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
'abcdefghijklmnopqrstuvwxy' | sed -e 's/\///'`
    FUJITSU_REL=`echo ${UNAME_RELEASE} | tr
'ABCDEFGHIJKLMNOPQRSTUVWXYZ' 'abcdefghijklmnopqrstuvwxy' | sed -e 's/
/_/'`
    echo "sparc-fujitsu-${FUJITSU_SYS}${FUJITSU_REL}"
    exit ;;
i*86:BSD/386:*:* | i*86:BSD/OS:*:* | *:Ascend\ Embedded/OS:*:*)
    echo ${UNAME_MACHINE}-pc-bsdi${UNAME_RELEASE}
    exit ;;
sparc*:BSD/OS:*:*)
    echo sparc-unknown-bsdi${UNAME_RELEASE}
    exit ;;
*:BSD/OS:*:*)
    echo ${UNAME_MACHINE}-unknown-bsdi${UNAME_RELEASE}
    exit ;;
*:FreeBSD:*:*)
    UNAME_PROCESSOR=`/usr/bin/uname -p`
    case ${UNAME_PROCESSOR} in
        amd64)
            echo x86_64-unknown-freebsd`echo ${UNAME_RELEASE}|sed -e
's/[-(].*//'` ;;
        *)
            echo ${UNAME_MACHINE}-unknown-freebsd`echo
${UNAME_RELEASE}|sed -e 's/[-(].*//'` ;;
    esac
    exit ;;
i*:CYGWIN*:**)
    echo ${UNAME_MACHINE}-pc-cygwin
    exit ;;
*:MINGW64*:**)
    echo ${UNAME_MACHINE}-pc-mingw64
    exit ;;
*:MINGW*:**)
    echo ${UNAME_MACHINE}-pc-mingw32
    exit ;;
i*:MSYS*:**)
    echo ${UNAME_MACHINE}-pc-msys
    exit ;;
i*:windows32*:**)
    # uname -m includes "-pc" on this system.
    echo ${UNAME_MACHINE}-mingw32
    exit ;;
i*:PW*:**)
    echo ${UNAME_MACHINE}-pc-pw32
    exit ;;
*:Interix*:**)
    case ${UNAME_MACHINE} in
        x86)

```

```

        echo i586-pc-interix${UNAME_RELEASE}
        exit ;;
    authenticamd | genuineintel | EM64T)
        echo x86_64-unknown-interix${UNAME_RELEASE}
        exit ;;
    IA64)
        echo ia64-unknown-interix${UNAME_RELEASE}
        exit ;;
    esac ;;
[345]86:Windows_95:* | [345]86:Windows_98:* |
[345]86:Windows_NT:*)
    echo i${UNAME_MACHINE}-pc-mks
    exit ;;
8664:Windows_NT:*)
    echo x86_64-pc-mks
    exit ;;
i*:Windows_NT*:*) | Pentium*:Windows_NT*:*)
    # How do we know it's Interix rather than the generic POSIX
subsystem?
    # It also conflicts with pre-2.0 versions of AT&T UWIN. Should we
    # UNAME_MACHINE based on the output of uname instead of i386?
    echo i586-pc-interix
    exit ;;
i*:UWIN*:*)
    echo ${UNAME_MACHINE}-pc-uwin
    exit ;;
amd64:CYGWIN*:*:* | x86_64:CYGWIN*:*:*)
    echo x86_64-unknown-cygwin
    exit ;;
p*:CYGWIN*:*)
    echo powerpcle-unknown-cygwin
    exit ;;
prep*:SunOS:5.*:*)
    echo powerpcle-unknown-solaris2`echo ${UNAME_RELEASE}|sed -e
's/[^.]*/'\`
    exit ;;
*:GNU*:*:*)
    # the GNU system
    echo `echo ${UNAME_MACHINE}|sed -e 's,[-/].*$,,'\`-unknown-
gnu`echo ${UNAME_RELEASE}|sed -e 's,/.*$,,\`
    exit ;;
*:GNU/*:*:*)
    # other systems with GNU libc and userland
    echo ${UNAME_MACHINE}-unknown-`echo ${UNAME_SYSTEM} | sed
's,^[^/]*/,,' | tr '[A-Z]' '[a-z]'\`echo ${UNAME_RELEASE}|sed -e 's/[-
(].*//'\`-gnu
    exit ;;
i*86:Minix*:*:*)
    echo ${UNAME_MACHINE}-pc-minix
    exit ;;
aarch64:Linux*:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-gnu

```



```

    exit ;;
aarch64_be:Linux:*:*)
    UNAME_MACHINE=aarch64_be
    echo ${UNAME_MACHINE}-unknown-linux-gnu
    exit ;;
alpha:Linux:*:*)
    case `sed -n '/^cpu model/s/^.*: \(.*\)/\1/p' < /proc/cpuinfo` in
        EV5)    UNAME_MACHINE=alphaev5 ;;
        EV56)   UNAME_MACHINE=alphaev56 ;;
        PCA56)  UNAME_MACHINE=alphapca56 ;;
        PCA57)  UNAME_MACHINE=alphapca56 ;;
        EV6)    UNAME_MACHINE=alphaev6 ;;
        EV67)   UNAME_MACHINE=alphaev67 ;;
        EV68*)  UNAME_MACHINE=alphaev68 ;;
    esac
    objdump --private-headers /bin/sh | grep -q ld.so.1
    if test "$?" = 0 ; then LIBC="gnulibc1" ; else LIBC="" ; fi
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
arm*:Linux:*:*)
    eval $set_cc_for_build
    if echo __ARM_EABI__ | $CC_FOR_BUILD -E - 2>/dev/null \
        | grep -q __ARM_EABI__
    then
        echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    else
        if echo __ARM_PCS_VFP | $CC_FOR_BUILD -E - 2>/dev/null \
            | grep -q __ARM_PCS_VFP
        then
            echo ${UNAME_MACHINE}-unknown-linux-gnueabi
        else
            echo ${UNAME_MACHINE}-unknown-linux-gnueabihf
        fi
    fi
    exit ;;
avr32*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
cris:Linux:*:*)
    echo ${UNAME_MACHINE}-axis-linux-${LIBC}
    exit ;;
crisv32:Linux:*:*)
    echo ${UNAME_MACHINE}-axis-linux-${LIBC}
    exit ;;
frv:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
hexagon:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
i*86:Linux:*:*)
    LIBC=gnu

```

```

eval `set_cc_for_build
sed 's/^ // ' << EOF >>$dummy.c
#ifdef __dietlibc__
LIBC=dietlibc
#endif
EOF
eval `set_cc_for_build -E $dummy.c 2>/dev/null | grep '^LIBC'`
echo "${UNAME_MACHINE}-pc-linux-${LIBC}"
exit ;;
ia64:Linux:*:*)
echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
exit ;;
m32r*:Linux:*:*)
echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
exit ;;
m68*:Linux:*:*)
echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
exit ;;
mips:Linux:*:* | mips64:Linux:*:*)
eval `set_cc_for_build
sed 's/^ // ' << EOF >>$dummy.c
#undef CPU
#undef ${UNAME_MACHINE}
#undef ${UNAME_MACHINE}el
#if defined(__MIPSEL__) || defined(__MIPSEL) || defined(_MIPSEL)
|| defined(MIPSEL)
CPU=${UNAME_MACHINE}el
#else
#if defined(__MIPSEB__) || defined(__MIPSEB) || defined(_MIPSEB)
|| defined(MIPSEB)
CPU=${UNAME_MACHINE}
#else
CPU=
#endif
#endif
EOF
test x"${CPU}" != x && { echo "${CPU}-unknown-linux-${LIBC}";
exit; }
;;
or32:Linux:*:*)
echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
exit ;;
padre:Linux:*:*)
echo sparc-unknown-linux-${LIBC}
exit ;;
parisc64:Linux:*:* | hppa64:Linux:*:*)
echo hppa64-unknown-linux-${LIBC}
exit ;;
parisc:Linux:*:* | hppa:Linux:*:*)
# Look for CPU level

```

```

case `grep '^cpu^[a-z]*:' /proc/cpuinfo 2>/dev/null | cut -d' ' -
f2` in
    PA7*) echo hppa1.1-unknown-linux-${LIBC} ;;
    PA8*) echo hppa2.0-unknown-linux-${LIBC} ;;
    *)    echo hppa-unknown-linux-${LIBC} ;;
esac
exit ;;
ppc64:Linux:*:*)
    echo powerpc64-unknown-linux-${LIBC}
    exit ;;
ppc:Linux:*:*)
    echo powerpc-unknown-linux-${LIBC}
    exit ;;
s390:Linux:*:* | s390x:Linux:*:*)
    echo ${UNAME_MACHINE}-ibm-linux
    exit ;;
sh64*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
sh*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
sparc:Linux:*:* | sparc64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
tile*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
vax:Linux:*:*)
    echo ${UNAME_MACHINE}-dec-linux-${LIBC}
    exit ;;
x86_64:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
xtensa*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-${LIBC}
    exit ;;
i*86:DYNIX/ptx:4*:*)
    # ptx 4.0 does uname -s correctly, with DYNIX/ptx in there.
    # earlier versions are messed up and put the nodename in both
    # sysname and nodename.
    echo i386-sequent-sysv4
    exit ;;
i*86:UNIX_SV:4.2MP:2.*)
    # Unixware is an offshoot of SVR4, but it has its own version
    # number series starting with 2...
    # I am not positive that other SVR4 systems won't match this,
    # I just have to hope. -- rms.
    # Use sysv4.2uw... so that sysv4* matches it.
    echo ${UNAME_MACHINE}-pc-sysv4.2uw${UNAME_VERSION}
    exit ;;
i*86:OS/2:*:*)

```

```

# If we were able to find `uname`, then EMX Unix compatibility
# is probably installed.
echo ${UNAME_MACHINE}-pc-os2-emx
exit ;;
i*86:XTS-300:*:*STOP)
echo ${UNAME_MACHINE}-unknown-stop
exit ;;
i*86:atheos:*:**)
echo ${UNAME_MACHINE}-unknown-atheos
exit ;;
i*86:syllable:*:**)
echo ${UNAME_MACHINE}-pc-syllable
exit ;;
i*86:LynxOS:2.*:* | i*86:LynxOS:3.[01]*:* | i*86:LynxOS:4.[02]*:*
echo i386-unknown-lynxos${UNAME_RELEASE}
exit ;;
i*86:*DOS:*:**)
echo ${UNAME_MACHINE}-pc-msdosdjgpp
exit ;;
i*86:*:4.*:* | i*86:SYSTEM_V:4.*:*
UNAME_REL=`echo ${UNAME_RELEASE} | sed 's/\\MP$//'\`
if grep Novell /usr/include/link.h >/dev/null 2>/dev/null; then
echo ${UNAME_MACHINE}-univel-sysv${UNAME_REL}
else
echo ${UNAME_MACHINE}-pc-sysv${UNAME_REL}
fi
exit ;;
i*86:*:5:[678]*)
# UnixWare 7.x, OpenUNIX and OpenServer 6.
case ` /bin/uname -X | grep "^Machine" ` in
*486*) UNAME_MACHINE=i486 ;;
*Pentium) UNAME_MACHINE=i586 ;;
*Pent*|*Celeron) UNAME_MACHINE=i686 ;;
esac
echo ${UNAME_MACHINE}-unknown-
sysv${UNAME_RELEASE}${UNAME_SYSTEM}${UNAME_VERSION}
exit ;;
i*86:*:3.2:*)
if test -f /usr/options/cb.name; then
UNAME_REL=`sed -n 's/. *Version //p' </usr/options/cb.name`
echo ${UNAME_MACHINE}-pc-isc$UNAME_REL
elif /bin/uname -X 2>/dev/null >/dev/null ; then
UNAME_REL=`(/bin/uname -X|grep Release|sed -e 's/. *= //' )`
(/bin/uname -X|grep i80486 >/dev/null) &&
UNAME_MACHINE=i486
(/bin/uname -X|grep '^Machine.*Pentium' >/dev/null) \
&& UNAME_MACHINE=i586
(/bin/uname -X|grep '^Machine.*Pent *II' >/dev/null) \
&& UNAME_MACHINE=i686
(/bin/uname -X|grep '^Machine.*Pentium Pro' >/dev/null) \
&& UNAME_MACHINE=i686
echo ${UNAME_MACHINE}-pc-sco$UNAME_REL

```

```

else
    echo ${UNAME_MACHINE}-pc-sysv32
fi
exit ;;
pc:*:*:*)
# Left here for compatibility:
# uname -m prints for DJGPP always 'pc', but it prints nothing
about
# the processor, so we play safe by assuming i586.
# Note: whatever this is, it MUST be the same as what config.sub
# prints for the "djgpp" host, or else GDB configury will decide
that
# this is a cross-build.
echo i586-pc-msdosdjgpp
exit ;;
Intel:Mach:3*:**)
echo i386-pc-mach3
exit ;;
paragon:*:*:*)
echo i860-intel-osf1
exit ;;
i860:*:4.*:**) # i860-SVR4
if grep Stardent /usr/include/sys/uadmin.h >/dev/null 2>&1 ; then
    echo i860-stardent-sysv${UNAME_RELEASE} # Stardent Vistra i860-
SVR4
else # Add other i860-SVR4 vendors below as they are discovered.
    echo i860-unknown-sysv${UNAME_RELEASE} # Unknown i860-SVR4
fi
exit ;;
mini*:CTIX:SYS*5:*)
# "miniframe"
echo m68010-convergent-sysv
exit ;;
mc68k:UNIX:SYSTEM5:3.51m)
echo m68k-convergent-sysv
exit ;;
M680?0:D-NIX:5.3:*)
echo m68k-diab-dnix
exit ;;
M68*:*:R3V[5678]*:*)
test -r /sysV68 && { echo 'm68k-motorola-sysv'; exit; } ;;
3[345]??:*:4.0:3.0 | 3[34]??A:*:4.0:3.0 | 3[34]??,*:*:4.0:3.0 |
3[34]??/*:*:4.0:3.0 | 4400:*:4.0:3.0 | 4850:*:4.0:3.0 |
SKA40:*:4.0:3.0 | SDS2:*:4.0:3.0 | SHG2:*:4.0:3.0 | S7501*:*:4.0:3.0)
OS_REL=''
test -r /etc/.releid \
&& OS_REL=`sed -n 's/[^ ]* [^ ]* \([0-9][0-9]\)\.*/\1/p' <
/etc/.releid`
/bin/uname -p 2>/dev/null | grep 86 >/dev/null \
&& { echo i486-ncr-sysv4.3${OS_REL}; exit; }
/bin/uname -p 2>/dev/null | /bin/grep entium >/dev/null \
&& { echo i586-ncr-sysv4.3${OS_REL}; exit; } ;;

```

```

3[34]?:*:4.0:* | 3[34]??,?:*:4.0:*)
  /bin/uname -p 2>/dev/null | grep 86 >/dev/null \
    && { echo i486-ncr-sysv4; exit; } ;;
NCR?:*:4.2:* | MPRAS?:*:4.2:*)
  OS_REL='.3'
  test -r /etc/.relid \
    && OS_REL=`sed -n 's/[^ ]* [^ ]* \([0-9][0-9]\)\.*/\1/p' <
/etc/.relid`
  /bin/uname -p 2>/dev/null | grep 86 >/dev/null \
    && { echo i486-ncr-sysv4.3${OS_REL}; exit; }
  /bin/uname -p 2>/dev/null | /bin/grep entium >/dev/null \
    && { echo i586-ncr-sysv4.3${OS_REL}; exit; }
  /bin/uname -p 2>/dev/null | /bin/grep pteron >/dev/null \
    && { echo i586-ncr-sysv4.3${OS_REL}; exit; } ;;
m68*:LynxOS:2.*:* | m68*:LynxOS:3.0*:**)
  echo m68k-unknown-lynxos${UNAME_RELEASE}
  exit ;;
mc68030:UNIX_System_V:4.*:**)
  echo m68k-atari-sysv4
  exit ;;
TSUNAMI:LynxOS:2.*:**)
  echo sparc-unknown-lynxos${UNAME_RELEASE}
  exit ;;
rs6000:LynxOS:2.*:**)
  echo rs6000-unknown-lynxos${UNAME_RELEASE}
  exit ;;
PowerPC:LynxOS:2.*:* | PowerPC:LynxOS:3.[01]*:* |
PowerPC:LynxOS:4.[02]*:**)
  echo powerpc-unknown-lynxos${UNAME_RELEASE}
  exit ;;
SM[BE]S:UNIX_SV:*:**)
  echo mips-dde-sysv${UNAME_RELEASE}
  exit ;;
RM*:ReliantUNIX-*:*:**)
  echo mips-sni-sysv4
  exit ;;
RM*:SINIX-*:*:**)
  echo mips-sni-sysv4
  exit ;;
*:SINIX-*:*:**)
  if uname -p 2>/dev/null >/dev/null ; then
    UNAME_MACHINE=`(uname -p) 2>/dev/null`
    echo ${UNAME_MACHINE}-sni-sysv4
  else
    echo ns32k-sni-sysv
  fi
  exit ;;
PENTIUM?:*:4.0*:**) # Unisys `ClearPath HMP IX 4000' SVR4/MP effort
  # says <Richard.M.Bartel@ccMail.Census.GOV>
  echo i586-unisys-sysv4
  exit ;;
*:UNIX_System_V:4*:FTX*)

```

```

# From Gerald Hewes <hewes@openmarket.com>.
# How about differentiating between stratus architectures? -djm
echo hppal.1-stratus-sysv4
exit ;;
*:*:*:FTX*)
# From seanf@swdc.stratus.com.
echo i860-stratus-sysv4
exit ;;
i*86:VOS:*:*)
# From Paul.Green@stratus.com.
echo ${UNAME_MACHINE}-stratus-vos
exit ;;
*:VOS:*:*)
# From Paul.Green@stratus.com.
echo hppal.1-stratus-vos
exit ;;
mc68*:A/UX:*:*)
echo m68k-apple-aux${UNAME_RELEASE}
exit ;;
news*:NEWS-OS:6:*:*)
echo mips-sony-newsos6
exit ;;
R[34]000:*System_V*:*:* | R4000:UNIX_SYSV:*:* | R*000:UNIX_SV:*:*)
if [ -d /usr/nec ]; then
    echo mips-nec-sysv${UNAME_RELEASE}
else
    echo mips-unknown-sysv${UNAME_RELEASE}
fi
exit ;;
BeBox:BeOS:*:*) # BeOS running on hardware made by Be, PPC only.
echo powerpc-be-beos
exit ;;
BeMac:BeOS:*:*) # BeOS running on Mac or Mac clone, PPC only.
echo powerpc-apple-beos
exit ;;
BePC:BeOS:*:*) # BeOS running on Intel PC compatible.
echo i586-pc-beos
exit ;;
BePC:Haiku:*:*) # Haiku running on Intel PC compatible.
echo i586-pc-haiku
exit ;;
x86_64:Haiku:*:*)
echo x86_64-unknown-haiku
exit ;;
SX-4:SUPER-UX:*:*)
echo sx4-nec-superux${UNAME_RELEASE}
exit ;;
SX-5:SUPER-UX:*:*)
echo sx5-nec-superux${UNAME_RELEASE}
exit ;;
SX-6:SUPER-UX:*:*)
echo sx6-nec-superux${UNAME_RELEASE}

```

```

    exit ;;
SX-7: SUPER-UX: *: *)
    echo sx7-nec-superux${UNAME_RELEASE}
    exit ;;
SX-8: SUPER-UX: *: *)
    echo sx8-nec-superux${UNAME_RELEASE}
    exit ;;
SX-8R: SUPER-UX: *: *)
    echo sx8r-nec-superux${UNAME_RELEASE}
    exit ;;
Power*: Rhapsody: *: *)
    echo powerpc-apple-rhapsody${UNAME_RELEASE}
    exit ;;
*: Rhapsody: *: *)
    echo ${UNAME_MACHINE}-apple-rhapsody${UNAME_RELEASE}
    exit ;;
*: Darwin: *: *)
    UNAME_PROCESSOR=`uname -p` || UNAME_PROCESSOR=unknown
    case $UNAME_PROCESSOR in
        i386)
            eval $set_cc_for_build
            if [ "$CC_FOR_BUILD" != 'no_compiler_found' ]; then
                if (echo '#ifdef __LP64__'; echo IS_64BIT_ARCH; echo
'#endif') | \
                    (CCOPTS= $CC_FOR_BUILD -E - 2>/dev/null) | \
                    grep IS_64BIT_ARCH >/dev/null
                then
                    UNAME_PROCESSOR="x86_64"
                fi
            fi ;;
        unknown) UNAME_PROCESSOR=powerpc ;;
    esac
    echo ${UNAME_PROCESSOR}-apple-darwin${UNAME_RELEASE}
    exit ;;
*: procnto*: *: * | *: QNX: [0123456789]*: *)
    UNAME_PROCESSOR=`uname -p`
    if test "$UNAME_PROCESSOR" = "x86"; then
        UNAME_PROCESSOR=i386
        UNAME_MACHINE=pc
    fi
    echo ${UNAME_PROCESSOR}-${UNAME_MACHINE}-nto-qnx${UNAME_RELEASE}
    exit ;;
*: QNX: *: 4*)
    echo i386-pc-qnx
    exit ;;
NEO-?: NONSTOP_KERNEL: *: *)
    echo neo-tandem-nsk${UNAME_RELEASE}
    exit ;;
NSE-?: NONSTOP_KERNEL: *: *)
    echo nse-tandem-nsk${UNAME_RELEASE}
    exit ;;
NSR-?: NONSTOP_KERNEL: *: *)

```



```

    echo nsr-tandem-nsk${UNAME_RELEASE}
    exit ;;
*:NonStop-UX:*:*)
    echo mips-compaq-nonstopux
    exit ;;
BS2000:POSIX*:*:*)
    echo bs2000-siemens-sysv
    exit ;;
DS/*:UNIX_System_V:*:*)
    echo ${UNAME_MACHINE}-${UNAME_SYSTEM}-${UNAME_RELEASE}
    exit ;;
*:Plan9:*:*)
    # "uname -m" is not consistent, so use $cputype instead. 386
    # is converted to i386 for consistency with other x86
    # operating systems.
    if test "$cputype" = "386"; then
        UNAME_MACHINE=i386
    else
        UNAME_MACHINE="$cputype"
    fi
    echo ${UNAME_MACHINE}-unknown-plan9
    exit ;;
*:TOPS-10:*:*)
    echo pdp10-unknown-tops10
    exit ;;
*:TENEX:*:*)
    echo pdp10-unknown-tenex
    exit ;;
KS10:TOPS-20:*:* | KL10:TOPS-20:*:* | TYPE4:TOPS-20:*:*)
    echo pdp10-dec-tops20
    exit ;;
XKL-1:TOPS-20:*:* | TYPE5:TOPS-20:*:*)
    echo pdp10-xkl-tops20
    exit ;;
*:TOPS-20:*:*)
    echo pdp10-unknown-tops20
    exit ;;
*:ITS:*:*)
    echo pdp10-unknown-its
    exit ;;
SEI:*:*:SEIUX)
    echo mips-sei-seiux${UNAME_RELEASE}
    exit ;;
*:DragonFly:*:*)
    echo ${UNAME_MACHINE}-unknown-dragonfly`echo ${UNAME_RELEASE}|sed
-e 's/[-(].*//'`
    exit ;;
*:*VMS:*:*)
    UNAME_MACHINE=`(uname -p) 2>/dev/null`
    case "${UNAME_MACHINE}" in
        A*) echo alpha-dec-vms ; exit ;;
        I*) echo ia64-dec-vms ; exit ;;

```

```

        V*) echo vax-dec-vms ; exit ;;
    esac ;;
*:XENIX:*:SysV)
    echo i386-pc-xenix
    exit ;;
i*86:skyos:*:*)
    echo ${UNAME_MACHINE}-pc-skyos`echo ${UNAME_RELEASE}` | sed -e
's/ .*$//'
    exit ;;
i*86:rdos:*:*)
    echo ${UNAME_MACHINE}-pc-rdos
    exit ;;
i*86:AROS:*:*)
    echo ${UNAME_MACHINE}-pc-aros
    exit ;;
x86_64:VMkernel:*:*)
    echo ${UNAME_MACHINE}-unknown-esx
    exit ;;
esac

eval $set_cc_for_build
cat >$dummy.c <<EOF
#ifdef _SEQUENT_
# include <sys/types.h>
# include <sys/utsname.h>
#endif
main ()
{
#ifdef (sony)
#ifdef (MIPSEB)
    /* BFD wants "bsd" instead of "newsos".  Perhaps BFD should be
    changed,
    I don't know.... */
    printf ("mips-sony-bsd\n"); exit (0);
#else
#include <sys/param.h>
    printf ("m68k-sony-newsos%s\n",
#ifdef NEWSOS4
        "4"
#else
        ""
#endif
    ); exit (0);
#endif
#endif

#ifdef (__arm) && defined (__acorn) && defined (__unix)
    printf ("arm-acorn-riscix\n"); exit (0);
#endif

#ifdef (hp300) && !defined (hpux)
    printf ("m68k-hp-bsd\n"); exit (0);

```

```

#endif

#if defined (NeXT)
#if !defined (__ARCHITECTURE__)
#define __ARCHITECTURE__ "m68k"
#endif
int version;
version=`(hostinfo | sed -n 's/.*NeXT Mach \([0-9]*\)*/\1/p')
2>/dev/null`;
if (version < 4)
    printf ("%s-next-nextstep%d\n", __ARCHITECTURE__, version);
else
    printf ("%s-next-openstep%d\n", __ARCHITECTURE__, version);
exit (0);
#endif

#if defined (MULTIMAX) || defined (n16)
#if defined (UMAXV)
    printf ("ns32k-encore-sysv\n"); exit (0);
#else
#if defined (CMU)
    printf ("ns32k-encore-mach\n"); exit (0);
#else
    printf ("ns32k-encore-bsd\n"); exit (0);
#endif
#endif
#endif

#if defined (__386BSD__)
    printf ("i386-pc-bsd\n"); exit (0);
#endif

#if defined (sequent)
#if defined (i386)
    printf ("i386-sequent-dynix\n"); exit (0);
#endif
#if defined (ns32000)
    printf ("ns32k-sequent-dynix\n"); exit (0);
#endif
#endif

#if defined (__SEQUENT_)
    struct utsname un;

    uname(&un);

    if (strncmp(un.version, "V2", 2) == 0) {
        printf ("i386-sequent-ptx2\n"); exit (0);
    }
    if (strncmp(un.version, "V1", 2) == 0) { /* XXX is V1 correct? */
        printf ("i386-sequent-ptx1\n"); exit (0);
    }
}

```

```

    printf ("i386-sequent-ptx\n"); exit (0);

#endif

#if defined (vax)
# if !defined (ultrix)
# include <sys/param.h>
# if defined (BSD)
#   if BSD == 43
#     printf ("vax-dec-bsd4.3\n"); exit (0);
#   else
#     if BSD == 199006
#       printf ("vax-dec-bsd4.3reno\n"); exit (0);
#     else
#       printf ("vax-dec-bsd\n"); exit (0);
#     endif
#   endif
# else
#     printf ("vax-dec-bsd\n"); exit (0);
#   endif
# else
#     printf ("vax-dec-ultrix\n"); exit (0);
#   endif
#endif

#if defined (alliant) && defined (i860)
    printf ("i860-alliant-bsd\n"); exit (0);
#endif

    exit (1);
}
EOF

$CC_FOR_BUILD -o $dummy $dummy.c 2>/dev/null && SYSTEM_NAME=`$dummy`
&&
    { echo "$SYSTEM_NAME"; exit; }

# Apollos put the system type in the environment.

test -d /usr/apollo && { echo ${ISP}-apollo-${SYSTYPE}; exit; }

# Convex versions that predate uname can use getsysinfo(1)

if [ -x /usr/convex/getsysinfo ]
then
    case `getsysinfo -f cpu_type` in
    c1*)
        echo c1-convex-bsd
        exit ;;
    c2*)
        if getsysinfo -f scalar_acc
        then echo c32-convex-bsd

```

```

        else echo c2-convex-bsd
        fi
        exit ;;
c34*)
        echo c34-convex-bsd
        exit ;;
c38*)
        echo c38-convex-bsd
        exit ;;
c4*)
        echo c4-convex-bsd
        exit ;;
esac
fi

```

```

cat >&2 <<EOF
$0: unable to guess system type

```

This script, last modified \$timestamp, has failed to recognize the operating system you are using. It is advised that you download the most up to date version of the config scripts from

http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.guess;hb=HEAD
and

http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.sub;hb=HEAD

If the version you run (\$0) is already up to date, please send the following data and any information you think might be pertinent to <config-patches@gnu.org> in order to provide the needed information to handle your system.

```
config.guess timestamp = $timestamp
```

```

uname -m = `(uname -m) 2>/dev/null || echo unknown`
uname -r = `(uname -r) 2>/dev/null || echo unknown`
uname -s = `(uname -s) 2>/dev/null || echo unknown`
uname -v = `(uname -v) 2>/dev/null || echo unknown`

```

```

/usr/bin/uname -p = `(/usr/bin/uname -p) 2>/dev/null`
/bin/uname -X     = `(/bin/uname -X) 2>/dev/null`

```

```

hostinfo          = `(hostinfo) 2>/dev/null`
/bin/universe     = `(/bin/universe) 2>/dev/null`
/usr/bin/arch -k  = `(/usr/bin/arch -k) 2>/dev/null`
/bin/arch         = `(/bin/arch) 2>/dev/null`
/usr/bin/oslevel  = `(/usr/bin/oslevel) 2>/dev/null`
/usr/convex/getsysinfo = `(/usr/convex/getsysinfo) 2>/dev/null`

```

```
UNAME_MACHINE = ${UNAME_MACHINE}
UNAME_RELEASE = ${UNAME_RELEASE}
UNAME_SYSTEM = ${UNAME_SYSTEM}
UNAME_VERSION = ${UNAME_VERSION}
EOF
```

```
exit 1
```

```
# Local variables:
# eval: (add-hook 'write-file-hooks 'time-stamp)
# time-stamp-start: "timestamp="
# time-stamp-format: "%:y-%02m-%02d"
# time-stamp-end: ""
# End:
```

```
File = config.h
```

```
/* config.h. Generated from config.h.in by configure. */
/* config.h.in. Generated from configure.ac by autoheader. */

/* Enable bash completion */
#define DBUS_BASH_COMPLETION 1

/* Build test code */
/* #undef DBUS_BUILD_TESTS */

/* Disable assertion checking */
#define DBUS_DISABLE_ASSERT 1

/* Disable public API sanity checking */
/* #undef DBUS_DISABLE_CHECKS */

/* Support a verbose mode */
/* #undef DBUS_ENABLE_VERBOSE_MODE */

/* Defined to the gcc version if gcov is enabled, to force a rebuild
due to
config.h changing */
/* #undef DBUS_GCOV_ENABLED */

/* Where to put test sockets */
#define DBUS_TEST_SOCKET_DIR ""

/* Disable GLib assertion macros */
#define G_DISABLE_ASSERT 1

/* Disable GLib public API sanity checking */
/* #undef G_DISABLE_CHECKS */

/* Have abstract socket namespace */
```

```
#define HAVE_ABSTRACT_SOCKETS 1

/* Define to 1 if you have the <dlfcn.h> header file. */
#define HAVE_DLFCN_H 1

/* Define to 1 if you have the <expat.h> header file. */
#define HAVE_EXPAT_H 1

/* Define to 1 if you have the <inttypes.h> header file. */
#define HAVE_INTTYPES_H 1

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Have socklen_t type */
#define HAVE_SOCKLEN_T 1

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE_STDINT_H 1

/* Define to 1 if you have the <stdlib.h> header file. */
#define HAVE_STDLIB_H 1

/* Define to 1 if you have the <strings.h> header file. */
#define HAVE_STRINGS_H 1

/* Define to 1 if you have the <string.h> header file. */
#define HAVE_STRING_H 1

/* Define to 1 if you have the <sys/stat.h> header file. */
#define HAVE_SYS_STAT_H 1

/* Define to 1 if you have the <sys/types.h> header file. */
#define HAVE_SYS_TYPES_H 1

/* Define to 1 if you have the <unistd.h> header file. */
#define HAVE_UNISTD_H 1

/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#define LT_OBJDIR ".libs/"

/* Name of package */
#define PACKAGE "dbus-glib"

/* Define to the address where bug reports for this package should be
sent. */
#define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLi
b"
```

```
/* Define to the full name of this package. */
#define PACKAGE_NAME "dbus-glib"

/* Define to the full name and version of this package. */
#define PACKAGE_STRING "dbus-glib 0.100.2"

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME "dbus-glib"

/* Define to the home page for this package. */
#define PACKAGE_URL ""

/* Define to the version of this package. */
#define PACKAGE_VERSION "0.100.2"

/* Define to 1 if you have the ANSI C header files. */
#define STDC_HEADERS 1

/* Full path to test file test/core/test-service-glib in builddir */
#define TEST_CORE_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/core/test-service-glib"

/* Full path to test file test/test-exit in builddir */
#define TEST_EXIT_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-exit"

/* Full path to test file test/interfaces/test-service in builddir */
#define TEST_INTERFACES_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/interfaces/test-service"

/* Full path to test file test/test-segfault in builddir */
#define TEST_SEGFAULT_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-segfault"

/* Full path to test file test/test-service in builddir */
#define TEST_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-service"

/* Full path to test file test/data/valid-service-files in builddir */
#define TEST_SERVICE_DIR
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
```



```

toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/data/valid-service-files"

/* Full path to test file test/test-shell-service in builddir */
#define TEST_SHELL_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-shell-service"

/* Full path to test file test/test-sleep-forever in builddir */
#define TEST_SLEEP_FOREVER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-sleep-forever"

/* Version number of package */
#define VERSION "0.100.2"

File = config.h.cmake

/* config.h. Generated by cmake from config.h.cmake */

#ifndef _DBUS_CONFIG_H
#define _DBUS_CONFIG_H
/*****/
/* indicate that we are building with cmake */
#define DBUS_CMAKE 1

#cmakedefine HAVE_GNUC_VARARGS 1

#cmakedefine DBUS_CONSOLE_AUTH_DIR "@DBUS_CONSOLE_AUTH_DIR@"
#cmakedefine DBUS_DATADIR "@DBUS_DATADIR@"
#cmakedefine DBUS_BINDIR "@DBUS_BINDIR@"
#cmakedefine DBUS_SYSTEM_CONFIG_FILE "@DBUS_SYSTEM_CONFIG_FILE@"
#cmakedefine DBUS_SESSION_CONFIG_FILE "@DBUS_SESSION_CONFIG_FILE@"
#cmakedefine DBUS_DAEMON_NAME "@DBUS_DAEMON_NAME@"
#cmakedefine DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"@DBUS_SYSTEM_BUS_DEFAULT_ADDRESS@"
#cmakedefine DBUS_MACHINE_UUID_FILE "@DBUS_MACHINE_UUID_FILE@"
#cmakedefine DBUS_SESSION_BUS_DEFAULT_ADDRESS
"@DBUS_SESSION_BUS_DEFAULT_ADDRESS@"
#cmakedefine DBUS_DAEMONDIR "@DBUS_DAEMONDIR@"
#cmakedefine PACKAGE "@PACKAGE@"
/* Version number of package */
#cmakedefine DBUS_MAJOR_VERSION @DBUS_MAJOR_VERSION@
#cmakedefine DBUS_MINOR_VERSION @DBUS_MINOR_VERSION@
#cmakedefine DBUS_MICRO_VERSION @DBUS_MICRO_VERSION@
#cmakedefine DBUS_VERSION ((@DBUS_MAJOR_VERSION@ << 16) |
(@DBUS_MINOR_VERSION@ << 8) | (@DBUS_MICRO_VERSION@))
#cmakedefine DBUS_VERSION_STRING "@DBUS_VERSION_STRING@"

```

```

#cmakedefine DBUS_ENABLE_STATS

#define VERSION DBUS_VERSION_STRING

#define TEST_LISTEN          "@TEST_LISTEN@"

// test binaries
#define DBUS_TEST_EXEC "@DBUS_TEST_EXEC@"
#define DBUS_EXEEXT "@EXEEXT@"

/* Full path to test file test/test-exit in builddir */
#define TEST_BUS_BINARY      "@TEST_BUS_BINARY@"

/* Some dbus features */
#cmakedefine DBUS_BUILD_TESTS 1
#cmakedefine DBUS_ENABLE_ANSI 1
#cmakedefine DBUS_ENABLE_VERBOSE_MODE 1
#cmakedefine DBUS_DISABLE_ASSERTS 1
#cmakedefine DBUS_DISABLE_CHECKS 1
/* xml docs */
/* doxygen */
#cmakedefine DBUS_GCOV_ENABLED 1

/* abstract-sockets */

#cmakedefine HAVE_ABSTRACT_SOCKETS 1

#cmakedefine DBUS_PATH_OR_ABSTRACT_VALUE 1

#if (defined DBUS_PATH_OR_ABSTRACT_VALUE)
#define DBUS_PATH_OR_ABSTRACT @DBUS_PATH_OR_ABSTRACT_VALUE@
#endif

#ifdef DBUS_PATH_OR_ABSTRACT_VALUE
#undef DBUS_PATH_OR_ABSTRACT_VALUE
#endif

/* selinux */
#cmakedefine DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX 1
/* kqueue */
#cmakedefine HAVE_CONSOLE_OWNER_FILE 1
#define DBUS_CONSOLE_OWNER_FILE "@DBUS_CONSOLE_OWNER_FILE@"

#cmakedefine DBUS_HAVE_ATOMIC_INT 1
#cmakedefine DBUS_USE_ATOMIC_INT_486 1
#if (defined(__i386__) || defined(__x86_64__))
# define DBUS_HAVE_ATOMIC_INT 1
# define DBUS_USE_ATOMIC_INT_486 1
#endif

#cmakedefine DBUS_BUILD_X11 1

```

```

/* For the moment, the cmake build system doesn't have an equivalent
of
 * the autoconf build system's --disable-x11-autolaunch */
#ifdef DBUS_BUILD_X11
# define DBUS_ENABLE_X11_AUTOLAUNCH 1
#endif

#define _DBUS_VA_COPY_ASSIGN(a1,a2) { a1 = a2; }

#cmakedefine DBUS_VA_COPY_FUNC
#if (defined DBUS_VA_COPY_FUNC)
# define DBUS_VA_COPY @DBUS_VA_COPY_FUNC@
#endif

#ifdef DBUS_VA_COPY_FUNC
#undef DBUS_VA_COPY_FUNC
#endif

#cmakedefine DBUS_VA_COPY_AS_ARRAY @DBUS_VA_COPY_AS_ARRAY@

// headers
/* Define to 1 if you have dirent.h */
#cmakedefine HAVE_DIRENT_H 1

/* Define to 1 if you have io.h */
#cmakedefine HAVE_IO_H 1

/* Define to 1 if you have grp.h */
#cmakedefine HAVE_GRP_H 1

/* Define to 1 if you have sys/poll.h */
#cmakedefine HAVE_POLL 1

/* Define to 1 if you have sys/time.h */
#cmakedefine HAVE_SYS_TIME 1

/* Define to 1 if you have sys/wait.h */
#cmakedefine HAVE_SYS_WAIT 1

/* Define to 1 if you have time.h */
#cmakedefine HAVE_TIME_H 1

/* Define to 1 if you have ws2tcpip.h */
#cmakedefine HAVE_WS2TCPIP_H

/* Define to 1 if you have wspiapi.h */
#cmakedefine HAVE_WSPIAPI_H 1

/* Define to 1 if you have unistd.h */
#cmakedefine HAVE_UNISTD_H 1

/* Define to 1 if you have stdio.h */

```

```
#cmakedefine HAVE_STDIO_H 1

/* Define to 1 if you have sys/syslimits.h */
#cmakedefine HAVE_SYS_SYSLIMITS_H 1

/* Define to 1 if you have errno.h */
#cmakedefine HAVE_ERRNO_H 1

/* Define to 1 if you have signal.h */
#cmakedefine HAVE_SIGNAL_H 1

/* Define to 1 if you have locale.h */
#cmakedefine HAVE_LOCALE_H 1

/* Define to 1 if you have inttypes.h */
#cmakedefine HAVE_INTTYPES_H 1

/* Define to 1 if you have stdint.h */
#cmakedefine HAVE_STDINT_H 1

// symbols
/* Define to 1 if you have backtrace */
#cmakedefine HAVE_BACKTRACE 1

/* Define to 1 if you have getgrouplist */
#cmakedefine HAVE_GETGROUPLIST 1

/* Define to 1 if you have getpeerucred */
#cmakedefine HAVE_GETPEERUCRED 1

/* Define to 1 if you have nanosleep */
#cmakedefine HAVE_NANOSLEEP 1

/* Define to 1 if you have getpwnam_r */
#cmakedefine HAVE_POSIX_GETPWNAM_R 1

/* Define to 1 if you have socketpair */
#cmakedefine HAVE_SOCKETPAIR 1

/* Define to 1 if you have setenv */
#cmakedefine HAVE_SETENV 1

/* Define to 1 if you have unsetenv */
#cmakedefine HAVE_UNSETENV 1

/* Define to 1 if you have clearenv */
#cmakedefine HAVE_CLEARENV 1

/* Define to 1 if you have writev */
#cmakedefine HAVE_WRITEV 1

/* Define to 1 if you have socklen_t */
```

```

#cmakedefine    HAVE_SOCKETLEN_T 1

/* Define to 1 if you have setlocale */
#cmakedefine    HAVE_SETLOCALE 1

/* Define to 1 if you have localeconv */
#cmakedefine    HAVE_LOCALECONV 1

/* Define to 1 if you have strtoll */
#cmakedefine    HAVE_STRTOLL 1

/* Define to 1 if you have strtoull */
#cmakedefine    HAVE_STRTOULL 1

// structs
/* Define to 1 if you have struct cmsgred */
#cmakedefine    HAVE_CMSGCRED 1

// system type defines
#if defined(_WIN32) || defined(_WIN64) || defined (_WIN32_WCE)
# define DBUS_WIN
# define DBUS_WIN_FIXME 1
# ifdef _WIN32_WCE
#   define DBUS_WINCE
# else
#   define DBUS_WIN32
# endif
#else
# define DBUS_UNIX
#endif

#if defined(_WIN32) || defined(_WIN64)
// mingw mode_t
# ifdef HAVE_STDIO_H
#   include <stdio.h>
# endif
# ifndef _MSC_VER
#   define uid_t int
#   define gid_t int
# else
#   define snprintf _snprintf
#   typedef int mode_t;
#   if !defined(_WIN32_WCE)
#     define strtoll _strtoi64
#     define strtoull _strtoui64
#     define HAVE_STRTOLL 1
#     define HAVE_STRTOULL 1
#   endif
# endif
#endif // defined(_WIN32) || defined(_WIN64)

#ifdef interface

```

```

#undef interface
#endif

#ifndef SIGHUP
#define SIGHUP 1
#endif

#ifdef DBUS_VERBOSE_C_S 1
#ifdef DBUS_VERBOSE_C_S
#define _dbus_verbose_C_S printf
#else
#define _dbus_verbose_C_S _dbus_verbose
#endif
#endif

# if defined(_MSC_VER) && !defined(inline)
#define inline __inline
#endif

#endif // _DBUS_CONFIG_H

File = config.h.in

/* config.h.in. Generated from configure.ac by autoheader. */

/* Enable bash completion */
#undef DBUS_BASH_COMPLETION

/* Build test code */
#undef DBUS_BUILD_TESTS

/* Disable assertion checking */
#undef DBUS_DISABLE_ASSERT

/* Disable public API sanity checking */
#undef DBUS_DISABLE_CHECKS

/* Support a verbose mode */
#undef DBUS_ENABLE_VERBOSE_MODE

/* Defined to the gcc version if gcov is enabled, to force a rebuild
due to
    config.h changing */
#undef DBUS_GCOV_ENABLED

/* Where to put test sockets */
#undef DBUS_TEST_SOCKET_DIR

/* Disable GLib assertion macros */
#undef G_DISABLE_ASSERT

```

```
/* Disable GLib public API sanity checking */
#undef G_DISABLE_CHECKS

/* Have abstract socket namespace */
#undef HAVE_ABSTRACT_SOCKETS

/* Define to 1 if you have the <dlfcn.h> header file. */
#undef HAVE_DLFCN_H

/* Define to 1 if you have the <expat.h> header file. */
#undef HAVE_EXPAT_H

/* Define to 1 if you have the <inttypes.h> header file. */
#undef HAVE_INTTYPES_H

/* Define to 1 if you have the <memory.h> header file. */
#undef HAVE_MEMORY_H

/* Have socklen_t type */
#undef HAVE_SOCKLEN_T

/* Define to 1 if you have the <stdint.h> header file. */
#undef HAVE_STDINT_H

/* Define to 1 if you have the <stdlib.h> header file. */
#undef HAVE_STDLIB_H

/* Define to 1 if you have the <strings.h> header file. */
#undef HAVE_STRINGS_H

/* Define to 1 if you have the <string.h> header file. */
#undef HAVE_STRING_H

/* Define to 1 if you have the <sys/stat.h> header file. */
#undef HAVE_SYS_STAT_H

/* Define to 1 if you have the <sys/types.h> header file. */
#undef HAVE_SYS_TYPES_H

/* Define to 1 if you have the <unistd.h> header file. */
#undef HAVE_UNISTD_H

/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#undef LT_OBJDIR

/* Name of package */
#undef PACKAGE

/* Define to the address where bug reports for this package should be
sent. */
```

```
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package. */
#undef PACKAGE_NAME

/* Define to the full name and version of this package. */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package. */
#undef PACKAGE_TARNAME

/* Define to the home page for this package. */
#undef PACKAGE_URL

/* Define to the version of this package. */
#undef PACKAGE_VERSION

/* Define to 1 if you have the ANSI C header files. */
#undef STDC_HEADERS

/* Full path to test file test/core/test-service-glib in builddir */
#undef TEST_CORE_SERVICE_BINARY

/* Full path to test file test/test-exit in builddir */
#undef TEST_EXIT_BINARY

/* Full path to test file test/interfaces/test-service in builddir */
#undef TEST_INTERFACES_SERVICE_BINARY

/* Full path to test file test/test-segfault in builddir */
#undef TEST_SEGFAULT_BINARY

/* Full path to test file test/test-service in builddir */
#undef TEST_SERVICE_BINARY

/* Full path to test file test/data/valid-service-files in builddir */
#undef TEST_SERVICE_DIR

/* Full path to test file test/test-shell-service in builddir */
#undef TEST_SHELL_SERVICE_BINARY

/* Full path to test file test/test-sleep-forever in builddir */
#undef TEST_SLEEP_FOREVER_BINARY

/* Version number of package */
#undef VERSION

File = config.h.in.~1~

/* config.h.in.  Generated from configure.ac by autoheader.  */
```



```
/* Define if building universal (internal helper macro) */
#undef AC_APPLE_UNIVERSAL_BUILD

/* poll doesn't work on devices */
#undef BROKEN_POLL

/* Directory for installing the binaries */
#undef DBUS_BINDIR

/* Define to build test code into the library and binaries */
#undef DBUS_BUILD_TESTS

/* Define to build X11 functionality */
#undef DBUS_BUILD_X11

/* whether -export-dynamic was passed to libtool */
#undef DBUS_BUILT_R_DYNAMIC

/* Use dnotify on Linux */
#undef DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX

/* Use inotify */
#undef DBUS_BUS_ENABLE_INOTIFY

/* Use kqueue */
#undef DBUS_BUS_ENABLE_KQUEUE

/* Directory to check for console ownership */
#undef DBUS_CONSOLE_AUTH_DIR

/* File to check for console ownership */
#undef DBUS_CONSOLE_OWNER_FILE

/* Defined if we run on a cygwin API based system */
#undef DBUS_CYGWIN

/* Directory for installing the DBUS daemon */
#undef DBUS_DAEMONDIR

/* Name of executable */
#undef DBUS_DAEMON_NAME

/* Directory for installing DBUS data files */
#undef DBUS_DATADIR

/* Disable assertion checking */
#undef DBUS_DISABLE_ASSERT

/* Disable public API sanity checking */
#undef DBUS_DISABLE_CHECKS
```

```
/* Define to build test code into the library and binaries */
#undef DBUS_ENABLE_EMBEDDED_TESTS

/* Use launchd autolaunch */
#undef DBUS_ENABLE_LAUNCHD

/* Define to build independent test binaries */
#undef DBUS_ENABLE_MODULAR_TESTS

/* Define to enable bus daemon usage statistics */
#undef DBUS_ENABLE_STATS

/* Build with caching of user data */
#undef DBUS_ENABLE_USERDB_CACHE

/* Support a verbose mode */
#undef DBUS_ENABLE_VERBOSE_MODE

/* Define to enable X11 auto-launch */
#undef DBUS_ENABLE_X11_AUTOLAUNCH

/* Extension for executables, typically empty or .exe */
#undef DBUS_EXEEXT

/* Defined if gcov is enabled to force a rebuild due to config.h
changing */
#undef DBUS_GCOV_ENABLED

/* Define to use epoll(4) on Linux */
#undef DBUS_HAVE_LINUX_EPOLL

/* Define to printf modifier for 64 bit integer type */
#undef DBUS_INT64_PRINTF_MODIFIER

/* Directory for installing the libexec binaries */
#undef DBUS_LIBEXECDIR

/* Prefix for installing DBUS */
#undef DBUS_PREFIX

/* Where per-session bus puts its sockets */
#undef DBUS_SESSION_SOCKET_DIR

/* The default D-Bus address of the system bus */
#undef DBUS_SYSTEM_BUS_DEFAULT_ADDRESS

/* The name of the socket the system bus listens on by default */
#undef DBUS_SYSTEM_SOCKET

/* Full path to the daemon in the builddir */
#undef DBUS_TEST_EXEC
```

```
/* Full path to the launch helper test program in the builddir */
#undef DBUS_TEST_LAUNCH_HELPER_BINARY

/* Where to put test sockets */
#undef DBUS_TEST_SOCKET_DIR

/* Defined if we run on a Unix-based system */
#undef DBUS_UNIX

/* User for running the system BUS daemon */
#undef DBUS_USER

/* Use the gcc __sync extension */
#undef DBUS_USE_SYNC

/* A 'va_copy' style function */
#undef DBUS_VA_COPY

/* 'va_lists' cannot be copied as values */
#undef DBUS_VA_COPY_AS_ARRAY

/* Defined if we run on a W32 API based system */
#undef DBUS_WIN

/* Defined if we run on a W32 CE API based system */
#undef DBUS_WINCE

/* Define if GLib, GObject, GIO are available */
#undef DBUS_WITH_GLIB

/* The name of the gettext domain */
#undef GETTEXT_PACKAGE

/* Disable GLib public API sanity checking */
#undef G_DISABLE_CHECKS

/* Have abstract socket namespace */
#undef HAVE_ABSTRACT_SOCKETS

/* Define to 1 if you have the `accept4' function. */
#undef HAVE_ACCEPT4

/* Adt audit API */
#undef HAVE_ADT

/* Define to 1 if you have the `backtrace' function. */
#undef HAVE_BACKTRACE

/* Define to 1 if you have the <byteswap.h> header file. */
#undef HAVE_BYTESWAP_H

/* Define to 1 if you have the `clearenv' function. */
```

```
#undef HAVE_CLEARENV

/* Have cmsgcred structure */
#undef HAVE_CMSGCRED

/* Have console owner file */
#undef HAVE_CONSOLE_OWNER_FILE

/* Define to 1 if you have the < crt_externs.h > header file. */
#undef HAVE_CRT_EXTERNS_H

/* Have the ddfd member of DIR */
#undef HAVE_DDFD

/* Define to 1 if you have the declaration of `LOG_PERROR', and to 0
if you
    don't. */
#undef HAVE_DECL_LOG_PERROR

/* Define to 1 if you have the declaration of `MSG_NOSIGNAL', and to 0
if you
    don't. */
#undef HAVE_DECL_MSG_NOSIGNAL

/* Define to 1 if you have the < dirent.h > header file. */
#undef HAVE_DIRENT_H

/* Have dirfd function */
#undef HAVE_DIRFD

/* Define to 1 if you have the < dlfcn.h > header file. */
#undef HAVE_DLFCN_H

/* Define to 1 if you have the < errno.h > header file. */
#undef HAVE_ERRNO_H

/* Define to 1 if you have the < execinfo.h > header file. */
#undef HAVE_EXECINFO_H

/* Define to 1 if you have the < expat.h > header file. */
#undef HAVE_EXPAT_H

/* Define to 1 if you have the `fpathconf' function. */
#undef HAVE_FPATHCONF

/* Define to 1 if you have the `getgrouplist' function. */
#undef HAVE_GETGROUPLIST

/* Define to 1 if you have the `getpeereid' function. */
#undef HAVE_GETPEEREID

/* Define to 1 if you have the `getpeerucred' function. */
```

```
#undef HAVE_GETPEERUCRED

/* Define to 1 if you have the `getresuid' function. */
#undef HAVE_GETRESUID

/* Have GNU-style varargs macros */
#undef HAVE_GNUC_VARARGS

/* Define to 1 if you have the `inotify_init1' function. */
#undef HAVE_INOTIFY_INIT1

/* Define to 1 if you have the <inttypes.h> header file. */
#undef HAVE_INTTYPES_H

/* Have ISO C99 varargs macros */
#undef HAVE_ISO_VARARGS

/* Define to 1 if you have the `issetugid' function. */
#undef HAVE_ISSETUGID

/* audit daemon SELinux support */
#undef HAVE_LIBAUDIT

/* Define to 1 if you have the `nsl' library (-lnsl). */
#undef HAVE_LIBNSL

/* Define to 1 if you have the `localeconv' function. */
#undef HAVE_LOCALECONV

/* Define to 1 if you have the <locale.h> header file. */
#undef HAVE_LOCALE_H

/* Define to 1 if you have the <memory.h> header file. */
#undef HAVE_MEMORY_H

/* Define if we have CLOCK_MONOTONIC */
#undef HAVE_MONOTONIC_CLOCK

/* Define to 1 if you have the `nanosleep' function. */
#undef HAVE_NANOSLEEP

/* Have non-POSIX function getpwnam_r */
#undef HAVE_NONPOSIX_GETPWNAM_R

/* Define if your system needs _NSGetEnviron to set up the environment
*/
#undef HAVE_NSGETENVIRON

/* Define to 1 if you have the `pipe2' function. */
#undef HAVE_PIPE2

/* Define to 1 if you have the `poll' function. */
```

```
#undef HAVE_POLL

/* Have POSIX function getpwnam_r */
#undef HAVE_POSIX_GETPWNAM_R

/* SELinux support */
#undef HAVE_SELINUX

/* Define to 1 if you have the `setenv' function. */
#undef HAVE_SETENV

/* Define to 1 if you have the `setlocale' function. */
#undef HAVE_SETLOCALE

/* Define to 1 if you have the `setrlimit' function. */
#undef HAVE_SETRLIMIT

/* Define to 1 if you have the <signal.h> header file. */
#undef HAVE_SIGNAL_H

/* Define to 1 if you have the `socketpair' function. */
#undef HAVE_SOCKETPAIR

/* Have socklen_t type */
#undef HAVE_SOCKLEN_T

/* Define to 1 if you have the <stdint.h> header file. */
#undef HAVE_STDINT_H

/* Define to 1 if you have the <stdlib.h> header file. */
#undef HAVE_STDLIB_H

/* Define to 1 if you have the <strings.h> header file. */
#undef HAVE_STRINGS_H

/* Define to 1 if you have the <string.h> header file. */
#undef HAVE_STRING_H

/* Define to 1 if you have the `strtoll' function. */
#undef HAVE_STRTOLL

/* Define to 1 if you have the `strtoull' function. */
#undef HAVE_STRTOULL

/* Define to 1 if you have the <syslog.h> header file. */
#undef HAVE_SYSLOG_H

/* Have systemd */
#undef HAVE_SYSTEMD

/* Define to 1 if you have the <sys/inotify.h> header file. */
#undef HAVE_SYS_INOTIFY_H
```

```
/* Define to 1 if you have the <sys/resource.h> header file. */
#undef HAVE_SYS_RESOURCE_H

/* Define to 1 if you have the <sys/stat.h> header file. */
#undef HAVE_SYS_STAT_H

/* Define to 1 if you have the <sys/syslimits.h> header file. */
#undef HAVE_SYS_SYSLIMITS_H

/* Define to 1 if you have the <sys/types.h> header file. */
#undef HAVE_SYS_TYPES_H

/* Define to 1 if you have the <sys/uio.h> header file. */
#undef HAVE_SYS_UIO_H

/* Define to 1 if you have the <unistd.h> header file. */
#undef HAVE_UNISTD_H

/* Supports sending UNIX file descriptors */
#undef HAVE_UNIX_FD_PASSING

/* Define to 1 if you have the `unsetenv' function. */
#undef HAVE_UNSETENV

/* Define to 1 if you have the `usleep' function. */
#undef HAVE_USLEEP

/* Define to 1 if you have the `vasprintf' function. */
#undef HAVE_VASPRINTF

/* Define to 1 if you have the `vsnprintf' function. */
#undef HAVE_VSNPRINTF

/* Define to 1 if you have the `writev' function. */
#undef HAVE_WRITEV

/* Define to 1 if you have the <ws2tcpip.h> header file. */
#undef HAVE_WS2TCPIP_H

/* Define to 1 if you have the <wspiapi.h> header file. */
#undef HAVE_WSPIAPI_H

/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#undef LT_OBJDIR

/* Define to 1 if your C compiler doesn't accept -c and -o together.
*/
#undef NO_MINUS_C_MINUS_O
```

```
/* Name of package */
#undef PACKAGE

/* Define to the address where bug reports for this package should be
sent. */
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package. */
#undef PACKAGE_NAME

/* Define to the full name and version of this package. */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package. */
#undef PACKAGE_TARNAME

/* Define to the home page for this package. */
#undef PACKAGE_URL

/* Define to the version of this package. */
#undef PACKAGE_VERSION

/* The size of `char', as computed by sizeof. */
#undef SIZEOF_CHAR

/* The size of `int', as computed by sizeof. */
#undef SIZEOF_INT

/* The size of `long', as computed by sizeof. */
#undef SIZEOF_LONG

/* The size of `long long', as computed by sizeof. */
#undef SIZEOF_LONG_LONG

/* The size of `short', as computed by sizeof. */
#undef SIZEOF_SHORT

/* The size of `void *', as computed by sizeof. */
#undef SIZEOF_VOID_P

/* The size of `__int64', as computed by sizeof. */
#undef SIZEOF___INT64

/* Define to 1 if you have the ANSI C header files. */
#undef STDC_HEADERS

/* Full path to the daemon in the builddir */
#undef TEST_BUS_BINARY

/* Listening address for regression tests */
#undef TEST_LISTEN
```



```

/* Enable extensions on AIX 3, Interix. */
#ifndef _ALL_SOURCE
# undef _ALL_SOURCE
#endif
/* Enable GNU extensions on systems that have them. */
#ifndef _GNU_SOURCE
# undef _GNU_SOURCE
#endif
/* Enable threading extensions on Solaris. */
#ifndef _POSIX_PTHREAD_SEMANTICS
# undef _POSIX_PTHREAD_SEMANTICS
#endif
/* Enable extensions on HP NonStop. */
#ifndef _TANDEM_SOURCE
# undef _TANDEM_SOURCE
#endif
/* Enable general extensions on Solaris. */
#ifndef __EXTENSIONS__
# undef __EXTENSIONS__
#endif

/* Version number of package */
#undef VERSION

/* Define to add Valgrind instrumentation */
#undef WITH_VALGRIND

/* Define WORDS_BIGENDIAN to 1 if your processor stores words with the
most
significant byte first (like Motorola and SPARC, unlike Intel). */
#if defined AC_APPLE_UNIVERSAL_BUILD
# if defined __BIG_ENDIAN__
#  define WORDS_BIGENDIAN 1
# endif
#else
# ifndef WORDS_BIGENDIAN
#  undef WORDS_BIGENDIAN
# endif
#endif

/* Use the compiler-provided endianness defines to
allow universal compiling. */
#ifdef __BIG_ENDIAN__
#define WORDS_BIGENDIAN 1
#endif

/* Define to 1 if the X Window System is missing or not being used. */
#undef X_DISPLAY_MISSING

```

```

#if defined(HAVE_NSGETENVIRON) && defined(HAVE_CRT_EXTERNS_H)
# include <sys/time.h>
# include <crt_extrns.h>
# define environ (*_NSGetEnviron())
#endif

/* Define to 1 if on MINIX. */
#undef _MINIX

/* Define to 2 if the system does not provide POSIX.1 features except
with
this defined. */
#undef _POSIX_1_SOURCE

/* Define to 1 if you need to in order for `stat' and other things to
work. */
#undef _POSIX_SOURCE

/* Defined to get newer W32 CE APIs */
#undef _WIN32_WCE

/* Define to `__inline__' or `__inline' if that's what the C compiler
calls it, or to nothing if 'inline' is not supported under any
name. */
#ifndef __cplusplus
#undef inline
#endif

File = config.h.in~

/* config.h.in. Generated from configure.ac by autoheader. */

/* Enable bash completion */
#undef DBUS_BASH_COMPLETION

/* Build test code */
#undef DBUS_BUILD_TESTS

/* Disable assertion checking */
#undef DBUS_DISABLE_ASSERT

/* Disable public API sanity checking */
#undef DBUS_DISABLE_CHECKS

/* Support a verbose mode */
#undef DBUS_ENABLE_VERBOSE_MODE

```

```
/* Defined to the gcc version if gcov is enabled, to force a rebuild
due to
    config.h changing */
#undef DBUS_GCOV_ENABLED

/* Where to put test sockets */
#undef DBUS_TEST_SOCKET_DIR

/* Disable GLib assertion macros */
#undef G_DISABLE_ASSERT

/* Disable GLib public API sanity checking */
#undef G_DISABLE_CHECKS

/* Have abstract socket namespace */
#undef HAVE_ABSTRACT_SOCKETS

/* Define to 1 if you have the <dlfcn.h> header file. */
#undef HAVE_DLFCN_H

/* Define to 1 if you have the <expat.h> header file. */
#undef HAVE_EXPAT_H

/* Define to 1 if you have the <inttypes.h> header file. */
#undef HAVE_INTTYPES_H

/* Define to 1 if you have the <memory.h> header file. */
#undef HAVE_MEMORY_H

/* Have socklen_t type */
#undef HAVE_SOCKLEN_T

/* Define to 1 if you have the <stdint.h> header file. */
#undef HAVE_STDINT_H

/* Define to 1 if you have the <stdlib.h> header file. */
#undef HAVE_STDLIB_H

/* Define to 1 if you have the <strings.h> header file. */
#undef HAVE_STRINGS_H

/* Define to 1 if you have the <string.h> header file. */
#undef HAVE_STRING_H

/* Define to 1 if you have the <sys/stat.h> header file. */
#undef HAVE_SYS_STAT_H

/* Define to 1 if you have the <sys/types.h> header file. */
#undef HAVE_SYS_TYPES_H

/* Define to 1 if you have the <unistd.h> header file. */
#undef HAVE_UNISTD_H
```

```
/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#undef LT_OBJDIR

/* Name of package */
#undef PACKAGE

/* Define to the address where bug reports for this package should be
sent. */
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package. */
#undef PACKAGE_NAME

/* Define to the full name and version of this package. */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package. */
#undef PACKAGE_TARNAME

/* Define to the home page for this package. */
#undef PACKAGE_URL

/* Define to the version of this package. */
#undef PACKAGE_VERSION

/* Define to 1 if you have the ANSI C header files. */
#undef STDC_HEADERS

/* Full path to test file test/core/test-service-glib in builddir */
#undef TEST_CORE_SERVICE_BINARY

/* Full path to test file test/test-exit in builddir */
#undef TEST_EXIT_BINARY

/* Full path to test file test/interfaces/test-service in builddir */
#undef TEST_INTERFACES_SERVICE_BINARY

/* Full path to test file test/test-segfault in builddir */
#undef TEST_SEGFAULT_BINARY

/* Full path to test file test/test-service in builddir */
#undef TEST_SERVICE_BINARY

/* Full path to test file test/data/valid-service-files in builddir */
#undef TEST_SERVICE_DIR

/* Full path to test file test/test-shell-service in builddir */
#undef TEST_SHELL_SERVICE_BINARY
```

```
/* Full path to test file test/test-sleep-forever in builddir */
#undef TEST_SLEEP_FOREVER_BINARY

/* Version number of package */
#undef VERSION

File = config.h.in~.~1~

/* config.h.in.  Generated from configure.ac by autoheader.  */

/* Define if building universal (internal helper macro) */
#undef AC_APPLE_UNIVERSAL_BUILD

/* poll doesn't work on devices */
#undef BROKEN_POLL

/* Directory for installing the binaries */
#undef DBUS_BINDIR

/* Define to build test code into the library and binaries */
#undef DBUS_BUILD_TESTS

/* Define to build X11 functionality */
#undef DBUS_BUILD_X11

/* whether -export-dynamic was passed to libtool */
#undef DBUS_BUILT_R_DYNAMIC

/* Use dnotify on Linux */
#undef DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX

/* Use inotify */
#undef DBUS_BUS_ENABLE_INOTIFY

/* Use kqueue */
#undef DBUS_BUS_ENABLE_KQUEUE

/* Directory to check for console ownership */
#undef DBUS_CONSOLE_AUTH_DIR

/* File to check for console ownership */
#undef DBUS_CONSOLE_OWNER_FILE

/* Defined if we run on a cygwin API based system */
#undef DBUS_CYGWIN

/* Directory for installing the DBUS daemon */
#undef DBUS_DAEMONDIR

/* Name of executable */
```

```
#undef DBUS_DAEMON_NAME

/* Directory for installing DBUS data files */
#undef DBUS_DATADIR

/* Disable assertion checking */
#undef DBUS_DISABLE_ASSERT

/* Disable public API sanity checking */
#undef DBUS_DISABLE_CHECKS

/* Define to build test code into the library and binaries */
#undef DBUS_ENABLE_EMBEDDED_TESTS

/* Use launchd autolaunch */
#undef DBUS_ENABLE_LAUNCHD

/* Define to build independent test binaries */
#undef DBUS_ENABLE_MODULAR_TESTS

/* Define to enable bus daemon usage statistics */
#undef DBUS_ENABLE_STATS

/* Build with caching of user data */
#undef DBUS_ENABLE_USERDB_CACHE

/* Support a verbose mode */
#undef DBUS_ENABLE_VERBOSE_MODE

/* Define to enable X11 auto-launch */
#undef DBUS_ENABLE_X11_AUTOLAUNCH

/* Extension for executables, typically empty or .exe */
#undef DBUS_EXEEXT

/* Defined if gcov is enabled to force a rebuild due to config.h
changing */
#undef DBUS_GCOV_ENABLED

/* Define to use epoll(4) on Linux */
#undef DBUS_HAVE_LINUX_EPOLL

/* Define to printf modifier for 64 bit integer type */
#undef DBUS_INT64_PRINTF_MODIFIER

/* Directory for installing the libexec binaries */
#undef DBUS_LIBEXECDIR

/* Prefix for installing DBUS */
#undef DBUS_PREFIX

/* Where per-session bus puts its sockets */
```

```
#undef DBUS_SESSION_SOCKET_DIR

/* The default D-Bus address of the system bus */
#undef DBUS_SYSTEM_BUS_DEFAULT_ADDRESS

/* The name of the socket the system bus listens on by default */
#undef DBUS_SYSTEM_SOCKET

/* Full path to the daemon in the builddir */
#undef DBUS_TEST_EXEC

/* Full path to the launch helper test program in the builddir */
#undef DBUS_TEST_LAUNCH_HELPER_BINARY

/* Where to put test sockets */
#undef DBUS_TEST_SOCKET_DIR

/* Defined if we run on a Unix-based system */
#undef DBUS_UNIX

/* User for running the system BUS daemon */
#undef DBUS_USER

/* Use the gcc __sync extension */
#undef DBUS_USE_SYNC

/* A 'va_copy' style function */
#undef DBUS_VA_COPY

/* 'va_lists' cannot be copied as values */
#undef DBUS_VA_COPY_AS_ARRAY

/* Defined if we run on a W32 API based system */
#undef DBUS_WIN

/* Defined if we run on a W32 CE API based system */
#undef DBUS_WINCE

/* Define if GLib, GObject, GIO are available */
#undef DBUS_WITH_GLIB

/* The name of the gettext domain */
#undef GETTEXT_PACKAGE

/* Disable GLib public API sanity checking */
#undef G_DISABLE_CHECKS

/* Have abstract socket namespace */
#undef HAVE_ABSTRACT_SOCKETS

/* Define to 1 if you have the `accept4' function. */
#undef HAVE_ACCEPT4
```

```
/* Adt audit API */
#undef HAVE_ADT

/* Define to 1 if you have the `backtrace' function. */
#undef HAVE_BACKTRACE

/* Define to 1 if you have the <byteswap.h> header file. */
#undef HAVE_BYTESWAP_H

/* Define to 1 if you have the `clearenv' function. */
#undef HAVE_CLEARENV

/* Have msgcred structure */
#undef HAVE_MSGCRED

/* Have console owner file */
#undef HAVE_CONSOLE_OWNER_FILE

/* Define to 1 if you have the < crt_externs.h> header file. */
#undef HAVE_CRT_EXTERNS_H

/* Have the ddfid member of DIR */
#undef HAVE_DDFID

/* Define to 1 if you have the declaration of `LOG_PERROR', and to 0
if you
   don't. */
#undef HAVE_DECL_LOG_PERROR

/* Define to 1 if you have the declaration of `MSG_NOSIGNAL', and to 0
if you
   don't. */
#undef HAVE_DECL_MSG_NOSIGNAL

/* Define to 1 if you have the <dirent.h> header file. */
#undef HAVE_DIRENT_H

/* Have dirfd function */
#undef HAVE_DIRFD

/* Define to 1 if you have the <dlfcn.h> header file. */
#undef HAVE_DLFCN_H

/* Define to 1 if you have the <errno.h> header file. */
#undef HAVE_ERRNO_H

/* Define to 1 if you have the <execinfo.h> header file. */
#undef HAVE_EXECINFO_H

/* Define to 1 if you have the <expat.h> header file. */
#undef HAVE_EXPAT_H
```



```
/* Define to 1 if you have the `fpathconf' function. */
#undef HAVE_FPATHCONF

/* Define to 1 if you have the `getgrouplist' function. */
#undef HAVE_GETGROUPLIST

/* Define to 1 if you have the `getpeereid' function. */
#undef HAVE_GETPEEREID

/* Define to 1 if you have the `getpeerucred' function. */
#undef HAVE_GETPEERUCRED

/* Define to 1 if you have the `getresuid' function. */
#undef HAVE_GETRESUID

/* Have GNU-style varargs macros */
#undef HAVE_GNUC_VARARGS

/* Define to 1 if you have the `inotify_init1' function. */
#undef HAVE_INOTIFY_INIT1

/* Define to 1 if you have the <inttypes.h> header file. */
#undef HAVE_INTTYPES_H

/* Have ISO C99 varargs macros */
#undef HAVE_ISO_VARARGS

/* Define to 1 if you have the `issetugid' function. */
#undef HAVE_ISSETUGID

/* audit daemon SELinux support */
#undef HAVE_LIBAUDIT

/* Define to 1 if you have the `nsl' library (-lnsl). */
#undef HAVE_LIBNSL

/* Define to 1 if you have the `localeconv' function. */
#undef HAVE_LOCALECONV

/* Define to 1 if you have the <locale.h> header file. */
#undef HAVE_LOCALE_H

/* Define to 1 if you have the <memory.h> header file. */
#undef HAVE_MEMORY_H

/* Define if we have CLOCK_MONOTONIC */
#undef HAVE_MONOTONIC_CLOCK

/* Define to 1 if you have the `nanosleep' function. */
#undef HAVE_NANOSLEEP
```

```
/* Have non-POSIX function getpwnam_r */
#undef HAVE_NONPOSIX_GETPWNAM_R

/* Define if your system needs _NSGetEnviron to set up the environment
*/
#undef HAVE_NSGETENVIRON

/* Define to 1 if you have the `pipe2' function. */
#undef HAVE_PIPE2

/* Define to 1 if you have the `poll' function. */
#undef HAVE_POLL

/* Have POSIX function getpwnam_r */
#undef HAVE_POSIX_GETPWNAM_R

/* SELinux support */
#undef HAVE_SELINUX

/* Define to 1 if you have the `setenv' function. */
#undef HAVE_SETENV

/* Define to 1 if you have the `setlocale' function. */
#undef HAVE_SETLOCALE

/* Define to 1 if you have the `setrlimit' function. */
#undef HAVE_SETRLIMIT

/* Define to 1 if you have the <signal.h> header file. */
#undef HAVE_SIGNAL_H

/* Define to 1 if you have the `socketpair' function. */
#undef HAVE_SOCKETPAIR

/* Have socklen_t type */
#undef HAVE_SOCKLEN_T

/* Define to 1 if you have the <stdint.h> header file. */
#undef HAVE_STDINT_H

/* Define to 1 if you have the <stdlib.h> header file. */
#undef HAVE_STDLIB_H

/* Define to 1 if you have the <strings.h> header file. */
#undef HAVE_STRINGS_H

/* Define to 1 if you have the <string.h> header file. */
#undef HAVE_STRING_H

/* Define to 1 if you have the `strtoll' function. */
#undef HAVE_STRTOLL
```

```
/* Define to 1 if you have the `strtoull' function. */
#undef HAVE_STRTOULL

/* Define to 1 if you have the <syslog.h> header file. */
#undef HAVE_SYSLOG_H

/* Have systemd */
#undef HAVE_SYSTEMD

/* Define to 1 if you have the <sys/inotify.h> header file. */
#undef HAVE_SYS_INOTIFY_H

/* Define to 1 if you have the <sys/resource.h> header file. */
#undef HAVE_SYS_RESOURCE_H

/* Define to 1 if you have the <sys/stat.h> header file. */
#undef HAVE_SYS_STAT_H

/* Define to 1 if you have the <sys/syslimits.h> header file. */
#undef HAVE_SYS_SYSLIMITS_H

/* Define to 1 if you have the <sys/types.h> header file. */
#undef HAVE_SYS_TYPES_H

/* Define to 1 if you have the <sys/uio.h> header file. */
#undef HAVE_SYS_UIO_H

/* Define to 1 if you have the <unistd.h> header file. */
#undef HAVE_UNISTD_H

/* Supports sending UNIX file descriptors */
#undef HAVE_UNIX_FD_PASSING

/* Define to 1 if you have the `unsetenv' function. */
#undef HAVE_UNSETENV

/* Define to 1 if you have the `usleep' function. */
#undef HAVE_USLEEP

/* Define to 1 if you have the `vasprintf' function. */
#undef HAVE_VASPRINTF

/* Define to 1 if you have the `vsnprintf' function. */
#undef HAVE_VSNPRINTF

/* Define to 1 if you have the `writev' function. */
#undef HAVE_WRITEV

/* Define to 1 if you have the <ws2tcpip.h> header file. */
#undef HAVE_WS2TCPIP_H

/* Define to 1 if you have the <wspiapi.h> header file. */
```

```
#undef HAVE_WSPIAPI_H

/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#undef LT_OBJDIR

/* Define to 1 if your C compiler doesn't accept -c and -o together.
*/
#undef NO_MINUS_C_MINUS_O

/* Name of package */
#undef PACKAGE

/* Define to the address where bug reports for this package should be
sent. */
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package. */
#undef PACKAGE_NAME

/* Define to the full name and version of this package. */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package. */
#undef PACKAGE_TARNAME

/* Define to the home page for this package. */
#undef PACKAGE_URL

/* Define to the version of this package. */
#undef PACKAGE_VERSION

/* The size of `char', as computed by sizeof. */
#undef SIZEOF_CHAR

/* The size of `int', as computed by sizeof. */
#undef SIZEOF_INT

/* The size of `long', as computed by sizeof. */
#undef SIZEOF_LONG

/* The size of `long long', as computed by sizeof. */
#undef SIZEOF_LONG_LONG

/* The size of `short', as computed by sizeof. */
#undef SIZEOF_SHORT

/* The size of `void *', as computed by sizeof. */
#undef SIZEOF_VOID_P

/* The size of `__int64', as computed by sizeof. */
```

```

#undef SIZEOF___INT64

/* Define to 1 if you have the ANSI C header files. */
#undef STDC_HEADERS

/* Full path to the daemon in the builddir */
#undef TEST_BUS_BINARY

/* Listening address for regression tests */
#undef TEST_LISTEN

/* Enable extensions on AIX 3, Interix. */
#ifndef _ALL_SOURCE
# undef _ALL_SOURCE
#endif
/* Enable GNU extensions on systems that have them. */
#ifndef _GNU_SOURCE
# undef _GNU_SOURCE
#endif
/* Enable threading extensions on Solaris. */
#ifndef _POSIX_PTHREAD_SEMANTICS
# undef _POSIX_PTHREAD_SEMANTICS
#endif
/* Enable extensions on HP NonStop. */
#ifndef _TANDEM_SOURCE
# undef _TANDEM_SOURCE
#endif
/* Enable general extensions on Solaris. */
#ifndef __EXTENSIONS__
# undef __EXTENSIONS__
#endif

/* Version number of package */
#undef VERSION

/* Define to add Valgrind instrumentation */
#undef WITH_VALGRIND

/* Define WORDS_BIGENDIAN to 1 if your processor stores words with the
most
significant byte first (like Motorola and SPARC, unlike Intel). */
#if defined AC_APPLE_UNIVERSAL_BUILD
# if defined __BIG_ENDIAN__
# define WORDS_BIGENDIAN 1
# endif
#else
# ifndef WORDS_BIGENDIAN
# undef WORDS_BIGENDIAN
# endif
#endif

```

```

        /* Use the compiler-provided endianness defines to
allow universal compiling. */
        #if defined(__BIG_ENDIAN__)
        #define WORDS_BIGENDIAN 1
        #endif

/* Define to 1 if the X Window System is missing or not being used. */
#undef X_DISPLAY_MISSING

#if defined(HAVE_NSGETENVIRON) && defined(HAVE_CRT_EXTERNS_H)
# include <sys/time.h>
# include <crt_extrns.h>
# define environ (*_NSGetEnviron())
#endif

/* Define to 1 if on MINIX. */
#undef _MINIX

/* Define to 2 if the system does not provide POSIX.1 features except
with
    this defined. */
#undef _POSIX_1_SOURCE

/* Define to 1 if you need to in order for `stat' and other things to
work. */
#undef _POSIX_SOURCE

/* Defined to get newer W32 CE APIs */
#undef _WIN32_WCE

/* Define to `__inline__' or `__inline' if that's what the C compiler
calls it, or to nothing if 'inline' is not supported under any
name. */
#ifndef __cplusplus
#undef inline
#endif

File = config.h.~1~

/* config.h. Generated from config.h.in by configure. */
/* config.h.in. Generated from configure.ac by autoheader. */

/* Define if building universal (internal helper macro) */
/* #undef AC_APPLE_UNIVERSAL_BUILD */

/* poll doesn't work on devices */

```

```
/* #undef BROKEN_POLL */

/* Directory for installing the binaries */
#define DBUS_BINDIR "/usr/bin"

/* Define to build test code into the library and binaries */
/* #undef DBUS_BUILD_TESTS */

/* Define to build X11 functionality */
/* #undef DBUS_BUILD_X11 */

/* whether -export-dynamic was passed to libtool */
/* #undef DBUS_BUILT_R_DYNAMIC */

/* Use dnotify on Linux */
/* #undef DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX */

/* Use inotify */
#define DBUS_BUS_ENABLE_INOTIFY 1

/* Use kqueue */
/* #undef DBUS_BUS_ENABLE_KQUEUE */

/* Directory to check for console ownership */
#define DBUS_CONSOLE_AUTH_DIR "/var/run/console/"

/* File to check for console ownership */
#define DBUS_CONSOLE_OWNER_FILE ""

/* Defined if we run on a cygwin API based system */
/* #undef DBUS_CYGWIN */

/* Directory for installing the DBUS daemon */
#define DBUS_DAEMONDIR "/usr/bin"

/* Name of executable */
#define DBUS_DAEMON_NAME "dbus-daemon"

/* Directory for installing DBUS data files */
#define DBUS_DATADIR "/usr/share"

/* Disable assertion checking */
#define DBUS_DISABLE_ASSERT 1

/* Disable public API sanity checking */
#define DBUS_DISABLE_CHECKS 1

/* Define to build test code into the library and binaries */
/* #undef DBUS_ENABLE_EMBEDDED_TESTS */

/* Use launchd autolaunch */
/* #undef DBUS_ENABLE_LAUNCHD */
```

```

/* Define to build independent test binaries */
/* #undef DBUS_ENABLE_MODULAR_TESTS */

/* Define to enable bus daemon usage statistics */
/* #undef DBUS_ENABLE_STATS */

/* Build with caching of user data */
#define DBUS_ENABLE_USERDB_CACHE 1

/* Support a verbose mode */
/* #undef DBUS_ENABLE_VERBOSE_MODE */

/* Define to enable X11 auto-launch */
/* #undef DBUS_ENABLE_X11_AUTOLAUNCH */

/* Extension for executables, typically empty or .exe */
#define DBUS_EXEEXT ""

/* Defined if gcov is enabled to force a rebuild due to config.h
changing */
/* #undef DBUS_GCOV_ENABLED */

/* Define to use epoll(4) on Linux */
#define DBUS_HAVE_LINUX_EPOLL 1

/* Define to printf modifier for 64 bit integer type */
#define DBUS_INT64_PRINTF_MODIFIER "ll"

/* Directory for installing the libexec binaries */
#define DBUS_LIBEXECDIR "/usr/lib/dbus"

/* Prefix for installing DBUS */
#define DBUS_PREFIX "/usr"

/* Where per-session bus puts its sockets */
#define DBUS_SESSION_SOCKET_DIR "/tmp"

/* The default D-Bus address of the system bus */
#define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"unix:path=/var/run/dbus/system_bus_socket"

/* The name of the socket the system bus listens on by default */
#define DBUS_SYSTEM_SOCKET "/var/run/dbus/system_bus_socket"

/* Full path to the daemon in the builddir */
#define DBUS_TEST_EXEC
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/test"

/* Full path to the launch helper test program in the builddir */

```



```

#define DBUS_TEST_LAUNCH_HELPER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-launch-helper-test"

/* Where to put test sockets */
#define DBUS_TEST_SOCKET_DIR "/tmp"

/* Defined if we run on a Unix-based system */
#define DBUS_UNIX 1

/* User for running the system BUS daemon */
#define DBUS_USER "messagebus"

/* Use the gcc __sync extension */
#define DBUS_USE_SYNC 1

/* A 'va_copy' style function */
#define DBUS_VA_COPY va_copy

/* 'va_lists' cannot be copied as values */
/* #undef DBUS_VA_COPY_AS_ARRAY */

/* Defined if we run on a W32 API based system */
/* #undef DBUS_WIN */

/* Defined if we run on a W32 CE API based system */
/* #undef DBUS_WINCE */

/* Define if GLib, GObject, GIO are available */
#define DBUS_WITH_GLIB 1

/* The name of the gettext domain */
#define GETTEXT_PACKAGE "dbus-1"

/* Disable GLib public API sanity checking */
#define G_DISABLE_CHECKS 1

/* Have abstract socket namespace */
#define HAVE_ABSTRACT_SOCKETS 1

/* Define to 1 if you have the `accept4' function. */
#define HAVE_ACCEPT4 1

/* Adt audit API */
/* #undef HAVE_ADT */

/* Define to 1 if you have the `backtrace' function. */
#define HAVE_BACKTRACE 1

/* Define to 1 if you have the <byteswap.h> header file. */
#define HAVE_BYTESWAP_H 1

```

```
/* Define to 1 if you have the `clearenv' function. */
#define HAVE_CLEARENV 1

/* Have cmsgcred structure */
/* #undef HAVE_CMSGCRED */

/* Have console owner file */
/* #undef HAVE_CONSOLE_OWNER_FILE */

/* Define to 1 if you have the < crt_extrns.h > header file. */
/* #undef HAVE_CRT_EXTRNS_H */

/* Have the ddfd member of DIR */
/* #undef HAVE_DDFD */

/* Define to 1 if you have the declaration of `LOG_PERROR', and to 0
if you
    don't. */
#define HAVE_DECL_LOG_PERROR 1

/* Define to 1 if you have the declaration of `MSG_NOSIGNAL', and to 0
if you
    don't. */
#define HAVE_DECL_MSG_NOSIGNAL 1

/* Define to 1 if you have the < dirent.h > header file. */
#define HAVE_DIRENT_H 1

/* Have dirfd function */
#define HAVE_DIRFD 1

/* Define to 1 if you have the < dlfcn.h > header file. */
#define HAVE_DLFCN_H 1

/* Define to 1 if you have the < errno.h > header file. */
#define HAVE_ERRNO_H 1

/* Define to 1 if you have the < execinfo.h > header file. */
#define HAVE_EXECINFO_H 1

/* Define to 1 if you have the < expat.h > header file. */
#define HAVE_EXPAT_H 1

/* Define to 1 if you have the `fpathconf' function. */
#define HAVE_FPATHCONF 1

/* Define to 1 if you have the `getgrouplist' function. */
#define HAVE_GETGROUPLIST 1

/* Define to 1 if you have the `getpeereid' function. */
/* #undef HAVE_GETPEEREID */
```

```

/* Define to 1 if you have the `getpeerucred' function. */
/* #undef HAVE_GETPEERUCRED */

/* Define to 1 if you have the `getresuid' function. */
#define HAVE_GETRESUID 1

/* Have GNU-style varargs macros */
#define HAVE_GNUC_VARARGS 1

/* Define to 1 if you have the `inotify_init1' function. */
#define HAVE_INOTIFY_INIT1 1

/* Define to 1 if you have the <inttypes.h> header file. */
#define HAVE_INTTYPES_H 1

/* Have ISO C99 varargs macros */
#define HAVE_ISO_VARARGS 1

/* Define to 1 if you have the `issetugid' function. */
/* #undef HAVE_ISSETUGID */

/* audit daemon SELinux support */
/* #undef HAVE_LIBAUDIT */

/* Define to 1 if you have the `nsl' library (-lnsl). */
/* #undef HAVE_LIBNSL */

/* Define to 1 if you have the `localeconv' function. */
#define HAVE_LOCALECONV 1

/* Define to 1 if you have the <locale.h> header file. */
#define HAVE_LOCALE_H 1

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Define if we have CLOCK_MONOTONIC */
#define HAVE_MONOTONIC_CLOCK 1

/* Define to 1 if you have the `nanosleep' function. */
#define HAVE_NANOSLEEP 1

/* Have non-POSIX function getpwnam_r */
/* #undef HAVE_NONPOSIX_GETPWNAM_R */

/* Define if your system needs _NSGetEnviron to set up the environment
*/
/* #undef HAVE_NSGETENVIRON */

/* Define to 1 if you have the `pipe2' function. */
#define HAVE_PIPE2 1

```

```
/* Define to 1 if you have the `poll' function. */
#define HAVE_POLL 1

/* Have POSIX function getpwnam_r */
#define HAVE_POSIX_GETPWNAM_R 1

/* SELinux support */
/* #undef HAVE_SELINUX */

/* Define to 1 if you have the `setenv' function. */
#define HAVE_SETENV 1

/* Define to 1 if you have the `setlocale' function. */
#define HAVE_SETLOCALE 1

/* Define to 1 if you have the `setrlimit' function. */
#define HAVE_SETRLIMIT 1

/* Define to 1 if you have the <signal.h> header file. */
#define HAVE_SIGNAL_H 1

/* Define to 1 if you have the `socketpair' function. */
#define HAVE_SOCKETPAIR 1

/* Have socklen_t type */
#define HAVE_SOCKLEN_T 1

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE_STDINT_H 1

/* Define to 1 if you have the <stdlib.h> header file. */
#define HAVE_STDLIB_H 1

/* Define to 1 if you have the <strings.h> header file. */
#define HAVE_STRINGS_H 1

/* Define to 1 if you have the <string.h> header file. */
#define HAVE_STRING_H 1

/* Define to 1 if you have the `strtoll' function. */
#define HAVE_STRTOLL 1

/* Define to 1 if you have the `strtoull' function. */
#define HAVE_STRTOULL 1

/* Define to 1 if you have the <syslog.h> header file. */
#define HAVE_SYSLOG_H 1

/* Have systemd */
/* #undef HAVE_SYSTEMD */
```

```
/* Define to 1 if you have the <sys/inotify.h> header file. */
#define HAVE_SYS_INOTIFY_H 1

/* Define to 1 if you have the <sys/resource.h> header file. */
#define HAVE_SYS_RESOURCE_H 1

/* Define to 1 if you have the <sys/stat.h> header file. */
#define HAVE_SYS_STAT_H 1

/* Define to 1 if you have the <sys/syslimits.h> header file. */
/* #undef HAVE_SYS_SYSLIMITS_H */

/* Define to 1 if you have the <sys/types.h> header file. */
#define HAVE_SYS_TYPES_H 1

/* Define to 1 if you have the <sys/uio.h> header file. */
#define HAVE_SYS_UIO_H 1

/* Define to 1 if you have the <unistd.h> header file. */
#define HAVE_UNISTD_H 1

/* Supports sending UNIX file descriptors */
#define HAVE_UNIX_FD_PASSING 1

/* Define to 1 if you have the `unsetenv' function. */
#define HAVE_UNSETENV 1

/* Define to 1 if you have the `usleep' function. */
#define HAVE_USLEEP 1

/* Define to 1 if you have the `vasprintf' function. */
#define HAVE_VASPRINTF 1

/* Define to 1 if you have the `vsnprintf' function. */
#define HAVE_VSNPRINTF 1

/* Define to 1 if you have the `writev' function. */
#define HAVE_WRITEV 1

/* Define to 1 if you have the <ws2tcpip.h> header file. */
/* #undef HAVE_WS2TCPIP_H */

/* Define to 1 if you have the <wspiapi.h> header file. */
/* #undef HAVE_WSPIAPI_H */

/* Define to the sub-directory in which libtool stores uninstalled
libraries.
*/
#define LT_OBJDIR ".libs/"

/* Define to 1 if your C compiler doesn't accept -c and -o together.
*/
```

```
/* #undef NO_MINUS_C_MINUS_O */

/* Name of package */
#define PACKAGE "dbus"

/* Define to the address where bug reports for this package should be
sent. */
#define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"

/* Define to the full name of this package. */
#define PACKAGE_NAME "dbus"

/* Define to the full name and version of this package. */
#define PACKAGE_STRING "dbus 1.6.8"

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME "dbus"

/* Define to the home page for this package. */
#define PACKAGE_URL ""

/* Define to the version of this package. */
#define PACKAGE_VERSION "1.6.8"

/* The size of `char', as computed by sizeof. */
#define SIZEOF_CHAR 1

/* The size of `int', as computed by sizeof. */
#define SIZEOF_INT 4

/* The size of `long', as computed by sizeof. */
#define SIZEOF_LONG 4

/* The size of `long long', as computed by sizeof. */
#define SIZEOF_LONG_LONG 8

/* The size of `short', as computed by sizeof. */
#define SIZEOF_SHORT 2

/* The size of `void *', as computed by sizeof. */
#define SIZEOF_VOID_P 4

/* The size of `__int64', as computed by sizeof. */
#define SIZEOF___INT64 0

/* Define to 1 if you have the ANSI C header files. */
#define STDC_HEADERS 1

/* Full path to the daemon in the builddir */
#define TEST_BUS_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
```

```
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-  
r6.0-aragol/dbus-1.6.8/bus/dbus-daemon"
```

```
/* Listening address for regression tests */
```

```
#define TEST_LISTEN "unix:tmpdir=/tmp"
```

```
/* Enable extensions on AIX 3, Interix. */
```

```
#ifndef _ALL_SOURCE
```

```
# define _ALL_SOURCE 1
```

```
#endif
```

```
/* Enable GNU extensions on systems that have them. */
```

```
#ifndef _GNU_SOURCE
```

```
# define _GNU_SOURCE 1
```

```
#endif
```

```
/* Enable threading extensions on Solaris. */
```

```
#ifndef _POSIX_PTHREAD_SEMANTICS
```

```
# define _POSIX_PTHREAD_SEMANTICS 1
```

```
#endif
```

```
/* Enable extensions on HP NonStop. */
```

```
#ifndef _TANDEM_SOURCE
```

```
# define _TANDEM_SOURCE 1
```

```
#endif
```

```
/* Enable general extensions on Solaris. */
```

```
#ifndef __EXTENSIONS__
```

```
# define __EXTENSIONS__ 1
```

```
#endif
```

```
/* Version number of package */
```

```
#define VERSION "1.6.8"
```

```
/* Define to add Valgrind instrumentation */
```

```
/* #undef WITH_VALGRIND */
```

```
/* Define WORDS_BIGENDIAN to 1 if your processor stores words with the  
most
```

```
significant byte first (like Motorola and SPARC, unlike Intel). */
```

```
#if defined AC_APPLE_UNIVERSAL_BUILD
```

```
# if defined __BIG_ENDIAN__
```

```
# define WORDS_BIGENDIAN 1
```

```
# endif
```

```
#else
```

```
# ifndef WORDS_BIGENDIAN
```

```
/* # undef WORDS_BIGENDIAN */
```

```
# endif
```

```
#endif
```

```
/* Use the compiler-provided endianness defines to  
allow universal compiling. */
```

```
#if defined(__BIG_ENDIAN__)
```

```
#define WORDS_BIGENDIAN 1
```

```

        #endif

/* Define to 1 if the X Window System is missing or not being used. */
#define X_DISPLAY_MISSING 1

#if defined(HAVE_NSGETENVIRON) && defined(HAVE_CRT_EXTERNS_H)
# include <sys/time.h>
# include <crt_extrns.h>
# define environ (*_NSGetEnviron())
#endif

/* Define to 1 if on MINIX. */
/* #undef _MINIX */

/* Define to 2 if the system does not provide POSIX.1 features except
with
    this defined. */
/* #undef _POSIX_1_SOURCE */

/* Define to 1 if you need to in order for `stat' and other things to
work. */
/* #undef _POSIX_SOURCE */

/* Defined to get newer W32 CE APIs */
/* #undef _WIN32_WCE */

/* Define to `__inline__' or `__inline' if that's what the C compiler
calls it, or to nothing if 'inline' is not supported under any
name. */
#ifndef __cplusplus
/* #undef inline */
#endif

```

File = config.log

This file contains any messages produced by compilers while running configure, to aid debugging if configure makes a mistake.

It was created by dbus-glib configure 0.100.2, which was generated by GNU Autoconf 2.69. Invocation command line was

```

$ /home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/configure --build=i686-linux --
host=arm-oe-linux-gnueabi --target=arm-oe-linux-gnueabi --prefix=/usr
--exec_prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --
libexecdir=/usr/lib/dbus-glib --datadir=/usr/share --sysconfdir=/etc -

```



```
--sharedstatedir=/com --localstatedir=/var --libdir=/usr/lib --
includedir=/usr/include --oldincludedir=/usr/include --
infodir=/usr/share/info --mandir=/usr/share/man --disable-silent-rules
--disable-dependency-tracking --with-libtool-
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 --with-introspect-
xml=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/share/dbus/dbus-bus-
introspect.xml --with-dbus-binding-
tool=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/bin/dbus-binding-tool --
enable-nls
```

```
## ----- ##
## Platform. ##
## ----- ##
```

```
hostname = pc45
uname -m = i686
uname -r = 3.11.0-15-generic
uname -s = Linux
uname -v = #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014
```

```
/usr/bin/uname -p = unknown
/bin/uname -X = unknown
```

```
/bin/arch = unknown
/usr/bin/arch -k = unknown
/usr/convex/getsysinfo = unknown
/usr/bin/hostinfo = unknown
/bin/machine = unknown
/usr/bin/oslevel = unknown
/bin/universe = unknown
```

```
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/oe-core/scripts
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/bin/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1/usr/bin/crossscripts
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/sbin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/bin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/sbin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux//bin
PATH: /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/bin
PATH: /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/bin
```

```
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/oe-core/scripts
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/bitbake/bin
PATH: /home/gangadhar/bin
PATH: /usr/lib/lightdm/lightdm
PATH: /usr/local/sbin
PATH: /usr/local/bin
PATH: /usr/sbin
PATH: /usr/bin
PATH: /sbin
PATH: /bin
PATH: /usr/games
PATH: /home/gangadhar/delphi/ti-sdk-am335x-evm-07.00.00.00/linux-
devkit/sysroots/i686-arago-linux/usr/bin
```

```
## ----- ##
## Core tests. ##
## ----- ##
```

```
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/meta-openembedded/meta-
oe/site/endian-little
| # rp-pppoe
| rppoe_cv_pack_bitfields=${rppoe_cv_pack_bitfields=rev}
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/endian-
little
| ac_cv_c_littleendian=${ac_cv_c_littleendian=yes}
| ac_cv_c_bigendian=${ac_cv_c_bigendian=no}
|
| # libnet
| ac_cv_libnet_endianness=${ac_cv_libnet_endianness=lil}
|
| # libmemcached
| ac_cv_c_endian=little
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/arm-
common
| # general
| ac_cv_va_val_copy=${ac_cv_va_val_copy=yes}
|
ac_cv_func_lstat_dereferences_slashed_symlink=${ac_cv_func_lstat_deref
erences_slashed_symlink=yes}
|
ac_cv_func_lstat_empty_string_bug=${ac_cv_func_lstat_empty_string_bug=
no}
|
ac_cv_func_stat_empty_string_bug=${ac_cv_func_stat_empty_string_bug=no}
}
|
ac_cv_func_stat_ignores_trailing_slash=${ac_cv_func_stat_ignores_trail
ing_slash=no}
```

```
| ac_cv_header_netinet_sctp_h=${ac_cv_header_netinet_sctp_h=no}
|
ac_cv_header_netinet_sctp_uio_h=${ac_cv_header_netinet_sctp_uio_h=no}
| ac_cv_sctp=${ac_cv_sctp=no}
|
| # apache
| ac_cv_func_pthread_key_delete=${ac_cv_func_pthread_key_delete=yes}
| apr_cv_process_shared_works=${apr_cv_process_shared_works=no}
| ac_cv_sizeof_ssize_t=${ac_cv_sizeof_ssize_t=4}
| apr_cv_tcp_nodelay_with_cork=${apr_cv_tcp_nodelay_with_cork=yes}
|
| # bash
| ac_cv_c_long_double=${ac_cv_c_long_double=yes}
| bash_cv_func_sigsetjmp=${bash_cv_func_sigsetjmp=missing}
|
| # coreutils
| utils_cv_sys_open_max=${utils_cv_sys_open_max=1019}
|
| # cvs
| cvs_cv_func_printf_ptr=${cvs_cv_func_printf_ptr=yes}
|
| # db (sleepycat)
| db_cv_fcntl_f_setfd=${db_cv_fcntl_f_setfd=yes}
| db_cv_sprintf_count=${db_cv_sprintf_count=yes}
| db_cv_align_t=${db_cv_align_t='unsigned long long'}
| db_cv_alignp_t=${db_cv_alignp_t='unsigned long'}
|
| # D-BUS
| ac_cv_func_posix_getpwnam_r=${ac_cv_func_posix_getpwnam_r=yes}
|
| # edb
| db_cv_spinlocks=${db_cv_spinlocks=no}
|
| # ettercap
| ettercap_cv_type_socklen_t=${ettercap_cv_type_socklen_t=yes}
|
| # fget
| compat_cv_func_snprintf_works=${compat_cv_func_snprintf_works=yes}
| compat_cv_func_basename_works=${compat_cv_func_basename_works=no}
| compat_cv_func_dirname_works=${compat_cv_func_dirname_works=no}
|
| # fnmatch
| ac_cv_func_fnmatch_works=${ac_cv_func_fnmatch_works=yes}
|
| # gettext
| am_cv_func_working_getline=${am_cv_func_working_getline=yes}
|
| # glib
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_sizeof_system_thread=${glib_cv_sizeof_system_thread=4}
| glib_cv_stack_grows=${glib_cv_stack_grows=no}
| glib_cv_uscore=${glib_cv_uscore=no}
```

```

| glib_cv_use_pid_surrogate=${glib_cv_use_pid_surrogate=yes}
| glib_cv_has__inline=${glib_cv_has__inline=yes}
| glib_cv_has__inline__=${glib_cv_has__inline__=yes}
| glib_cv_hasinline=${glib_cv_hasinline=yes}
| glib_cv_sane_realloc=${glib_cv_sane_realloc=yes}
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_uscore=${glib_cv_uscore=no}
| glib_cv_va_val_copy=${glib_cv_va_val_copy=yes}
| glib_cv_rtdglobal_broken=${glib_cv_rtdglobal_broken=no}
|
glib_cv_sys_pthread_mutex_trylock_posix=${glib_cv_sys_pthread_mutex_tr
ylock_posix=yes}
|
glib_cv_sys_pthread_getspecific_posix=${glib_cv_sys_pthread_getspecifi
c_posix=yes}
|
glib_cv_sys_pthread_cond_timedwait_posix=${glib_cv_sys_pthread_cond_ti
medwait_posix=yes}
|
| # glib-2.0
| glib_cv_long_long_format=${glib_cv_long_long_format=ll}
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_sizeof_intmax_t=${glib_cv_sizeof_intmax_t=8}
| glib_cv_sizeof_ptrdiff_t=${glib_cv_sizeof_ptrdiff_t=4}
| glib_cv_sizeof_size_t=${glib_cv_sizeof_size_t=4}
| glib_cv_sizeof_system_thread=${glib_cv_sizeof_system_thread=4}
|
glib_cv_sys_use_pid_niceness_surrogate=${glib_cv_sys_use_pid_niceness_
surrogate=yes}
| ac_cv_alignof_guint32=4
| ac_cv_alignof_guint64=8
| ac_cv_alignof_unsigned_long=4
|
| #gstreamer
| as_cv_unaligned_access=${as_cv_unaligned_access=no}
|
| # httppc
| ac_cv_strerror_r_SUSv3=${ac_cv_strerror_r_SUSv3=no}
|
| # jikes
| ac_cv_sizeof_wchar_t=4
|
| # lftp
| ac_cv_need_trio=${ac_cv_need_trio=no}
| lftp_cv_va_val_copy=${lftp_cv_va_val_copy=yes}
|
| # libesntp
| acx_working_snprintf=${acx_working_snprintf=yes}
|
| # libidl
| libIDL_cv_long_long_format=${libIDL_cv_long_long_format=ll}
|

```

```

| # libnet
| ac_libnet_have_packet_socket=${ac_libnet_have_packet_socket=yes}
|
| # libpcap
| ac_cv_linux_vers=${ac_cv_linux_vers=2}
|
| # libxfce4util
| with_broken_putenv=${with_broken_putenv=no}
|
| # links
|
ac_cv_lib_png_png_create_info_struct=${ac_cv_lib_png_png_create_info_s
truct=yes}
|
| # mysql
| mysql_cv_func_atomic_sub=${mysql_cv_func_atomic_sub=no}
| mysql_cv_func_atomic_add=${mysql_cv_func_atomic_add=no}
|
| # nano
|
ac_cv_regexec_segfault_emptystr=${ac_cv_regexec_segfault_emptystr=no}
|
nano_cv_func_regexec_segv_emptystr=${nano_cv_func_regexec_segv_emptyst
r=no}
|
| # ORBit2
| ac_cv_alignof_CORBA_boolean=1
| ac_cv_alignof_CORBA_char=1
| ac_cv_alignof_CORBA_double=8
| ac_cv_alignof_CORBA_float=4
| ac_cv_alignof_CORBA_long=4
| ac_cv_alignof_CORBA_long_double=8
| ac_cv_alignof_CORBA_long_long=8
| ac_cv_alignof_CORBA_octet=1
| ac_cv_alignof_CORBA_pointer=4
| ac_cv_alignof_CORBA_short=2
| ac_cv_alignof_CORBA_struct=1
| ac_cv_alignof_CORBA_wchar=2
|
| # php
| ac_cv_pread=${ac_cv_pread=no}
| ac_cv_pwrite=${ac_cv_pwrite=no}
|
php_cv_lib_cookie_io_functions_use_off64_t=${php_cv_lib_cookie_io_func
tions_use_off64_t=yes}
|
| # rsync
| rsync_cv_HAVE_BROKEN_LARGEFILE=${rsync_cv_HAVE_BROKEN_LARGEFILE=no}
| rsync_cv_HAVE_SOCKETPAIR=${rsync_cv_HAVE_SOCKETPAIR=yes}
| rsync_cv_HAVE_LONGLONG=${rsync_cv_HAVE_LONGLONG=yes}
| rsync_cv_HAVE_OFF64_T=${rsync_cv_HAVE_OFF64_T=no}
| rsync_cv_HAVE_SHORT_INO_T=${rsync_cv_HAVE_SHORT_INO_T=no}

```

```

| rsync_cv_HAVE_UNSIGNED_CHAR=${rsync_cv_HAVE_UNSIGNED_CHAR=no}
| rsync_cv_HAVE_BROKEN_READDIR=${rsync_cv_HAVE_BROKEN_READDIR=no}
| rsync_cv_HAVE_GETTIMEOFDAY_TZ=${rsync_cv_HAVE_GETTIMEOFDAY_TZ=yes}
| rsync_cv_HAVE_C99_VSNPRINTF=${rsync_cv_HAVE_C99_VSNPRINTF=yes}
| rsync_cv_HAVE_SECURE_MKSTEMP=${rsync_cv_HAVE_SECURE_MKSTEMP=yes}
| rsync_cv_REPLACE_INET_NTOA=${rsync_cv_REPLACE_INET_NTOA=no}
| rsync_cv_REPLACE_INET_ATON=${rsync_cv_REPLACE_INET_ATON=no}
|
| # screen
| screen_cv_sys_bcopy_overlap=${screen_cv_sys_bcopy_overlap=no}
| screen_cv_sys_memcpy_overlap=${screen_cv_sys_memcpy_overlap=no}
| screen_cv_sys_memmove_overlap=${screen_cv_sys_memmove_overlap=no}
| screen_cv_sys_fifo_broken_impl=${screen_cv_sys_fifo_broken_impl=yes}
| screen_cv_sys_fifo_usable=${screen_cv_sys_fifo_usable=yes}
|
screen_cv_sys_select_broken_retval=${screen_cv_sys_select_broken_retval=no}
| screen_cv_sys_sockets_nofs=${screen_cv_sys_sockets_nofs=no}
| screen_cv_sys_sockets_usable=${screen_cv_sys_sockets_usable=yes}
| screen_cv_sys_terminfo_used=${screen_cv_sys_terminfo_used=yes}
|
| # slrn
| slrn_cv_va_val_copy=${slrn_cv_va_val_copy=yes}
|
| # socat
| ac_cv_ispeed_offset=${ac_cv_ispeed_offset=13}
| sc_cv_termios_ispeed=${sc_cv_termios_ispeed=yes}
|
| # ssh
|
ac_cv_have_space_d_name_in_struct_dirent=${ac_cv_dirent_have_space_d_name=yes}
| ac_cv_have_broken_snprintf=${ac_cv_have_broken_snprintf=no}
| ac_cv_have_accrights_in_msghdr=${ac_cv_have_accrights_in_msghdr=no}
| ac_cv_have_control_in_msghdr=${ac_cv_have_control_in_msghdr=yes}
| ac_cv_have_openpty_ctty_bug=${ac_cv_have_openpty_ctty_bug=yes}
|
| # startup-notification
| lf_cv_sane_realloc=yes
|
| # xffm
| jm_cv_func_working_readdir=yes
|
| # evolution-data-server
| ac_cv_libiconv_utf8=${ac_cv_libiconv_utf8=yes}
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/common-
linux
| # apr
| ac_cv_file_dev_zero=${ac_cv_file_dev_zero=yes}
| ac_cv_sizeof_pid_t=${ac_cv_sizeof_pid_t=4}
|

```

```

| # samba
| samba_cv_HAVE_IFACE_AIX=${samba_cv_HAVE_IFACE_AIX=no}
| samba_cv_HAVE_IFACE_IFCONF=${samba_cv_HAVE_IFACE_IFCONF=yes}
| samba_cv_HAVE_IFACE_IFREQ=${samba_cv_HAVE_IFACE_IFREQ=yes}
|
samba_cv_HAVE_IRIX_SPECIFIC_CAPABILITIES=${samba_cv_HAVE_IRIX_SPECIFIC
_CAPABILITIES=no}
|
| # db
| db_cv_path_ar=${db_cv_path_ar=/usr/bin/ar}
| db_cv_path_chmod=${db_cv_path_chmod=/bin/chmod}
| db_cv_path_cp=${db_cv_path_cp=/bin/cp}
| db_cv_path_ln=${db_cv_path_ln=/bin/ln}
| db_cv_path_mkdir=${db_cv_path_mkdir=/bin/mkdir}
| db_cv_path_ranlib=${db_cv_path_ranlib=/usr/bin/ranlib}
| db_cv_path_rm=${db_cv_path_rm=/bin/rm}
| db_cv_path_sh=${db_cv_path_sh=/bin/sh}
| db_cv_path_strip=${db_cv_path_strip=/usr/bin/strip}
|
| # bash
| bash_cv_have_mbstate_t=${bash_cv_have_mbstate_t=yes}
|
bash_cv_must_reinstall_sighandlers=${bash_cv_must_reinstall_sighandler
s=no}
| bash_cv_func_strcoll_broken=${bash_cv_func_strcoll_broken=no}
| bash_cv_dup2_broken=${bash_cv_dup2_broken=no}
| bash_cv_opendir_not_robust=${bash_cv_opendir_not_robust=no}
| bash_cv_type_rlimit=${bash_cv_type_rlimit=rlim_t}
| bash_cv_getenv_redef=${bash_cv_getenv_redef=yes}
| bash_cv_ulimit_maxfds=${bash_cv_ulimit_maxfds=yes}
| bash_cv_getcwd_calls_popen=${bash_cv_getcwd_calls_popen=no}
| bash_cv_printf_a_format=${bash_cv_printf_a_format=yes}
| bash_cv_pgrp_pipe=${bash_cv_pgrp_pipe=no}
| bash_cv_job_control_missing=${bash_cv_job_control_missing=present}
| bash_cv_sys_named_pipes=${bash_cv_sys_named_pipes=present}
| bash_cv_unusable_rtsigs=${bash_cv_unusable_rtsigs=no}
|
| # coreutils
| gl_cv_func_fstatat_zero_flag=${gl_cv_func_fstatat_zero_flag=yes}
|
| # mysql
| ac_cv_sys_restartable_syscalls=${ac_cv_sys_restartable_syscalls=yes}
| ac_cv_conv_longlong_to_float=${ac_cv_conv_longlong_to_float=yes}
|
| # TCL
| tcl_cv_api_serial=${tcl_cv_api_serial=termios}
|
| # python
| ac_cv_have_long_long_format=yes
|
| # apache
| ac_cv_o_nonblock_inherited=${ac_cv_o_nonblock_inherited=no}

```

```

configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/common-
glibc
| # general
| ac_cv_have_decl_sys_siglist=${ac_cv_have_decl_sys_siglist=yes}
| ac_cv_func_realloc_works=${ac_cv_func_realloc_works=yes}
| ac_cv_func_realloc_0_nonnull=${ac_cv_func_realloc_0_nonnull=yes}
| ac_cv_func_malloc_works=${ac_cv_func_malloc_works=yes}
| ac_cv_func_malloc_0_nonnull=${ac_cv_func_malloc_0_nonnull=yes}
| ac_cv_func_memcmp_working=${ac_cv_func_memcmp_working=yes}
| ac_cv_func_getpgrp_void=yes
| ac_cv_func_setpgrp_void=yes
| ac_cv_func_setgrent_void=yes
| ac_cv_func_getgrgid_r=${ac_cv_func_getgrgid_r=yes}
| ac_cv_func_getpwuid_r=${ac_cv_func_getpwuid_r=yes}
| ac_cv_func_posix_getpwuid_r=${ac_cv_func_posix_getpwuid_r=yes}
| ac_cv_func_posix_getgrgid_r=${ac_cv_func_posix_getgrgid_r=yes}
| ac_cv_type_uid_t={ac_cv_type_uid_t=yes}
| ac_cv_func_getaddrinfo=${ac_cv_func_getaddrinfo=yes}
|
| # bash
| bash_cv_under_sys_siglist=${bash_cv_under_sys_siglist=yes}
| bash_cv_sys_siglist=${bash_cv_sys_siglist=yes}
|
| # clamav
|
clamav_av_func_working_snprintf_long=${clamav_av_func_working_snprintf
_long=yes}
| clamav_av_have_in_port_t=${clamav_av_have_in_port_t=yes}
| clamav_av_have_in_addr_t=${clamav_av_have_in_addr_t=yes}
| ac_cv_func_mmap_fixed_mapped=${ac_cv_func_mmap_fixed_mapped=yes}
|
| # coreutils
| fu_cv_sys_stat_statfs2_bsize=${fu_cv_sys_stat_statfs2_bsize=yes}
|
| # glib
| glib_cv_strncpy=${glib_cv_strncpy=no}
| ac_cv_func_printf_unix98=${ac_cv_func_printf_unix98=yes}
| ac_cv_func_snprintf_c99=${ac_cv_func_snprintf_c99=yes}
| ac_cv_func_vsnprintf_c99=${ac_cv_func_vsnprintf_c99=yes}
|
glib_cv_compliant_posix_memalign=${glib_cv_compliant_posix_memalign=1}
| glib_cv_long_long_format=${glib_cv_long_long_format=ll}
| # glib-1.x requires this and pkgconfig-native / pkgconfig-nativesdk
use
| # that to avoid a dependency loop.
| glib_cv___va_copy=${glib_cv___va_copy=yes}
| glib_cv_va_copy=${glib_cv_va_copy=yes}
| glib_cv_have_qsort_r=${glib_cv_have_qsort_r=yes}
|
| #dbus-glib
| ac_cv_have_abstract_sockets=${ac_cv_have_abstract_sockets=yes}

```



```

|
| # gnucash
| am_cv_scanf_lld=${am_cv_scanf_lld=yes}
|
| # guile
|
ac_cv_func_pthread_attr_getstack=${ac_cv_func_pthread_attr_getstack=yes}
|
| #gcc-zlib
| ac_cv_func_getpagesize=${ac_cv_func_getpagesize=yes}
| ac_cv_func_memcpy=${ac_cv_func_memcpy=yes}
| ac_cv_func_strerror=${ac_cv_func_strerror=yes}
|
| # squid
| ac_cv_af_unix_large_dgram=${ac_cv_af_unix_large_dgram=yes}
| ac_cv_func_setresuid=${ac_cv_func_setresuid=yes}
| ac_cv_func_va_copy=${ac_cv_func_va_copy=yes}
| ac_cv_func___va_copy=${ac_cv_func___va_copy=yes}
| ac_cv_epoll_works=${ac_cv_epoll_works=yes}
|
| ac_cv_check_sjllj=ssjllj
|
| # m4
| gt_cv_locale_fr=${gt_cv_locale_fr=fr_FR}
| gl_cv_func_btowc_eof=${gl_cv_func_btowc_eof=yes}
| gl_cv_func_wcrtomb_retval=${gl_cv_func_wcrtomb_retval=yes}
| gl_cv_func_wctob_works=${gl_cv_func_wctob_works=yes}
|
gl_cv_func_mbrtowc_incomplete_state=${gl_cv_func_mbrtowc_incomplete_state=yes}
| gl_cv_func_mbrtowc_sanitycheck=${gl_cv_func_mbrtowc_sanitycheck=yes}
| gl_cv_func_mbrtowc_null_arg=${gl_cv_func_mbrtowc_null_arg=yes}
| gl_cv_func_mbrtowc_retval=${gl_cv_func_mbrtowc_retval=yes}
| gl_cv_func_mbrtowc_nul_retval=${gl_cv_func_mbrtowc_nul_retval=yes}
|
| # va_copy and __va_copy
| ac_cv_va_copy=${ac_cv_va_copy=yes}
| ac_cv___va_copy=${ac_cv___va_copy=yes}
| ac_cv_func_va_copy=${ac_cv_func_va_copy=yes}
| ac_cv_func___va_copy=${ac_cv_func___va_copy=yes}
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/arm-
linux
| ac_cv_func_setvbuf_reversed=no
| ac_cv_sizeof___int64=${ac_cv_sizeof___int64=0}
| ac_cv_sizeof_char=${ac_cv_sizeof_char=1}
| ac_cv_sizeof_wchar_t=${ac_cv_sizeof_wchar_t=1}
| ac_cv_sizeof_unsigned_char=${ac_cv_sizeof_unsigned_char=1}
| ac_cv_sizeof_bool=${ac_cv_sizeof_bool=1}
| ac_cv_sizeof_char_p=${ac_cv_sizeof_int_p=4}
| ac_cv_sizeof_int=${ac_cv_sizeof_int=4}

```

```

| ac_cv_sizeof_int_p=${ac_cv_sizeof_int_p=4}
| ac_cv_sizeof_long=${ac_cv_sizeof_long=4}
| ac_cv_sizeof_long_int=${ac_cv_sizeof_long_int=4}
| ac_cv_sizeof_long_long=${ac_cv_sizeof_long_long=8}
| ac_cv_sizeof_off_t=${ac_cv_sizeof_off_t=4}
| ac_cv_sizeof_short=${ac_cv_sizeof_short=2}
| ac_cv_sizeof_short_int=${ac_cv_sizeof_short_int=2}
| ac_cv_sizeof_size_t=${ac_cv_sizeof_size_t=4}
| ac_cv_sizeof_void_p=${ac_cv_sizeof_void_p=4}
| ac_cv_sizeof_long_p=${ac_cv_sizeof_long_p=4}
| ac_cv_sizeof_float=${ac_cv_sizeof_float=4}
| ac_cv_sizeof_double=${ac_cv_sizeof_double=8}
| ac_cv_sizeof_long_double=${ac_cv_sizeof_long_double=8}
| ac_cv_sizeof_ptrdiff_t=${glib_cv_sizeof_ptrdiff_t=4}
| ac_cv_sizeof_unsigned_short=${ac_cv_sizeof_unsigned_short=2}
| ac_cv_sizeof_unsigned=${ac_cv_sizeof_unsigned=4}
| ac_cv_sizeof_unsigned_int=${ac_cv_sizeof_unsigned_int=4}
| ac_cv_sizeof_unsigned_long=${ac_cv_sizeof_unsigned_long=4}
| ac_cv_sizeof_unsigned_long_long=${ac_cv_sizeof_unsigned_long_long=8}
| ac_cv_sizeof_signed_char=${ac_cv_sizeof_signed_char=1}
|
| ac_cv_uchar=${ac_cv_uchar=no}
| ac_cv_uint=${ac_cv_uint=yes}
| ac_cv_ulong=${ac_cv_ulong=yes}
| ac_cv_ushort=${ac_cv_ushort=yes}
| ac_cv_time_r_type=${ac_cv_time_r_type=POSIX}
|
| # samba
|
samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=${samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=yes}
|
samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=${samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=no}
|
samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=${samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=no}
| samba_cv_HAVE_BROKEN_GETGROUPS=${samba_cv_HAVE_BROKEN_GETGROUPS=no}
|
samba_cv_HAVE_BROKEN_LINUX_SENDFILE=${samba_cv_HAVE_BROKEN_LINUX_SENDFILE=yes}
| samba_cv_HAVE_BROKEN_READDIR=${samba_cv_HAVE_BROKEN_READDIR=yes}
|
samba_cv_HAVE_BROKEN_READDIR_NAME=${samba_cv_HAVE_BROKEN_READDIR_NAME=no}
| samba_cv_HAVE_C99_VSNPRINTF=${samba_cv_HAVE_C99_VSNPRINTF=yes}
| samba_cv_HAVE_DEV64_T=${samba_cv_HAVE_DEV64_T=no}
| samba_cv_HAVE_DEVICE_MAJOR_FN=${samba_cv_HAVE_DEVICE_MAJOR_FN=yes}
| samba_cv_HAVE_DEVICE_MINOR_FN=${samba_cv_HAVE_DEVICE_MINOR_FN=yes}
| samba_cv_HAVE_DQB_FSOFTLIMIT=${samba_cv_HAVE_DQB_FSOFTLIMIT=no}

```

```
|
samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=${samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=yes}
| samba_cv_HAVE_FAM_H=${samba_cv_HAVE_FAM_H=no}
| samba_cv_HAVE_FCNTL_LOCK=${samba_cv_HAVE_FCNTL_LOCK=yes}
| samba_cv_HAVE_FTRUNCATE_EXTEND=${samba_cv_HAVE_FTRUNCATE_EXTEND=yes}
| samba_cv_HAVE_FUNCTION_MACRO=${samba_cv_HAVE_FUNCTION_MACRO=yes}
| samba_cv_HAVE_GETTIMEOFDAY_TZ=${samba_cv_HAVE_GETTIMEOFDAY_TZ=yes}
| samba_cv_HAVE_INO64_T=${samba_cv_HAVE_INO64_T=no}
|
samba_cv_HAVE_INT16_FROM_RPC_RPC_H=${samba_cv_HAVE_INT16_FROM_RPC_RPC_H=no}
|
samba_cv_HAVE_INT32_FROM_RPC_RPC_H=${samba_cv_HAVE_INT32_FROM_RPC_RPC_H=no}
|
samba_cv_HAVE_KERNEL_CHANGE_NOTIFY=${samba_cv_HAVE_KERNEL_CHANGE_NOTIFY=yes}
|
samba_cv_HAVE_KERNEL_OPLOCKS_LINUX=${samba_cv_HAVE_KERNEL_OPLOCKS_LINUX=yes}
|
samba_cv_HAVE_KERNEL_SHARE_MODES=${samba_cv_HAVE_KERNEL_SHARE_MODES=yes}
|
samba_cv_HAVE_MAKEDEV=${samba_cv_HAVE_MAKEDEV=yes}
| samba_cv_HAVE_MMAP=${samba_cv_HAVE_MMAP=yes}
| samba_cv_HAVE_NATIVE_ICONV=${samba_cv_HAVE_NATIVE_ICONV=yes}
| samba_cv_HAVE_OFF64_T=${samba_cv_HAVE_OFF64_T=no}
| samba_cv_HAVE_ROOT=${samba_cv_HAVE_ROOT=yes}
|
samba_cv_HAVE_RPC_AUTH_ERROR_CONFLICT=${samba_cv_HAVE_RPC_AUTH_ERROR_CONFLICT=no}
| samba_cv_HAVE_SECURE_MKSTEMP=${samba_cv_HAVE_SECURE_MKSTEMP=yes}
| samba_cv_HAVE_SENDFILE=${samba_cv_HAVE_SENDFILE=yes}
| samba_cv_HAVE_SENDFILE64=${samba_cv_HAVE_SENDFILE64=yes}
| samba_cv_HAVE_SOCKET_SIN_LEN=${samba_cv_HAVE_SOCKET_SIN_LEN=no}
| samba_cv_HAVE_STAT_ST_BLKSIZE=${samba_cv_HAVE_STAT_ST_BLKSIZE=yes}
| samba_cv_HAVE_STAT_ST_BLOCKS=${samba_cv_HAVE_STAT_ST_BLOCKS=yes}
| samba_cv_HAVE_STRUCT_DIR64=${samba_cv_HAVE_STRUCT_DIR64=no}
| samba_cv_HAVE_STRUCT_DIRENT64=${samba_cv_HAVE_STRUCT_DIRENT64=yes}
| samba_cv_HAVE_STRUCT_FLOCK64=${samba_cv_HAVE_STRUCT_FLOCK64=yes}
| samba_cv_HAVE_TRUNCATED_SALT=${samba_cv_HAVE_TRUNCATED_SALT=no}
|
samba_cv_HAVE_UINT16_FROM_RPC_RPC_H=${samba_cv_HAVE_UINT16_FROM_RPC_RPC_H=no}
|
samba_cv_HAVE_UINT32_FROM_RPC_RPC_H=${samba_cv_HAVE_UINT32_FROM_RPC_RPC_H=no}
| samba_cv_HAVE_UNSIGNED_CHAR=${samba_cv_HAVE_UNSIGNED_CHAR=yes}
| samba_cv_HAVE_UTIMBUF=${samba_cv_HAVE_UTIMBUF=yes}
| samba_cv_HAVE_UT_UT_ADDR=${samba_cv_HAVE_UT_UT_ADDR=yes}
| samba_cv_HAVE_UT_UT_EXIT=${samba_cv_HAVE_UT_UT_EXIT=yes}
```

```

| samba_cv_HAVE_UT_UT_HOST=${samba_cv_HAVE_UT_UT_HOST=yes}
| samba_cv_HAVE_UT_UT_ID=${samba_cv_HAVE_UT_UT_ID=yes}
| samba_cv_HAVE_UT_UT_NAME=${samba_cv_HAVE_UT_UT_NAME=yes}
| samba_cv_HAVE_UT_UT_PID=${samba_cv_HAVE_UT_UT_PID=yes}
| samba_cv_HAVE_UT_UT_TIME=${samba_cv_HAVE_UT_UT_TIME=yes}
| samba_cv_HAVE_UT_UT_TV=${samba_cv_HAVE_UT_UT_TV=yes}
| samba_cv_HAVE_UT_UT_TYPE=${samba_cv_HAVE_UT_UT_TYPE=yes}
| samba_cv_HAVE_UT_UT_USER=${samba_cv_HAVE_UT_UT_USER=yes}
| samba_cv_HAVE_UX_UT_SYSLLEN=${samba_cv_HAVE_UX_UT_SYSLLEN=no}
| samba_cv_HAVE_VA_COPY=${samba_cv_HAVE_VA_COPY=yes}
| samba_cv_HAVE_WORKING_AF_LOCAL=${samba_cv_HAVE_WORKING_AF_LOCAL=yes}
| samba_cv_HAVE_Werror=${samba_cv_HAVE_Werror=yes}
|
samba_cv_PUTUTLINE_RETURNS_UTMP=${samba_cv_PUTUTLINE_RETURNS_UTMP=yes}
| samba_cv_QUOTA_WORKS=${samba_cv_QUOTA_WORKS=yes}
| samba_cv_REALPATH_TAKES_NULL=${samba_cv_REALPATH_TAKES_NULL=yes}
| samba_cv_REPLACE_GETPASS=${samba_cv_REPLACE_GETPASS=yes}
| samba_cv_REPLACE_INET_NTOA=${samba_cv_REPLACE_INET_NTOA=no}
| samba_cv_REPLACE_READDIR=${samba_cv_REPLACE_READDIR=no}
| samba_cv_RUN_QUOTA_TESTS=${samba_cv_RUN_QUOTA_TESTS=yes}
| samba_cv_SEEKDIR_RETURNS_VOID=${samba_cv_SEEKDIR_RETURNS_VOID=yes}
| samba_cv_SIZEOF_DEV_T=${samba_cv_SIZEOF_DEV_T=yes}
| samba_cv_SIZEOF_INO_T=${samba_cv_SIZEOF_INO_T=yes}
| samba_cv_SIZEOF_OFF_T=${samba_cv_SIZEOF_OFF_T=yes}
|
samba_cv_SYSCONF_SC_NGROUPS_MAX=${samba_cv_SYSCONF_SC_NGROUPS_MAX=yes}
|
samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=${samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=yes}
|
| samba_cv_SYSCONF_SC_NPROC_ONLN=${samba_cv_SYSCONF_SC_NPROC_ONLN=no}
| samba_cv_SYSCONF_SC_PAGESIZE=${samba_cv_SYSCONF_SC_PAGESIZE=yes}
| samba_cv_SYSQUOTA_FOUND=${samba_cv_SYSQUOTA_FOUND=yes}
| samba_cv_SYSQUOTA_WORKS=${samba_cv_SYSQUOTA_WORKS=yes}
| samba_cv_SYSQUOTA_WORKS_XFS=${samba_cv_SYSQUOTA_WORKS_XFS=yes}
| samba_cv_TRY_QUOTAS=${samba_cv_TRY_QUOTAS=no}
| samba_cv_TRY_SYS_QUOTAS=${samba_cv_TRY_SYS_QUOTAS=yes}
| samba_cv_USE_SETRESUID=${samba_cv_USE_SETRESUID=yes}
| samba_cv_WE_USE_SYS_QUOTAS=${samba_cv_WE_USE_SYS_QUOTAS=yes}
| samba_cv_WITH_AFS=${samba_cv_WITH_AFS=no}
| samba_cv_WITH_FAKE_KASERVER=${samba_cv_WITH_FAKE_KASERVER=no}
| samba_cv_WITH_QUOTAS=${samba_cv_WITH_QUOTAS=auto}
| samba_cv_WITH_SYS_QUOTAS=${samba_cv_WITH_SYS_QUOTAS=auto}
| samba_cv_WITH_VFS_AFSACL=${samba_cv_WITH_VFS_AFSACL=no}
| samba_cv_compiler_supports_ll=${samba_cv_compiler_supports_ll=yes}
| samba_cv_found_xfs_header=${samba_cv_found_xfs_header=yes}
| samba_cv_have_longlong=${samba_cv_have_longlong=yes}
| samba_cv_have_setresgid=${samba_cv_have_setresgid=yes}
| samba_cv_have_setresuid=${samba_cv_have_setresuid=yes}
| samba_cv_immediate_structures=${samba_cv_immediate_structures=yes}
|
samba_cv_optimize_out_funcation_calls=${samba_cv_optimize_out_funcation_calls=yes}

```

```

| samba_cv_sig_atomic_t=${samba_cv_sig_atomic_t=yes}
| samba_cv_socklen_t=${samba_cv_socklen_t=yes}
| samba_cv_struct_timespec=${samba_cv_struct_timespec=yes}
|
samba_cv_sysquotas_file=${samba_cv_sysquotas_file=lib/sysquotas_linux.
c}
| samba_cv_unixsocket=${samba_cv_unixsocket=yes}
| samba_cv_volatile=${samba_cv_volatile=yes}
|
| #older sambe defines
| samba_cv_USE_SETEUID=${samba_cv_USE_SETEUID=yes}
| samba_cv_USE_SETREUID=${samba_cv_USE_SETREUID=yes}
| samba_cv_USE_SETUIDX=${samba_cv_USE_SETUIDX=yes}
| samba_cv_LINUX_LFS_SUPPORT=${samba_cv_LINUX_LFS_SUPPORT=yes}
|
| # clamav
|
clamav_av_func_working_snprintf_long=${clamav_av_func_working_snprintf
_long=yes}
| clamav_av_have_in_port_t=${clamav_av_have_in_port_t=yes}
| clamav_av_have_in_addr_t=${clamav_av_have_in_addr_t=yes}
| ac_cv_func_mmap_fixed_mapped=${ac_cv_func_mmap_fixed_mapped=yes}
|
| #dbus
| ac_cv_have_abstract_sockets=${ac_cv_have_abstract_sockets=yes}
|
| # lftp
| ac_cv_file__dev_ptc_=yes
|
| # guile
|
ac_cv_func_pthread_attr_getstack=${ac_cv_func_pthread_attr_getstack=ye
s}
|
| # gnet
|
ac_cv_member_struct_sockaddr_sa_len=${ac_cv_member_struct_sockaddr_sa_
len=no}
|
ac_cv_gnet_have_abstract_sockets=${ac_cv_gnet_have_abstract_sockets=no
}
|
gnet_sockaddr_family_field_name=${gnet_sockaddr_family_field_name=ss_f
amily}
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdsk/sources/oe-core/meta/site/common
| # libelf
| mr_cv_target_elf=${mr_cv_target_elf=yes}
|
| # Normally kernels have /dev/random enabled
| ac_cv_file__dev_random=${ac_cv_file__dev_random=yes}
|

```

```
| # Avoid sudo to assume void for unsetenv in cross environment, or
else it conflicts with
| # target stdlib.h prototype which follows POSIX compliance. Mark
for upstream.
| sudo_cv_func_unsetenv_void=no
|
| # shadow dir info, to avoid searching build system
| shadow_cv_maildir=${localstatedir}/spool/mail
| shadow_cv_mailfile=Mailbox
| shadow_cv_utmpdir=${localstatedir}/run
| shadow_cv_logdir=${localstatedir}/log
| shadow_cv_passwd_dir=${bindir}
|
| # python: deactivate a runtime check for ipv6-support in python
>=2.7.1 that fails when cross-compiling
| ac_cv_buggy_getaddrinfo=no
configure:2228: loading site script
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vpl/usr/share/arm-oe-linux-
gnueabi_config_site.d/ncurses_config
| ac_cv_header_X=${ac_cv_header_X=no}
| ac_cv_header__CC=${ac_cv_header__CC=no}
| ac_cv_header_CFLAGS=${ac_cv_header_CFLAGS=no}
| ac_cv_header_CPPFLAGS=${ac_cv_header_CPPFLAGS=no}
| ac_cv_header__5=${ac_cv_header__5=no}
| ac_cv_header__c=${ac_cv_header__c=no}
| ac_cv_header_conftest__ac_ext=${ac_cv_header_conftest__ac_ext=no}
| ac_cv_header_curses_h=${ac_cv_header_curses_h=yes}
| ac_cv_header_inttypes_h=${ac_cv_header_inttypes_h=yes}
| ac_cv_header_memory_h=${ac_cv_header_memory_h=yes}
| ac_cv_header_ncurses_curses_h=${ac_cv_header_ncurses_curses_h=no}
| ac_cv_header_ncurses_h=${ac_cv_header_ncurses_h=yes}
| ac_cv_header_ncurses_termcap_h=${ac_cv_header_ncurses_termcap_h=no}
| ac_cv_header_stdcl=${ac_cv_header_stdcl=yes}
| ac_cv_header_stdint_h=${ac_cv_header_stdint_h=yes}
| ac_cv_header_stdlib_h=${ac_cv_header_stdlib_h=yes}
| ac_cv_header_string_h=${ac_cv_header_string_h=yes}
| ac_cv_header_strings_h=${ac_cv_header_strings_h=yes}
| ac_cv_header_sys_stat_h=${ac_cv_header_sys_stat_h=yes}
| ac_cv_header_sys_types_h=${ac_cv_header_sys_types_h=yes}
| ac_cv_header_unistd_h=${ac_cv_header_unistd_h=yes}
configure:2358: checking build system type
configure:2372: result: i686-pc-linux-gnu
configure:2392: checking host system type
configure:2405: result: arm-oe-linux-gnueabi
configure:2442: checking for a BSD-compatible install
configure:2510: result: /usr/bin/install -c
configure:2521: checking whether build environment is sane
configure:2576: result: yes
configure:2635: checking for arm-oe-linux-gnueabi-strip
configure:2662: result: arm-linux-gnueabi-hf-strip
configure:2727: checking for a thread-safe mkdir -p
```

```
configure:2766: result: /bin/mkdir -p
configure:2773: checking for gawk
configure:2789: found /usr/bin/gawk
configure:2800: result: gawk
configure:2811: checking whether make sets $(MAKE)
configure:2833: result: yes
configure:2929: checking whether to enable maintainer-specific
portions of Makefiles
configure:2938: result: no
configure:2963: checking whether make supports nested variables
configure:2980: result: yes
configure:3022: checking for arm-oe-linux-gnueabi-gcc
configure:3049: result: arm-linux-gnueabi-gcc -march=armv7-a -marm
-mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:3318: checking for C compiler version
configure:3327: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 --version >&5
arm-linux-gnueabi-gcc (crosstool-NG linaro-1.13.1-4.7-2013.03-
20130313 - Linaro GCC 2013.03) 4.7.3 20130226 (prerelease)
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.

configure:3338: $? = 0
configure:3327: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -v >&5
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
COLLECT_LTO_WRAPPER=/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/bin/./libexec/gcc/arm-linux-
gnueabi/4.7.3/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-linux/.build/src/gcc-linaro-4.7-2013.03/configure --
build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --
target=arm-linux-gnueabi --prefix=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-linux/install --with-
sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-
linux/install/arm-linux-gnueabi/libc --enable-
languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-
tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-
pkgversion='crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro
GCC 2013.03' --with-bugurl=https://bugs.launchpad.net/gcc-linaro --
enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-
```

```
libssp --with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-hf-linux/.build/arm-linux-gnueabi-hf/build/static --with-
mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
ppl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
cloog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-host-libstdcxx='-
L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static/lib -lpwl' --enable-
threads=posix --disable-libstdcxx-pch --enable-linker-build-id --
enable-gold --with-local-prefix=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-linux/install/arm-linux-gnueabi-hf/libc -
-enable-c99 --enable-long-long --with-mode=thumb
Thread model: posix
gcc version 4.7.3 20130226 (prerelease) (crosstool-NG linaro-1.13.1-
4.7-2013.03-20130313 - Linaro GCC 2013.03)
configure:3338: $? = 0
configure:3327: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -V >&5
arm-linux-gnueabi-hf-gcc: error: unrecognized command line option '-V'
arm-linux-gnueabi-hf-gcc: fatal error: no input files
compilation terminated.
configure:3338: $? = 1
configure:3327: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -qversion >&5
arm-linux-gnueabi-hf-gcc: error: unrecognized command line option '-
qversion'
arm-linux-gnueabi-hf-gcc: fatal error: no input files
compilation terminated.
configure:3338: $? = 1
configure:3358: checking whether the C compiler works
configure:3380: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
```



```
20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-style=gnu
confptest.c >&5
configure:3384: $? = 0
configure:3432: result: yes
configure:3435: checking for C compiler default output file name
configure:3437: result: a.out
configure:3443: checking for suffix of executables
configure:3450: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c >&5
configure:3454: $? = 0
configure:3476: result:
configure:3498: checking whether we are cross compiling
configure:3536: result: yes
configure:3541: checking for suffix of object files
configure:3563: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector confptest.c >&5
configure:3567: $? = 0
configure:3588: result: o
configure:3592: checking whether we are using the GNU C compiler
configure:3611: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector confptest.c >&5
configure:3611: $? = 0
configure:3620: result: yes
configure:3629: checking whether arm-linux-gnueabihf-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
```

```
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 accepts -g
configure:3649: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -g -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c >&5
configure:3649: $? = 0
configure:3690: result: yes
configure:3707: checking for arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to accept ISO C89
configure:3770: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confptest.c >&5
configure:3770: $? = 0
configure:3783: result: none needed
configure:3814: checking for style of include used by make
configure:3842: result: GNU
configure:3868: checking dependency style of arm-linux-gnueabi-hf-gcc
-march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:3979: result: none
configure:3995: checking for library containing strerror
configure:4026: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c >&5
configure:4026: $? = 0
configure:4043: result: none required
configure:4056: checking how to run the C preprocessor
```

```
configure:4126: result: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8
configure:4146: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
configure:4146: $? = 0
configure:4160: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
confptest.c:11:28: fatal error: ac_nonexistent.h: No such file or
directory
compilation terminated.
configure:4160: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus-glib"
| #define PACKAGE_TARNAME "dbus-glib"
| #define PACKAGE_VERSION "0.100.2"
| #define PACKAGE_STRING "dbus-glib 0.100.2"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLi
b"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus-glib"
| #define VERSION "0.100.2"
| /* end confdefs.h. */
| #include <ac_nonexistent.h>
configure:4189: checking for grep that handles long lines and -e
configure:4247: result: /bin/grep
configure:4252: checking for egrep
configure:4314: result: /bin/grep -E
configure:4319: checking for ANSI C header files
configure:4423: result: yes
configure:4570: checking whether gcc understands -Wfloat-equal
configure:4580: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wfloat-equal -isystem/home/gangadhar/gcc-linaro-arm-
```

```
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include
-fstack-protector confctest.c >&5
configure:4580: $? = 0
configure:4593: result: yes
configure:4738: checking how to print strings
configure:4765: result: printf
configure:4786: checking for a sed that does not truncate output
configure:4850: result: /bin/sed
configure:4868: checking for fgrep
configure:4930: result: /bin/grep -F
configure:4965: checking for ld used by arm-linux-gnueabi-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:5032: result: arm-linux-gnueabi-ld --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:5039: checking if the linker (arm-linux-gnueabi-ld --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1) is GNU ld
configure:5054: result: yes
configure:5066: checking for BSD- or MS-compatible name lister (nm)
configure:5115: result: arm-linux-gnueabi-nm
configure:5245: checking the name lister (arm-linux-gnueabi-nm)
interface
configure:5252: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector confctest.c >&5
configure:5255: arm-linux-gnueabi-nm "confctest.o"
configure:5258: output
00000000 B some_variable
configure:5265: result: BSD nm
configure:5268: checking whether ln -s works
configure:5272: result: yes
configure:5280: checking the maximum length of command line arguments
configure:5410: result: 1572864
configure:5427: checking whether the shell understands some XSI
constructs
configure:5437: result: yes
configure:5441: checking whether the shell understands "+="
configure:5447: result: yes
configure:5482: checking how to convert i686-pc-linux-gnu file names
to arm-oe-linux-gnueabi format
```

```
configure:5522: result: func_convert_file_noop
configure:5529: checking how to convert i686-pc-linux-gnu file names
to toolchain format
configure:5549: result: func_convert_file_noop
configure:5556: checking for arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to reload object files
configure:5563: result: -r
configure:5597: checking for arm-oe-linux-gnueabi-objdump
configure:5624: result: arm-linux-gnueabi-hf-objdump
configure:5696: checking how to recognize dependent libraries
configure:5898: result: pass_all
configure:5943: checking for arm-oe-linux-gnueabi-dlltool
configure:5973: result: no
configure:5983: checking for dlltool
configure:6013: result: no
configure:6043: checking how to associate runtime and link libraries
configure:6070: result: printf %s\n
configure:6086: checking for arm-oe-linux-gnueabi-ar
configure:6113: result: arm-linux-gnueabi-hf-ar
configure:6194: checking for archiver @FILE support
configure:6211: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector confctest.c >&5
configure:6211: $? = 0
configure:6214: arm-linux-gnueabi-hf-ar cru libconfctest.a @confctest.lst
>&5
configure:6217: $? = 0
configure:6222: arm-linux-gnueabi-hf-ar cru libconfctest.a @confctest.lst
>&5
arm-linux-gnueabi-hf-ar: confctest.o: No such file or directory
configure:6225: $? = 1
configure:6237: result: @
configure:6255: checking for arm-oe-linux-gnueabi-strip
configure:6282: result: arm-linux-gnueabi-hf-strip
configure:6354: checking for arm-oe-linux-gnueabi-ranlib
configure:6381: result: arm-linux-gnueabi-hf-ranlib
configure:6523: checking command to parse arm-linux-gnueabi-hf-nm
output from arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 object
configure:6643: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```

```
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector confptest.c >&5
figure:6646: $? = 0
figure:6650: arm-linux-gnueabihf-nm confptest.o \
| sed -n -e 's/^[^
]*([ABCDGIRSTW][ABCDGIRSTW]*)[ ]*[ ]*\([^_A-Za-z][_A-Za-z0-9]*\)$/\1 \2
\2/p' | sed '/ __gnu_lto/d' \> confptest.nm
figure:6653: $? = 0
figure:6719: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -Wall -Wchar-subscripts -
Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-
arith -Wcast-align -Wfloat-equal -Wsign-compare -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -
L/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-style=gnu
confptest.c conftstm.o >&5
figure:6722: $? = 0
figure:6760: result: ok
figure:6797: checking for sysroot
figure:6827: result:
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1
figure:7043: checking for arm-oe-linux-gnueabi-mt
figure:7073: result: no
figure:7083: checking for mt
figure:7099: found /bin/mt
figure:7110: result: mt
figure:7122: WARNING: using cross tools not prefixed with host
triplet
figure:7133: checking if mt is a manifest tool
figure:7139: mt '-?'
figure:7147: result: no
figure:7789: checking for sys/types.h
figure:7789: result: yes
figure:7789: checking for sys/stat.h
figure:7789: result: yes
figure:7789: checking for stdlib.h
figure:7789: result: yes
```

```
configure:7789: checking for string.h
configure:7789: result: yes
configure:7789: checking for memory.h
configure:7789: result: yes
configure:7789: checking for strings.h
configure:7789: result: yes
configure:7789: checking for inttypes.h
configure:7789: result: yes
configure:7789: checking for stdint.h
configure:7789: result: yes
configure:7789: checking for unistd.h
configure:7789: result: yes
configure:7803: checking for dlfcn.h
configure:7803: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector conftest.c >&5
configure:7803: $? = 0
configure:7803: result: yes
configure:8010: checking for objdir
configure:8025: result: .libs
configure:8292: checking if arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 supports -fno-rtti -fno-
exceptions
configure:8310: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -fno-rtti -fno-exceptions
conftest.c >&5
ccl: warning: command line option '-fno-rtti' is valid for C++/ObjC++
but not for C [enabled by default]
configure:8314: $? = 0
configure:8327: result: no
configure:8654: checking for arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```

```
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to produce PIC
configure:8661: result: -fPIC -DPIC
configure:8669: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 PIC flag -fPIC -DPIC works
configure:8687: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -fPIC -DPIC -DPIC conftest.c >&5
configure:8691: $? = 0
configure:8704: result: yes
configure:8733: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 static flag -static works
configure:8761: result: yes
configure:8776: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 supports -c -o file.o
configure:8797: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -o out/conftest2.o conftest.c >&5
configure:8801: $? = 0
configure:8823: result: yes
configure:8831: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 supports -c -o file.o
configure:8878: result: yes
configure:8911: checking whether the arm-linux-gnueabi-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 linker (arm-linux-gnueabi-ld --
```



```
configure:11782: result: yes
configure:11834: checking for library containing socket
configure:11865: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -Wall -Wchar-subscripts -
Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-
arith -Wcast-align -Wfloat-equal -Wsign-compare -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/include -fstack-protector -
L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/lib -Wl,-O1 -Wl,--hash-style=gnu
conftest.c >&5
configure:11865: $? = 0
configure:11882: result: none required
configure:11891: checking whether socklen_t is defined
configure:11911: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-gcc/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-gcc/include -fstack-protector conftest.c >&5
conftest.c: In function 'main':
conftest.c:35:11: warning: variable 'foo' set but not used [-Wunused-
but-set-variable]
configure:11911: $? = 0
configure:11917: result: yes
configure:11934: checking abstract socket namespace
configure:12000: result: yes
configure:12035: checking for XML_ParserCreate_MM in -lexpat
configure:12060: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -Wall -Wchar-subscripts -
Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-
arith -Wcast-align -Wfloat-equal -Wsign-compare -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/include -fstack-protector -
L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
```

```
20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-  
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-  
20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu  
confptest.c -lexpat >&5  
configure:12060: $? = 0  
configure:12069: result: yes  
configure:12074: checking expat.h usability  
configure:12074: arm-linux-gnueabi-gcc -march=armv7-a -marm -  
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-  
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-  
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-  
debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-  
prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal  
-Wsign-compare -fno-strict-aliasing -isystem/home/gangadhar/gcc-  
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-  
gnueabi/include -fstack-protector confptest.c >&5  
configure:12074: $? = 0  
configure:12074: result: yes  
configure:12074: checking expat.h presence  
configure:12074: arm-linux-gnueabi-gcc -E --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-  
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -  
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-  
20130313_linux/arm-linux-gnueabi/include -fstack-protector  
confptest.c  
configure:12074: $? = 0  
configure:12074: result: yes  
configure:12074: checking for expat.h  
configure:12074: result: yes  
configure:12111: checking for arm-oe-linux-gnueabi-pkg-config  
configure:12144: result: no  
configure:12154: checking for pkg-config  
configure:12172: found  
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/i686-linux/usr/bin/pkg-config  
configure:12184: result:  
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/i686-linux/usr/bin/pkg-config  
configure:12209: checking pkg-config is at least version 0.9.0  
configure:12212: result: yes  
configure:12222: checking for DBUS  
configure:12229: $PKG_CONFIG --exists --print-errors "dbus-1 >=  
1.2.16"  
configure:12232: $? = 0  
configure:12245: $PKG_CONFIG --exists --print-errors "dbus-1 >=  
1.2.16"  
configure:12248: $? = 0  
configure:12305: result: yes  
configure:12315: checking for DBUS_GLIB
```

```
configure:12322: $PKG_CONFIG --exists --print-errors "gobject-2.0 >=
2.26, gio-2.0 >= 2.26"
configure:12325: $? = 0
configure:12338: $PKG_CONFIG --exists --print-errors "gobject-2.0 >=
2.26, gio-2.0 >= 2.26"
configure:12341: $? = 0
configure:12398: result: yes
configure:12404: checking for DBUS_GLIB_THREADS
configure:12411: $PKG_CONFIG --exists --print-errors "gthread-2.0 >=
2.6"
configure:12414: $? = 0
configure:12427: $PKG_CONFIG --exists --print-errors "gthread-2.0 >=
2.6"
configure:12430: $? = 0
configure:12467: result: yes
configure:12499: checking for gtkdoc-check
configure:12532: result: no
configure:12541: checking for gtkdoc-rebase
configure:12574: result: no
configure:12585: checking for gtkdoc-mkpdf
configure:12618: result: no
configure:12747: checking whether to build gtk-doc documentation
configure:12749: result: no
configure:13066: checking that generated files are newer than
configure
configure:13072: result: done
configure:13131: creating ./config.status
```

```
## ----- ##
## Running config.status. ##
## ----- ##
```

This file was extended by dbus-glib config.status 0.100.2, which was generated by GNU Autoconf 2.69. Invocation command line was

```
CONFIG_FILES =
CONFIG_HEADERS =
CONFIG_LINKS =
CONFIG_COMMANDS =
$ ./config.status
```

on pc45

```
config.status:1202: creating Makefile
config.status:1202: creating m4/Makefile
config.status:1202: creating doc/Makefile
config.status:1202: creating doc/reference/Makefile
config.status:1202: creating doc/reference/version.xml
config.status:1202: creating dbus/Makefile
config.status:1202: creating dbus/examples/Makefile
config.status:1202: creating dbus/examples/statemachine/Makefile
config.status:1202: creating test/Makefile
```

```
config.status:1202: creating test/core/Makefile
config.status:1202: creating test/interfaces/Makefile
config.status:1202: creating test/data/valid-service-files/debug-
glib.service
config.status:1202: creating test/data/valid-service-files/debug-
echo.service
config.status:1202: creating test/data/valid-service-files/interfaces-
test.service
config.status:1202: creating test/lib/Makefile
config.status:1202: creating test/manual/Makefile
config.status:1202: creating tools/Makefile
config.status:1202: creating dbus-glib-1.pc
config.status:1202: creating dbus-glib-1-uninstalled.pc
config.status:1202: creating config.h
config.status:1431: executing depfiles commands
config.status:1431: executing libtool commands
configure:15369: WARNING: unrecognized options: --enable-nls
```

```
## ----- ##
## Cache variables. ##
## ----- ##
```

```
ac_cv___va_copy=yes
ac_cv_af_unix_large_dgram=yes
ac_cv_alignof_CORBA_boolean=1
ac_cv_alignof_CORBA_char=1
ac_cv_alignof_CORBA_double=8
ac_cv_alignof_CORBA_float=4
ac_cv_alignof_CORBA_long=4
ac_cv_alignof_CORBA_long_double=8
ac_cv_alignof_CORBA_long_long=8
ac_cv_alignof_CORBA_octet=1
ac_cv_alignof_CORBA_pointer=4
ac_cv_alignof_CORBA_short=2
ac_cv_alignof_CORBA_struct=1
ac_cv_alignof_CORBA_wchar=2
ac_cv_alignof_guint32=4
ac_cv_alignof_guint64=8
ac_cv_alignof_unsigned_long=4
ac_cv_buggy_getaddrinfo=no
ac_cv_build=i686-pc-linux-gnu
ac_cv_c_bigendian=no
ac_cv_c_compiler_gnu=yes
ac_cv_c_endian=little
ac_cv_c_littleendian=yes
ac_cv_c_long_double=yes
ac_cv_check_sjlj=ssjlj
ac_cv_conv_longlong_to_float=yes
ac_cv_dirent_have_space_d_name=yes
ac_cv_env_CC_set=set
ac_cv_env_CC_value='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```

```
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
ac_cv_env_CFLAGS_set=set
ac_cv_env_CFLAGS_value='-isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -
fstack-protector -O2 -pipe -g -feliminate-unused-debug-types'
ac_cv_env_CPPFLAGS_set=set
ac_cv_env_CPPFLAGS_value='-isystem/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include
-fstack-protector'
ac_cv_env_CPP_set=set
ac_cv_env_CPP_value='arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
ac_cv_env_DBUS_CFLAGS_set=
ac_cv_env_DBUS_CFLAGS_value=
ac_cv_env_DBUS_GLIB_CFLAGS_set=
ac_cv_env_DBUS_GLIB_CFLAGS_value=
ac_cv_env_DBUS_GLIB_LIBS_set=
ac_cv_env_DBUS_GLIB_LIBS_value=
ac_cv_env_DBUS_GLIB_THREADS_CFLAGS_set=
ac_cv_env_DBUS_GLIB_THREADS_CFLAGS_value=
ac_cv_env_DBUS_GLIB_THREADS_LIBS_set=
ac_cv_env_DBUS_GLIB_THREADS_LIBS_value=
ac_cv_env_DBUS_LIBS_set=
ac_cv_env_DBUS_LIBS_value=
ac_cv_env_GTKDOC_DEPS_CFLAGS_set=
ac_cv_env_GTKDOC_DEPS_CFLAGS_value=
ac_cv_env_GTKDOC_DEPS_LIBS_set=
ac_cv_env_GTKDOC_DEPS_LIBS_value=
ac_cv_env_LDFLAGS_set=set
ac_cv_env_LDFLAGS_value='-L/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-
rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu'
ac_cv_env_LIBS_set=
ac_cv_env_LIBS_value=
ac_cv_env_PKG_CONFIG_LIBDIR_set=set
ac_cv_env_PKG_CONFIG_LIBDIR_value=/home/gangadhar/newyoctobuild/tisdk/
build/arago-tmp-external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig
ac_cv_env_PKG_CONFIG_PATH_set=set
ac_cv_env_PKG_CONFIG_PATH_value=/home/gangadhar/newyoctobuild/tisdk/bu
ild/arago-tmp-external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig:/home/gangadhar/newyoctobuild/tisdk/build/arago
-tmp-external-linaro-toolchain/sysroots/delphi-vp1/usr/share/pkgconfig
ac_cv_env_PKG_CONFIG_set=
ac_cv_env_PKG_CONFIG_value=
ac_cv_env_build_alias_set=set
ac_cv_env_build_alias_value=i686-linux
ac_cv_env_host_alias_set=set
```

ac_cv_env_host_alias_value=arm-oe-linux-gnueabi
ac_cv_env_target_alias_set=set
ac_cv_env_target_alias_value=arm-oe-linux-gnueabi
ac_cv_epoll_works=yes
ac_cv_file__dev_ptc_=yes
ac_cv_file__dev_random=yes
ac_cv_file__dev_zero=yes
ac_cv_func__va_copy=yes
ac_cv_func_fnmatch_works=yes
ac_cv_func_getaddrinfo=yes
ac_cv_func_getgrgid_r=yes
ac_cv_func_getpagesize=yes
ac_cv_func_getpgrp_void=yes
ac_cv_func_getpwuid_r=yes
ac_cv_func_lstat_dereferences_slashed_symlink=yes
ac_cv_func_lstat_empty_string_bug=no
ac_cv_func_malloc_0_nonnull=yes
ac_cv_func_malloc_works=yes
ac_cv_func_memcmp_working=yes
ac_cv_func_memcpy=yes
ac_cv_func_mmap_fixed_mapped=yes
ac_cv_func_posix_getgrgid_r=yes
ac_cv_func_posix_getpwnam_r=yes
ac_cv_func_posix_getpwuid_r=yes
ac_cv_func_printf_unix98=yes
ac_cv_func_pthread_attr_getstack=yes
ac_cv_func_pthread_key_delete=yes
ac_cv_func_realloc_0_nonnull=yes
ac_cv_func_realloc_works=yes
ac_cv_func_setgrent_void=yes
ac_cv_func_setpgrp_void=yes
ac_cv_func_setresuid=yes
ac_cv_func_setvbuf_reversed=no
ac_cv_func_sprintf_c99=yes
ac_cv_func_stat_empty_string_bug=no
ac_cv_func_stat_ignores_trailing_slash=no
ac_cv_func_strerror=yes
ac_cv_func_va_copy=yes
ac_cv_func_vsnprintf_c99=yes
ac_cv_gnet_have_abstract_sockets=no
ac_cv_have_abstract_sockets=yes
ac_cv_have_accrights_in_msghdr=no
ac_cv_have_broken_sprintf=no
ac_cv_have_control_in_msghdr=yes
ac_cv_have_decl_sys_siglist=yes
ac_cv_have_long_long_format=yes
ac_cv_have_openpty_ctty_bug=yes
ac_cv_have_space_d_name_in_struct_dirent=yes
ac_cv_header_X=no
ac_cv_header__CC=no
ac_cv_header__CFLAGS=no
ac_cv_header__CPPFLAGS=no

```
ac_cv_header___5=no
ac_cv_header__c=no
ac_cv_header_confstest__ac_ext=no
ac_cv_header_curses_h=yes
ac_cv_header_dlfcn_h=yes
ac_cv_header_expats_h=yes
ac_cv_header_inttypes_h=yes
ac_cv_header_memory_h=yes
ac_cv_header_ncurses_curses_h=no
ac_cv_header_ncurses_h=yes
ac_cv_header_ncurses_termcap_h=no
ac_cv_header_netinet_sctp_h=no
ac_cv_header_netinet_sctp_uio_h=no
ac_cv_header_stdcl=yes
ac_cv_header_stdint_h=yes
ac_cv_header_stdlib_h=yes
ac_cv_header_string_h=yes
ac_cv_header_strings_h=yes
ac_cv_header_sys_stat_h=yes
ac_cv_header_sys_types_h=yes
ac_cv_header_unistd_h=yes
ac_cv_host=arm-oe-linux-gnueabi
ac_cv_ispeed_offset=13
ac_cv_lib_expat_XML_ParserCreate_MM=yes
ac_cv_lib_png_png_create_info_struct=yes
ac_cv_libiconv_utf8=yes
ac_cv_libnet_endianness=lil
ac_cv_linux_vers=2
ac_cv_member_struct_sockaddr_sa_len=no
ac_cv_need_trio=no
ac_cv_o_nonblock_inherited=no
ac_cv_objext=o
ac_cv_path_EGREP='/bin/grep -E'
ac_cv_path_FGREP='/bin/grep -F'
ac_cv_path_GREP=/bin/grep
ac_cv_path_SED=/bin/sed
ac_cv_path_ac_pt_PKG_CONFIG=/home/gangadhar/newyoctobuild/tisdk/build/
arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-
config
ac_cv_path_install='/usr/bin/install -c'
ac_cv_path_mkdir=/bin/mkdir
ac_cv_pread=no
ac_cv_prog_AR=arm-linux-gnueabihf-ar
ac_cv_prog_AWK=gawk
ac_cv_prog_CC='arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl'
ac_cv_prog_CPP='arm-linux-gnueabihf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
```



```
ac_cv_prog_OBJDUMP=arm-linux-gnueabi-hf-objdump
ac_cv_prog_RANLIB=arm-linux-gnueabi-hf-ranlib
ac_cv_prog_STRIP=arm-linux-gnueabi-hf-strip
ac_cv_prog_ac_ct_MANIFEST_TOOL=mt
ac_cv_prog_cc_c89=
ac_cv_prog_cc_g=yes
ac_cv_prog_make_make_set=yes
ac_cv_pwrite=no
ac_cv_regex_exec_segfault_emptystr=no
ac_cv_sctp=no
ac_cv_search_socket='none required'
ac_cv_search_strerror='none required'
ac_cv_sizeof__int64=0
ac_cv_sizeof_bool=1
ac_cv_sizeof_char=1
ac_cv_sizeof_char_p=4
ac_cv_sizeof_double=8
ac_cv_sizeof_float=4
ac_cv_sizeof_int=4
ac_cv_sizeof_int_p=4
ac_cv_sizeof_long=4
ac_cv_sizeof_long_double=8
ac_cv_sizeof_long_int=4
ac_cv_sizeof_long_long=8
ac_cv_sizeof_long_p=4
ac_cv_sizeof_off_t=4
ac_cv_sizeof_pid_t=4
ac_cv_sizeof_ptrdiff_t=4
ac_cv_sizeof_short=2
ac_cv_sizeof_short_int=2
ac_cv_sizeof_signed_char=1
ac_cv_sizeof_size_t=4
ac_cv_sizeof_ssize_t=4
ac_cv_sizeof_unsigned=4
ac_cv_sizeof_unsigned_char=1
ac_cv_sizeof_unsigned_int=4
ac_cv_sizeof_unsigned_long=4
ac_cv_sizeof_unsigned_long_long=8
ac_cv_sizeof_unsigned_short=2
ac_cv_sizeof_void_p=4
ac_cv_sizeof_wchar_t=4
ac_cv_strerror_r_SUSv3=no
ac_cv_sys_restartable_syscalls=yes
ac_cv_time_r_type=POSIX
ac_cv_type_uid_t='{ac_cv_type_uid_t=yes}'
ac_cv_uchar=no
ac_cv_uint=yes
ac_cv_ulong=yes
ac_cv_ushort=yes
ac_cv_va_copy=yes
ac_cv_va_val_copy=yes
ac_cv_CC_dependencies_compiler_type=none
```

am_cv_func_working_getline=yes
am_cv_make_support_nested_variables=yes
am_cv_scanf_lld=yes
apr_cv_process_shared_works=no
apr_cv_tcp_nodelay_with_cork=yes
as_cv_unaligned_access=no
bash_cv_dup2_broken=no
bash_cv_func_sigsetjmp=missing
bash_cv_func_strcoll_broken=no
bash_cv_getcwd_calls_popen=no
bash_cv_getenv_redef=yes
bash_cv_have_mbstate_t=yes
bash_cv_job_control_missing=present
bash_cv_must_reinstall_sighandlers=no
bash_cv_opendir_not_robust=no
bash_cv_pgrp_pipe=no
bash_cv_printf_a_format=yes
bash_cv_sys_named_pipes=present
bash_cv_sys_siglist=yes
bash_cv_type_rlimit=rlim_t
bash_cv_ulimit_maxfds=yes
bash_cv_under_sys_siglist=yes
bash_cv_unusable_rtsigs=no
compat_cv_func_basename_works=no
compat_cv_func_dirname_works=no
compat_cv_func_snprintf_works=yes
cvs_cv_func_printf_ptr=yes
db_cv_align_t='unsigned long long'
db_cv_alignp_t='unsigned long'
db_cv_fcntl_f_setfd=yes
db_cv_path_ar=/usr/bin/ar
db_cv_path_chmod=/bin/chmod
db_cv_path_cp=/bin/cp
db_cv_path_ln=/bin/ln
db_cv_path_mkdir=/bin/mkdir
db_cv_path_ranlib=/usr/bin/ranlib
db_cv_path_rm=/bin/rm
db_cv_path_sh=/bin/sh
db_cv_path_strip=/usr/bin/strip
db_cv_spinlocks=no
db_cv_sprintf_count=yes
ettercap_cv_type_socklen_t=yes
fu_cv_sys_stat_statfs2_bsize=yes
gl_cv_func_btowc_eof=yes
gl_cv_func_fstatat_zero_flag=yes
gl_cv_func_mbrtowc_incomplete_state=yes
gl_cv_func_mbrtowc_nul_retval=yes
gl_cv_func_mbrtowc_null_arg=yes
gl_cv_func_mbrtowc_retval=yes
gl_cv_func_mbrtowc_sanitycheck=yes
gl_cv_func_wcrtomb_retval=yes
gl_cv_func_wctob_works=yes

```
glib_cv___va_copy=yes
glib_cv_compliant_posix_memalign=1
glib_cv_has__inline=yes
glib_cv_has__inline__=yes
glib_cv_hasinline=yes
glib_cv_have_qsort_r=yes
glib_cv_long_long_format=ll
glib_cv_rtdglobal_broken=no
glib_cv_sane_realloc=yes
glib_cv_sizeof_gmutex=24
glib_cv_sizeof_intmax_t=8
glib_cv_sizeof_ptrdiff_t=4
glib_cv_sizeof_size_t=4
glib_cv_sizeof_system_thread=4
glib_cv_stack_grows=no
glib_cv_strncpy=no
glib_cv_sys_pthread_cond_timedwait_posix=yes
glib_cv_sys_pthread_getspecific_posix=yes
glib_cv_sys_pthread_mutex_trylock_posix=yes
glib_cv_sys_use_pid_niceness_surrogate=yes
glib_cv_uscore=no
glib_cv_use_pid_surrogate=yes
glib_cv_va_copy=yes
glib_cv_va_val_copy=yes
gt_cv_locale_fr=fr_FR
jm_cv_func_working_readdir=yes
lf_cv_sane_realloc=yes
lftp_cv_va_val_copy=yes
libIDL_cv_long_long_format=ll
lt_cv_ar_at_file=@
lt_cv_archive_cmds_need_lc=no
lt_cv_deplibs_check_method=pass_all
lt_cv_file_magic_cmd='$MAGIC_CMD'
lt_cv_file_magic_test_file=
lt_cv_ld_reload_flag=-r
lt_cv_nm_interface='BSD nm'
lt_cv_objdir=.libs
lt_cv_path_LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl'
lt_cv_path_NM=arm-linux-gnueabi-hf-nm
lt_cv_path_manifest_tool=no
lt_cv_prog_compiler_c_o=yes
lt_cv_prog_compiler_pic='-fPIC -DPIC'
lt_cv_prog_compiler_pic_works=yes
lt_cv_prog_compiler_rtti_exceptions=no
lt_cv_prog_compiler_static_works=yes
lt_cv_prog_gnu_ld=yes
lt_cv_sharedlib_from_linklib_cmd='printf %s\n'
lt_cv_shlibpath_overrides_runpath=no
```

```

lt_cv_sys_global_symbol_pipe='sed -n -e '\''s/^[
]\([ABCDGIRSTW][ABCDGIRSTW]*\) [ ]*\([_A-Za-z][_A-Za-z0-9]*\)$/\1 \2 \2/p'\'' | sed '\''/_gnu_lto/d'\''
lt_cv_sys_global_symbol_to_c_name_address='sed -n -e '\''s/^[
]*\) [ ]*$ / {\1\1", (void *) 0},/p'\'' -e '\''s/^[ABCDGIRSTW]* \([
]*\) \([
]*\)$/ {"\2", (void *) \&\2},/p'\''
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix='sed -n -e
'\''s/^[
]*\) [ ]*$ / {\1\1", (void *) 0},/p'\'' -e
'\''s/^[ABCDGIRSTW]* \([
]*\) \([
]*\)$/ {"\2", (void *)
&\2},/p'\'' -e '\''s/^[ABCDGIRSTW]* \([
]*\) \([
]*\)$/ {"lib\2",
(void *) \&\2},/p'\''
lt_cv_sys_global_symbol_to_cdecl='sed -n -e '\''s/^T .* \(.*)$/extern
int \1();/p'\'' -e '\''s/^[ABCDGIRSTW]* .* \(.*)$/extern char
\1;/p'\''
lt_cv_sys_lib_dlsearch_path_spec='/usr/lib /lib'
lt_cv_sys_max_cmd_len=1572864
lt_cv_to_host_file_cmd=func_convert_file_noop
lt_cv_to_tool_file_cmd=func_convert_file_noop
mr_cv_target_elf=yes
mysql_cv_func_atomic_add=no
mysql_cv_func_atomic_sub=no
nano_cv_func_regex_exec_segfault=no
php_cv_lib_cookie_io_functions_use_off64_t=yes
pkg_cv_DBUS_CFLAGS='-I/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1/usr/include/dbus-1.0
-I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/dbus-1.0/include '
pkg_cv_DBUS_GLIB_CFLAGS='-pthread -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include '
pkg_cv_DBUS_GLIB_LIBS='-lgio-2.0 -lgobject-2.0 -lglib-2.0 '
pkg_cv_DBUS_GLIB_THREADS_CFLAGS='-pthread -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include '
pkg_cv_DBUS_GLIB_THREADS_LIBS='-pthread -lgthread-2.0 -lrt -lglib-2.0
'
pkg_cv_DBUS_LIBS='-ldbus-1 '
rppoe_cv_pack_bitfields=rev
rsync_cv_HAVE_BROKEN_LARGEFILE=no
rsync_cv_HAVE_BROKEN_READDIR=no
rsync_cv_HAVE_C99_VSNPRINTF=yes
rsync_cv_HAVE_GETTIMEOFDAY_TZ=yes
rsync_cv_HAVE_LONGLONG=yes
rsync_cv_HAVE_OFF64_T=no
rsync_cv_HAVE_SECURE_MKSTEMP=yes
rsync_cv_HAVE_SHORT_INO_T=no
rsync_cv_HAVE_SOCKETPAIR=yes
rsync_cv_HAVE_UNSIGNED_CHAR=no

```

rsync_cv_REPLACE_INET_ATON=no
rsync_cv_REPLACE_INET_NTOA=no
samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=yes
samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=no
samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=no
samba_cv_HAVE_BROKEN_GETGROUPS=no
samba_cv_HAVE_BROKEN_LINUX_SENDFILE=yes
samba_cv_HAVE_BROKEN_READDIR=yes
samba_cv_HAVE_BROKEN_READDIR_NAME=no
samba_cv_HAVE_C99_VSNPRINTF=yes
samba_cv_HAVE_DEV64_T=no
samba_cv_HAVE_DEVICE_MAJOR_FN=yes
samba_cv_HAVE_DEVICE_MINOR_FN=yes
samba_cv_HAVE_DQB_FSOFTLIMIT=no
samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=yes
samba_cv_HAVE_FAM_H=no
samba_cv_HAVE_FCNTL_LOCK=yes
samba_cv_HAVE_FTRUNCATE_EXTEND=yes
samba_cv_HAVE_FUNCTION_MACRO=yes
samba_cv_HAVE_GETTIMEOFDAY_TZ=yes
samba_cv_HAVE_IFACE_AIX=no
samba_cv_HAVE_IFACE_IFCONF=yes
samba_cv_HAVE_IFACE_IFREQ=yes
samba_cv_HAVE_INO64_T=no
samba_cv_HAVE_INT16_FROM_RPC_RPC_H=no
samba_cv_HAVE_INT32_FROM_RPC_RPC_H=no
samba_cv_HAVE_IRIX_SPECIFIC_CAPABILITIES=no
samba_cv_HAVE_KERNEL_CHANGE_NOTIFY=yes
samba_cv_HAVE_KERNEL_OPLOCKS_LINUX=yes
samba_cv_HAVE_KERNEL_SHARE_MODES=yes
samba_cv_HAVE_MAKEDEV=yes
samba_cv_HAVE_MMAP=yes
samba_cv_HAVE_NATIVE_ICONV=yes
samba_cv_HAVE_OFF64_T=no
samba_cv_HAVE_ROOT=yes
samba_cv_HAVE_RPC_AUTH_ERROR_CONFLICT=no
samba_cv_HAVE_SECURE_MKSTEMP=yes
samba_cv_HAVE_SENDFILE64=yes
samba_cv_HAVE_SENDFILE=yes
samba_cv_HAVE_SOCKET_SIN_LEN=no
samba_cv_HAVE_STAT_ST_BLKSIZE=yes
samba_cv_HAVE_STAT_ST_BLOCKS=yes
samba_cv_HAVE_STRUCT_DIR64=no
samba_cv_HAVE_STRUCT_DIRENT64=yes
samba_cv_HAVE_STRUCT_FLOCK64=yes
samba_cv_HAVE_TRUNCATED_SALT=no
samba_cv_HAVE_UINT16_FROM_RPC_RPC_H=no
samba_cv_HAVE_UINT32_FROM_RPC_RPC_H=no
samba_cv_HAVE_UNSIGNED_CHAR=yes
samba_cv_HAVE_UTIMBUF=yes
samba_cv_HAVE_UT_UT_ADDR=yes
samba_cv_HAVE_UT_UT_EXIT=yes

samba_cv_HAVE_UT_UT_HOST=yes
samba_cv_HAVE_UT_UT_ID=yes
samba_cv_HAVE_UT_UT_NAME=yes
samba_cv_HAVE_UT_UT_PID=yes
samba_cv_HAVE_UT_UT_TIME=yes
samba_cv_HAVE_UT_UT_TV=yes
samba_cv_HAVE_UT_UT_TYPE=yes
samba_cv_HAVE_UT_UT_USER=yes
samba_cv_HAVE_UX_UT_SYSLIN=no
samba_cv_HAVE_VA_COPY=yes
samba_cv_HAVE_WORKING_AF_LOCAL=yes
samba_cv_HAVE_Werror=yes
samba_cv_LINUX_LFS_SUPPORT=yes
samba_cv_PUTUTLINE_RETURNS_UTMP=yes
samba_cv_QUOTA_WORKS=yes
samba_cv_REALPATH_TAKES_NULL=yes
samba_cv_REPLACE_GETPASS=yes
samba_cv_REPLACE_INET_NTOA=no
samba_cv_REPLACE_READDIR=no
samba_cv_RUN_QUOTA_TESTS=yes
samba_cv_SEEKDIR_RETURNS_VOID=yes
samba_cv_SIZEOF_DEV_T=yes
samba_cv_SIZEOF_INO_T=yes
samba_cv_SIZEOF_OFF_T=yes
samba_cv_SYSCONF_SC_NGROUPS_MAX=yes
samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=yes
samba_cv_SYSCONF_SC_NPROC_ONLN=no
samba_cv_SYSCONF_SC_PAGESIZE=yes
samba_cv_SYSQUOTA_FOUND=yes
samba_cv_SYSQUOTA_WORKS=yes
samba_cv_SYSQUOTA_WORKS_XFS=yes
samba_cv_TRY_QUOTAS=no
samba_cv_TRY_SYS_QUOTAS=yes
samba_cv_USE_SETEUID=yes
samba_cv_USE_SETRESUID=yes
samba_cv_USE_SETREUID=yes
samba_cv_USE_SETUIDX=yes
samba_cv_WE_USE_SYS_QUOTAS=yes
samba_cv_WITH_AFS=no
samba_cv_WITH_FAKE_KASERVER=no
samba_cv_WITH_QUOTAS=auto
samba_cv_WITH_SYS_QUOTAS=auto
samba_cv_WITH_VFS_AFSACL=no
samba_cv_compiler_supports_ll=yes
samba_cv_found_xfs_header=yes
samba_cv_have_longlong=yes
samba_cv_have_setresgid=yes
samba_cv_have_setresuid=yes
samba_cv_immediate_structures=yes
samba_cv_optimize_out_funcation_calls=yes
samba_cv_sig_atomic_t=yes
samba_cv_socklen_t=yes

```
samba_cv_struct_timespec=yes
samba_cv_sysquotas_file=lib/sysquotas_linux.c
samba_cv_unixsocket=yes
samba_cv_volatile=yes
sc_cv_termios_ispeed=yes
screen_cv_sys_bcopy_overlap=no
screen_cv_sys_fifo_broken_impl=yes
screen_cv_sys_fifo_usable=yes
screen_cv_sys_memcpy_overlap=no
screen_cv_sys_memmove_overlap=no
screen_cv_sys_select_broken_retval=no
screen_cv_sys_sockets_nofs=no
screen_cv_sys_sockets_usable=yes
screen_cv_sys_terminfo_used=yes
shadow_cv_logdir=/var/log
shadow_cv_maildir=/var/spool/mail
shadow_cv_mailfile=Mailbox
shadow_cv_passwd_dir=/usr/bin
shadow_cv_utmpdir=/var/run
slrn_cv_va_val_copy=yes
sudo_cv_func_unsetenv_void=no
tcl_cv_api_serial=termios
utils_cv_sys_open_max=1019
```

```
## ----- ##
## Output variables. ##
## ----- ##
```

```
ABSOLUTE_TOP_BUILDDIR='/home/gangadhar/newyoctobuild/tisdk/build/arago-
-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2'
ACLOCAL='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/missing --run aclocal-
1.12 '
AMDEPBACKSLASH=' '
AMDEP_FALSE=' '
AMDEP_TRUE='#'
AMTAR='${TAR-tar}'
AM_BACKSLASH='\ '
AM_DEFAULT_V='$ (AM_DEFAULT_VERBOSITY) '
AM_DEFAULT_VERBOSITY='1'
AM_V='$ (V) '
AR='arm-linux-gnueabi-hf-ar'
AUTOCONF='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/missing --run autoconf'
AUTOHEADER='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/missing --run
autoheader'
```

```
AUTOMAKE='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-  
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-  
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/missing --run automake-  
1.12'  
AWK='gawk'  
CC='arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -  
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1'  
CCDEPMODE='depmode=none'  
CFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-  
2013.03-20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_/  
include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -  
Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-prototypes -  
Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-equal -Wsign-  
compare -fno-strict-aliasing'  
CPP='arm-linux-gnueabi-gcc -E --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-  
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'  
CPPFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-  
2013.03-20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_/  
include -fstack-protector'  
CYGPATH_W='echo'  
DBUS_BASH_COMPLETION_FALSE='#'  
DBUS_BASH_COMPLETION_TRUE=''  
DBUS_BINDING_TOOL='/home/gangadhar/newyoctobuild/tisdk/build/arago-  
tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/dbus-  
binding-tool'  
DBUS_BUILD_TESTS_FALSE=''  
DBUS_BUILD_TESTS_TRUE='#'  
DBUS_CFLAGS='-I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-  
external-linaro-toolchain/sysroots/delphi-vp1/usr/include/dbus-1.0 -  
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/delphi-vp1/usr/lib/dbus-1.0/include '  
DBUS_GLIB_CFLAGS='-pthread -  
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -  
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include '  
DBUS_GLIB_LIBS='-lgio-2.0 -lgobject-2.0 -lglib-2.0 '  
DBUS_GLIB_THREADS_CFLAGS='-pthread -  
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -  
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-  
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include '  
DBUS_GLIB_THREADS_LIBS='-pthread -lgthread-2.0 -lrt -lglib-2.0 '  
DBUS_GLIB_TOOL_CFLAGS=''  
DBUS_GLIB_TOOL_LIBS='-lexpat'  
DBUS_LIBS='-ldbuss-1 '  
DBUS_PATH_OR_ABSTRACT='abstract'  
DEFS='-DHAVE_CONFIG_H'  
DEPDIR='.deps'  
DLLTOOL='false'
```



```
DSYMUTIL=''
DUMPBIN=''
ECHO_C=''
ECHO_N='-n'
ECHO_T=''
EGREP='/bin/grep -E'
ENABLE_GTK_DOC_FALSE=''
ENABLE_GTK_DOC_TRUE='#'
EXEEXT=''
EXPANDED_BINDIR='/usr/bin'
EXPANDED_DATADIR='/usr/share'
EXPANDED_LIBDIR='/usr/lib'
EXPANDED_LOCALSTATEDIR='/var'
EXPANDED_SYSCONFDIR='/etc'
FGREP='/bin/grep -F'
GLIB_GENMARSHAL='glib-genmarshal'
GREP='/bin/grep'
GTKDOC_CHECK=''
GTKDOC_DEPS_CFLAGS=''
GTKDOC_DEPS_LIBS=''
GTKDOC_MKPDF=''
GTKDOC_REBASE='true'
GTK_DOC_BUILD_HTML_FALSE='#'
GTK_DOC_BUILD_HTML_TRUE=''
GTK_DOC_BUILD_PDF_FALSE=''
GTK_DOC_BUILD_PDF_TRUE='#'
GTK_DOC_USE_LIBTOOL_FALSE='#'
GTK_DOC_USE_LIBTOOL_TRUE=''
GTK_DOC_USE_REBASE_FALSE='#'
GTK_DOC_USE_REBASE_TRUE=''
HAVE_GLIB_THREADS_FALSE='#'
HAVE_GLIB_THREADS_TRUE=''
HTML_DIR='${datadir}/gtk-doc/html'
INSTALL_DATA='${INSTALL} -m 644'
INSTALL_PROGRAM='${INSTALL}'
INSTALL_SCRIPT='${INSTALL}'
INSTALL_STRIP_PROGRAM='${install_sh} -c -s'
LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl'
LDFLAGS='-L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-style=gnu'
LIBOBJS=''
LIBS=''
LIBTOOL='${top_builddir}/arm-oe-linux-gnueabi-libtool'
LIPO=''
LN_S='ln -s'
LTLIBOBJS=''
LT_AGE='2'
LT_CURRENT='4'
```

```
LT_REVISION='2'
MAINT='#'
MAINTAINER_MODE_FALSE=''
MAINTAINER_MODE_TRUE='#'
MAKEINFO='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/missing --run makeinfo'
MANIFEST_TOOL=':'
MKDIR_P='/bin/mkdir -p'
NM='arm-linux-gnueabi-hf-nm'
NMEDIT=''
OBJDUMP='arm-linux-gnueabi-hf-objdump'
OBJEXT='o'
OTOOL64=''
OTOOL=''
PACKAGE='dbus-glib'
PACKAGE_BUGREPORT='https://bugs.freedesktop.org/enter_bug.cgi?product=
dbus&component=Glib'
PACKAGE_NAME='dbus-glib'
PACKAGE_STRING='dbus-glib 0.100.2'
PACKAGE_TARNAME='dbus-glib'
PACKAGE_URL=''
PACKAGE_VERSION='0.100.2'
PATH_SEPARATOR=':'
PKG_CONFIG='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config'
PKG_CONFIG_LIBDIR='/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1//usr/lib/pkgconfig'
PKG_CONFIG_PATH='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig:/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-
vp1/usr/share/pkgconfig'
RANLIB='arm-linux-gnueabi-hf-ranlib'
SED='/bin/sed'
SET_MAKE=''
SHELL='/bin/sh'
STRIP='arm-linux-gnueabi-hf-strip'
TEST_CORE_SERVICE_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/ar
ago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-
linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/core/test-
service-glib'
TEST_EXIT_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-exit'
TEST_INTERFACES_SERVICE_BINARY='/home/gangadhar/newyoctobuild/tisdk/bu
ild/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-
0.100.2/test/interfaces/test-service'
TEST_SEGFAULT_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-segfault'
```

```
TEST_SERVICE_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/arago-  
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-  
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-service'  
TEST_SERVICE_DIR='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-  
external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-  
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/data/valid-  
service-files'  
TEST_SHELL_SERVICE_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/a  
rago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-  
linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-shell-  
service'  
TEST_SLEEP_FOREVER_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/a  
rago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-  
linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-sleep-  
forever'  
TEST_SOCKET_DIR=''  
VERSION='0.100.2'  
ac_ct_AR=''  
ac_ct_CC=''  
ac_ct_DUMPBIN=''  
am__EXEEXT_FALSE=''  
am__EXEEXT_TRUE='#'  
am__fastdepCC_FALSE=''  
am__fastdepCC_TRUE='#'  
am__include='include'  
am__isrc=''  
am__leading_dot='.'  
am__nodep=''  
am__quote=''  
am__tar='$$${TAR-tar} chof - "$${stardir}"'  
am__untar='$$${TAR-tar} xf -'  
bindir='/usr/bin'  
build='i686-pc-linux-gnu'  
build_alias='i686-linux'  
build_cpu='i686'  
build_os='linux-gnu'  
build_vendor='pc'  
datadir='/usr/share'  
datarootdir='${prefix}/share'  
docdir='${datarootdir}/doc/${PACKAGE_TARNAME}'  
dvidir='${docdir}'  
exec_prefix='/usr'  
host='arm-oe-linux-gnueabi'  
host_alias='arm-oe-linux-gnueabi'  
host_cpu='arm'  
host_os='linux-gnueabi'  
host_vendor='oe'  
htmldir='${docdir}'  
includedir='/usr/include'  
infodir='/usr/share/info'
```

```

install_sh='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/install-sh'
libdir='/usr/lib'
libexecdir='/usr/lib/dbus-glib'
localedir='${datarootdir}/locale'
localstatedir='/var'
mandir='/usr/share/man'
mkdir_p='${MKDIR_P}'
oldincludedir='/usr/include'
pdfdir='${docdir}'
prefix='/usr'
program_transform_name='s,x,x,'
psdir='${docdir}'
sbindir='/usr/sbin'
sharedstatedir='/com'
sysconfdir='/etc'
target_alias='arm-oe-linux-gnueabi'

## ----- ##
## confdefs.h. ##
## ----- ##

/* confdefs.h */
#define PACKAGE_NAME "dbus-glib"
#define PACKAGE_TARNAME "dbus-glib"
#define PACKAGE_VERSION "0.100.2"
#define PACKAGE_STRING "dbus-glib 0.100.2"
#define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLi
b"
#define PACKAGE_URL ""
#define PACKAGE "dbus-glib"
#define VERSION "0.100.2"
#define STDC_HEADERS 1
#define DBUS_BASH_COMPLETION 1
#define DBUS_DISABLE_ASSERT 1
#define G_DISABLE_ASSERT 1
#define HAVE_SYS_TYPES_H 1
#define HAVE_SYS_STAT_H 1
#define HAVE_STDLIB_H 1
#define HAVE_STRING_H 1
#define HAVE_MEMORY_H 1
#define HAVE_STRINGS_H 1
#define HAVE_INTTYPES_H 1
#define HAVE_STDINT_H 1
#define HAVE_UNISTD_H 1
#define HAVE_DLFCN_H 1
#define LT_OBJDIR ".libs/"
#define HAVE_SOCKLEN_T 1
#define HAVE_ABSTRACT_SOCKETS 1
#define HAVE_EXPAT_H 1

```

```
#define TEST_SERVICE_DIR
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/data/valid-service-files"
#define TEST_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-service"
#define TEST_SHELL_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-shell-service"
#define TEST_CORE_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/core/test-service-glib"
#define TEST_INTERFACES_SERVICE_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/interfaces/test-service"
#define TEST_EXIT_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-exit"
#define TEST_SEGFAULT_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-segfault"
#define TEST_SLEEP_FOREVER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2/test/test-sleep-forever"
#define DBUS_TEST_SOCKET_DIR ""
```

configure: exit 0

File = config.log.~1~

This file contains any messages produced by compilers while running configure, to aid debugging if configure makes a mistake.

It was created by dbus configure 1.6.8, which was generated by GNU Autoconf 2.69. Invocation command line was

```
$ /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/configure --build=i686-linux
--host=arm-oe-linux-gnueabi --target=arm-oe-linux-gnueabi --
prefix=/usr --exec_prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin -
-libexecdir=/usr/lib/dbus --datadir=/usr/share --sysconfdir=/etc --
```

```
sharedstatedir=/com --localstatedir=/var --libdir=/usr/lib --
includedir=/usr/include --oldincludedir=/usr/include --
infodir=/usr/share/info --mandir=/usr/share/man --disable-silent-rules
--disable-dependency-tracking --with-libtool-
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 --disable-tests --disable-checks
--disable-xml-docs --disable-doxygen-docs --disable-libaudit --with-
xml=expat --disable-systemd --without-systemdsystemunitdir --without-x
--enable-nls
```

```
## ----- ##
## Platform. ##
## ----- ##
```

```
hostname = pc45
uname -m = i686
uname -r = 3.11.0-15-generic
uname -s = Linux
uname -v = #25~precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014
```

```
/usr/bin/uname -p = unknown
/bin/uname -X = unknown
```

```
/bin/arch = unknown
/usr/bin/arch -k = unknown
/usr/convex/getsysinfo = unknown
/usr/bin/hostinfo = unknown
/bin/machine = unknown
/usr/bin/oslevel = unknown
/bin/universe = unknown
```

```
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/oe-core/scripts
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/bin/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1/usr/bin/crossscripts
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/sbin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/usr/bin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux/sbin
PATH: /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/i686-linux//bin
PATH: /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/bin
PATH: /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/bin
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/oe-core/scripts
PATH: /home/gangadhar/newyoctobuild/tisdk/sources/bitbake/bin
PATH: /home/gangadhar/bin
```

```
PATH: /usr/lib/lightdm/lightdm
PATH: /usr/local/sbin
PATH: /usr/local/bin
PATH: /usr/sbin
PATH: /usr/bin
PATH: /sbin
PATH: /bin
PATH: /usr/games
PATH: /home/gangadhar/delphi/ti-sdk-am335x-evm-07.00.00.00/linux-
devkit/sysroots/i686-arago-linux/usr/bin
```

```
## ----- ##
## Core tests. ##
## ----- ##
```

```
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/meta-openembedded/meta-
oe/site/endian-little
| # rp-pppoe
| rppoe_cv_pack_bitfields=${rppoe_cv_pack_bitfields=rev}
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/endian-
little
| ac_cv_c_littleendian=${ac_cv_c_littleendian=yes}
| ac_cv_c_bigendian=${ac_cv_c_bigendian=no}
|
| # libnet
| ac_cv_libnet_endianness=${ac_cv_libnet_endianness=lil}
|
| # libmemcached
| ac_cv_c_endian=little
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/arm-
common
| # general
| ac_cv_va_val_copy=${ac_cv_va_val_copy=yes}
|
ac_cv_func_lstat_dereferences_slashed_symlink=${ac_cv_func_lstat_deref
erences_slashed_symlink=yes}
|
ac_cv_func_lstat_empty_string_bug=${ac_cv_func_lstat_empty_string_bug=
no}
|
ac_cv_func_stat_empty_string_bug=${ac_cv_func_stat_empty_string_bug=no
}
|
ac_cv_func_stat_ignores_trailing_slash=${ac_cv_func_stat_ignores_trail
ing_slash=no}
| ac_cv_header_netinet_sctp_h=${ac_cv_header_netinet_sctp_h=no}
|
ac_cv_header_netinet_sctp_uio_h=${ac_cv_header_netinet_sctp_uio_h=no}
```

```
| ac_cv_sctp=${ac_cv_sctp=no}
|
| # apache
| ac_cv_func_pthread_key_delete=${ac_cv_func_pthread_key_delete=yes}
| apr_cv_process_shared_works=${apr_cv_process_shared_works=no}
| ac_cv_sizeof_ssize_t=${ac_cv_sizeof_ssize_t=4}
| apr_cv_tcp_nodelay_with_cork=${apr_cv_tcp_nodelay_with_cork=yes}
|
| # bash
| ac_cv_c_long_double=${ac_cv_c_long_double=yes}
| bash_cv_func_sigsetjmp=${bash_cv_func_sigsetjmp=missing}
|
| # coreutils
| utils_cv_sys_open_max=${utils_cv_sys_open_max=1019}
|
| # cvs
| cvs_cv_func_printf_ptr=${cvs_cv_func_printf_ptr=yes}
|
| # db (sleepycat)
| db_cv_fcntl_f_setfd=${db_cv_fcntl_f_setfd=yes}
| db_cv_sprintf_count=${db_cv_sprintf_count=yes}
| db_cv_align_t=${db_cv_align_t='unsigned long long'}
| db_cv_alignp_t=${db_cv_alignp_t='unsigned long'}
|
| # D-BUS
| ac_cv_func_posix_getpwnam_r=${ac_cv_func_posix_getpwnam_r=yes}
|
| # edb
| db_cv_spinlocks=${db_cv_spinlocks=no}
|
| # ettercap
| ettercap_cv_type_socklen_t=${ettercap_cv_type_socklen_t=yes}
|
| # fget
| compat_cv_func_snprintf_works=${compat_cv_func_snprintf_works=yes}
| compat_cv_func_basename_works=${compat_cv_func_basename_works=no}
| compat_cv_func_dirname_works=${compat_cv_func_dirname_works=no}
|
| # fnmatch
| ac_cv_func_fnmatch_works=${ac_cv_func_fnmatch_works=yes}
|
| # gettext
| am_cv_func_working_getline=${am_cv_func_working_getline=yes}
|
| # glib
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_sizeof_system_thread=${glib_cv_sizeof_system_thread=4}
| glib_cv_stack_grows=${glib_cv_stack_grows=no}
| glib_cv_uscore=${glib_cv_uscore=no}
| glib_cv_use_pid_surrogate=${glib_cv_use_pid_surrogate=yes}
| glib_cv_has__inline=${glib_cv_has__inline=yes}
| glib_cv_has__inline__=${glib_cv_has__inline__=yes}
```



```

| glib_cv_hasinline=${glib_cv_hasinline=yes}
| glib_cv_sane_realloc=${glib_cv_sane_realloc=yes}
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_uscore=${glib_cv_uscore=no}
| glib_cv_va_val_copy=${glib_cv_va_val_copy=yes}
| glib_cv_rtlldglobal_broken=${glib_cv_rtlldglobal_broken=no}
|
glib_cv_sys_pthread_mutex_trylock_posix=${glib_cv_sys_pthread_mutex_tr
ylock_posix=yes}
|
glib_cv_sys_pthread_getspecific_posix=${glib_cv_sys_pthread_getspecifi
c_posix=yes}
|
glib_cv_sys_pthread_cond_timedwait_posix=${glib_cv_sys_pthread_cond_ti
medwait_posix=yes}
|
| # glib-2.0
| glib_cv_long_long_format=${glib_cv_long_long_format=ll}
| glib_cv_sizeof_gmutex=${glib_cv_sizeof_gmutex=24}
| glib_cv_sizeof_intmax_t=${glib_cv_sizeof_intmax_t=8}
| glib_cv_sizeof_ptrdiff_t=${glib_cv_sizeof_ptrdiff_t=4}
| glib_cv_sizeof_size_t=${glib_cv_sizeof_size_t=4}
| glib_cv_sizeof_system_thread=${glib_cv_sizeof_system_thread=4}
|
glib_cv_sys_use_pid_niceness_surrogate=${glib_cv_sys_use_pid_niceness_
surrogate=yes}
| ac_cv_alignof_guint32=4
| ac_cv_alignof_guint64=8
| ac_cv_alignof_unsigned_long=4
|
| #gstreamer
| as_cv_unaligned_access=${as_cv_unaligned_access=no}
|
| # httpcc
| ac_cv_strerror_r_SUSv3=${ac_cv_strerror_r_SUSv3=no}
|
| # jikes
| ac_cv_sizeof_wchar_t=4
|
| # lftp
| ac_cv_need_trio=${ac_cv_need_trio=no}
| lftp_cv_va_val_copy=${lftp_cv_va_val_copy=yes}
|
| # libesmtplib
| acx_working_snprintf=${acx_working_snprintf=yes}
|
| # libidl
| libIDL_cv_long_long_format=${libIDL_cv_long_long_format=ll}
|
| # libnet
| ac_libnet_have_packet_socket=${ac_libnet_have_packet_socket=yes}
|

```

```
| # libpcap
| ac_cv_linux_vers=${ac_cv_linux_vers=2}
|
| # libxfce4util
| with_broken_putenv=${with_broken_putenv=no}
|
| # links
|
ac_cv_lib_png_png_create_info_struct=${ac_cv_lib_png_png_create_info_struct=yes}
|
| # mysql
| mysql_cv_func_atomic_sub=${mysql_cv_func_atomic_sub=no}
| mysql_cv_func_atomic_add=${mysql_cv_func_atomic_add=no}
|
| # nano
|
ac_cv_regexexec_segfault_emptystr=${ac_cv_regexexec_segfault_emptystr=no}
|
nano_cv_func_regexexec_segv_emptystr=${nano_cv_func_regexexec_segv_emptystr=no}
|
| # ORBit2
| ac_cv_alignof_CORBA_boolean=1
| ac_cv_alignof_CORBA_char=1
| ac_cv_alignof_CORBA_double=8
| ac_cv_alignof_CORBA_float=4
| ac_cv_alignof_CORBA_long=4
| ac_cv_alignof_CORBA_long_double=8
| ac_cv_alignof_CORBA_long_long=8
| ac_cv_alignof_CORBA_octet=1
| ac_cv_alignof_CORBA_pointer=4
| ac_cv_alignof_CORBA_short=2
| ac_cv_alignof_CORBA_struct=1
| ac_cv_alignof_CORBA_wchar=2
|
| # php
| ac_cv_pread=${ac_cv_pread=no}
| ac_cv_pwrite=${ac_cv_pwrite=no}
|
php_cv_lib_cookie_io_functions_use_off64_t=${php_cv_lib_cookie_io_functions_use_off64_t=yes}
|
| # rsync
| rsync_cv_HAVE_BROKEN_LARGEFILE=${rsync_cv_HAVE_BROKEN_LARGEFILE=no}
| rsync_cv_HAVE_SOCKETPAIR=${rsync_cv_HAVE_SOCKETPAIR=yes}
| rsync_cv_HAVE_LONGLONG=${rsync_cv_HAVE_LONGLONG=yes}
| rsync_cv_HAVE_OFF64_T=${rsync_cv_HAVE_OFF64_T=no}
| rsync_cv_HAVE_SHORT_INO_T=${rsync_cv_HAVE_SHORT_INO_T=no}
| rsync_cv_HAVE_UNSIGNED_CHAR=${rsync_cv_HAVE_UNSIGNED_CHAR=no}
| rsync_cv_HAVE_BROKEN_READDIR=${rsync_cv_HAVE_BROKEN_READDIR=no}
| rsync_cv_HAVE_GETTIMEOFDAY_TZ=${rsync_cv_HAVE_GETTIMEOFDAY_TZ=yes}
```

```

| rsync_cv_HAVE_C99_VSNPRINTF=${rsync_cv_HAVE_C99_VSNPRINTF=yes}
| rsync_cv_HAVE_SECURE_MKSTEMP=${rsync_cv_HAVE_SECURE_MKSTEMP=yes}
| rsync_cv_REPLACE_INET_NTOA=${rsync_cv_REPLACE_INET_NTOA=no}
| rsync_cv_REPLACE_INET_ATON=${rsync_cv_REPLACE_INET_ATON=no}
|
| # screen
| screen_cv_sys_bcopy_overlap=${screen_cv_sys_bcopy_overlap=no}
| screen_cv_sys_memcpy_overlap=${screen_cv_sys_memcpy_overlap=no}
| screen_cv_sys_memmove_overlap=${screen_cv_sys_memmove_overlap=no}
| screen_cv_sys_fifo_broken_impl=${screen_cv_sys_fifo_broken_impl=yes}
| screen_cv_sys_fifo_usable=${screen_cv_sys_fifo_usable=yes}
|
screen_cv_sys_select_broken_retval=${screen_cv_sys_select_broken_retval=no}
| screen_cv_sys_sockets_nofs=${screen_cv_sys_sockets_nofs=no}
| screen_cv_sys_sockets_usable=${screen_cv_sys_sockets_usable=yes}
| screen_cv_sys_terminfo_used=${screen_cv_sys_terminfo_used=yes}
|
| # slrn
| slrn_cv_va_val_copy=${slrn_cv_va_val_copy=yes}
|
| # socat
| ac_cv_ispeed_offset=${ac_cv_ispeed_offset=13}
| sc_cv_termios_ispeed=${sc_cv_termios_ispeed=yes}
|
| # ssh
|
ac_cv_have_space_d_name_in_struct_dirent=${ac_cv_dirent_have_space_d_name=yes}
| ac_cv_have_broken_snprintf=${ac_cv_have_broken_snprintf=no}
| ac_cv_have_accrights_in_msghdr=${ac_cv_have_accrights_in_msghdr=no}
| ac_cv_have_control_in_msghdr=${ac_cv_have_control_in_msghdr=yes}
| ac_cv_have_openpty_ctty_bug=${ac_cv_have_openpty_ctty_bug=yes}
|
| # startup-notification
| lf_cv_sane_realloc=yes
|
| # xffm
| jm_cv_func_working_readdir=yes
|
| # evolution-data-server
| ac_cv_libiconv_utf8=${ac_cv_libiconv_utf8=yes}
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/common-linux
| # apr
| ac_cv_file__dev_zero=${ac_cv_file__dev_zero=yes}
| ac_cv_sizeof_pid_t=${ac_cv_sizeof_pid_t=4}
|
| # samba
| samba_cv_HAVE_IFACE_AIX=${samba_cv_HAVE_IFACE_AIX=no}
| samba_cv_HAVE_IFACE_IFCONF=${samba_cv_HAVE_IFACE_IFCONF=yes}

```

```
| samba_cv_HAVE_IFACE_IFREQ=${samba_cv_HAVE_IFACE_IFREQ=yes}
|
samba_cv_HAVE_IRIX_SPECIFIC_CAPABILITIES=${samba_cv_HAVE_IRIX_SPECIFIC
_CAPABILITIES=no}
|
| # db
| db_cv_path_ar=${db_cv_path_ar=/usr/bin/ar}
| db_cv_path_chmod=${db_cv_path_chmod=/bin/chmod}
| db_cv_path_cp=${db_cv_path_cp=/bin/cp}
| db_cv_path_ln=${db_cv_path_ln=/bin/ln}
| db_cv_path_mkdir=${db_cv_path_mkdir=/bin/mkdir}
| db_cv_path_ranlib=${db_cv_path_ranlib=/usr/bin/ranlib}
| db_cv_path_rm=${db_cv_path_rm=/bin/rm}
| db_cv_path_sh=${db_cv_path_sh=/bin/sh}
| db_cv_path_strip=${db_cv_path_strip=/usr/bin/strip}
|
| # bash
| bash_cv_have_mbstate_t=${bash_cv_have_mbstate_t=yes}
|
bash_cv_must_reinstall_sighandlers=${bash_cv_must_reinstall_sighandler
s=no}
| bash_cv_func_strcoll_broken=${bash_cv_func_strcoll_broken=no}
| bash_cv_dup2_broken=${bash_cv_dup2_broken=no}
| bash_cv_opendir_not_robust=${bash_cv_opendir_not_robust=no}
| bash_cv_type_rlimit=${bash_cv_type_rlimit=rlim_t}
| bash_cv_getenv_redef=${bash_cv_getenv_redef=yes}
| bash_cv_ulimit_maxfds=${bash_cv_ulimit_maxfds=yes}
| bash_cv_getcwd_calls_popen=${bash_cv_getcwd_calls_popen=no}
| bash_cv_printf_a_format=${bash_cv_printf_a_format=yes}
| bash_cv_pgrp_pipe=${bash_cv_pgrp_pipe=no}
| bash_cv_job_control_missing=${bash_cv_job_control_missing=present}
| bash_cv_sys_named_pipes=${bash_cv_sys_named_pipes=present}
| bash_cv_unusable_rtsigs=${bash_cv_unusable_rtsigs=no}
|
| # coreutils
| gl_cv_func_fstatat_zero_flag=${gl_cv_func_fstatat_zero_flag=yes}
|
| # mysql
| ac_cv_sys_restartable_syscalls=${ac_cv_sys_restartable_syscalls=yes}
| ac_cv_conv_longlong_to_float=${ac_cv_conv_longlong_to_float=yes}
|
| # TCL
| tcl_cv_api_serial=${tcl_cv_api_serial=termios}
|
| # python
| ac_cv_have_long_long_format=yes
|
| # apache
| ac_cv_o_nonblock_inherited=${ac_cv_o_nonblock_inherited=no}
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/common-
glibc
```

```

| # general
| ac_cv_have_decl_sys_siglist=${ac_cv_have_decl_sys_siglist=yes}
| ac_cv_func_realloc_works=${ac_cv_func_realloc_works=yes}
| ac_cv_func_realloc_0_nonnull=${ac_cv_func_realloc_0_nonnull=yes}
| ac_cv_func_malloc_works=${ac_cv_func_malloc_works=yes}
| ac_cv_func_malloc_0_nonnull=${ac_cv_func_malloc_0_nonnull=yes}
| ac_cv_func_memcmp_working=${ac_cv_func_memcmp_working=yes}
| ac_cv_func_getpgrp_void=yes
| ac_cv_func_setpgrp_void=yes
| ac_cv_func_setgrent_void=yes
| ac_cv_func_getgrgid_r=${ac_cv_func_getgrgid_r=yes}
| ac_cv_func_getpwuid_r=${ac_cv_func_getpwuid_r=yes}
| ac_cv_func_posix_getpwuid_r=${ac_cv_func_posix_getpwuid_r=yes}
| ac_cv_func_posix_getgrgid_r=${ac_cv_func_posix_getgrgid_r=yes}
| ac_cv_type_uid_t=${ac_cv_type_uid_t=yes}
| ac_cv_func_getaddrinfo=${ac_cv_func_getaddrinfo=yes}
|
| # bash
| bash_cv_under_sys_siglist=${bash_cv_under_sys_siglist=yes}
| bash_cv_sys_siglist=${bash_cv_sys_siglist=yes}
|
| # clamav
|
clamav_av_func_working_snprintf_long=${clamav_av_func_working_snprintf
_long=yes}
| clamav_av_have_in_port_t=${clamav_av_have_in_port_t=yes}
| clamav_av_have_in_addr_t=${clamav_av_have_in_addr_t=yes}
| ac_cv_func_mmap_fixed_mapped=${ac_cv_func_mmap_fixed_mapped=yes}
|
| # coreutils
| fu_cv_sys_stat_statfs2_bsize=${fu_cv_sys_stat_statfs2_bsize=yes}
|
| # glib
| glib_cv_strncpy=${glib_cv_strncpy=no}
| ac_cv_func_printf_unix98=${ac_cv_func_printf_unix98=yes}
| ac_cv_func_snprintf_c99=${ac_cv_func_snprintf_c99=yes}
| ac_cv_func_vsnprintf_c99=${ac_cv_func_vsnprintf_c99=yes}
|
glib_cv_compliant_posix_memalign=${glib_cv_compliant_posix_memalign=1}
| glib_cv_long_long_format=${glib_cv_long_long_format=ll}
| # glib-1.x requires this and pkgconfig-native / pkgconfig-nativesdk
use
| # that to avoid a dependency loop.
| glib_cv__va_copy=${glib_cv__va_copy=yes}
| glib_cv_va_copy=${glib_cv_va_copy=yes}
| glib_cv_have_qsort_r=${glib_cv_have_qsort_r=yes}
|
| #dbus-glib
| ac_cv_have_abstract_sockets=${ac_cv_have_abstract_sockets=yes}
|
| # gnuCash
| am_cv_scanf_lld=${am_cv_scanf_lld=yes}

```

```

|
| # guile
|
ac_cv_func_pthread_attr_getstack=${ac_cv_func_pthread_attr_getstack=yes}
|
| #gcc-zlib
| ac_cv_func_getpagesize=${ac_cv_func_getpagesize=yes}
| ac_cv_func_memcpy=${ac_cv_func_memcpy=yes}
| ac_cv_func_strerror=${ac_cv_func_strerror=yes}
|
| # squid
| ac_cv_af_unix_large_dgram=${ac_cv_af_unix_large_dgram=yes}
| ac_cv_func_setresuid=${ac_cv_func_setresuid=yes}
| ac_cv_func_va_copy=${ac_cv_func_va_copy=yes}
| ac_cv_func__va_copy=${ac_cv_func__va_copy=yes}
| ac_cv_epoll_works=${ac_cv_epoll_works=yes}
|
| ac_cv_check_sj1j=ss1j
|
| # m4
| gt_cv_locale_fr=${gt_cv_locale_fr=fr_FR}
| gl_cv_func_btowc_eof=${gl_cv_func_btowc_eof=yes}
| gl_cv_func_wcrtomb_retval=${gl_cv_func_wcrtomb_retval=yes}
| gl_cv_func_wctob_works=${gl_cv_func_wctob_works=yes}
|
gl_cv_func_mbrtowc_incomplete_state=${gl_cv_func_mbrtowc_incomplete_state=yes}
| gl_cv_func_mbrtowc_sanitycheck=${gl_cv_func_mbrtowc_sanitycheck=yes}
| gl_cv_func_mbrtowc_null_arg=${gl_cv_func_mbrtowc_null_arg=yes}
| gl_cv_func_mbrtowc_retval=${gl_cv_func_mbrtowc_retval=yes}
| gl_cv_func_mbrtowc_nul_retval=${gl_cv_func_mbrtowc_nul_retval=yes}
|
| # va_copy and __va_copy
| ac_cv_va_copy=${ac_cv_va_copy=yes}
| ac_cv__va_copy=${ac_cv__va_copy=yes}
| ac_cv_func_va_copy=${ac_cv_func_va_copy=yes}
| ac_cv_func__va_copy=${ac_cv_func__va_copy=yes}
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdsk/sources/oe-core/meta/site/arm-
linux
| ac_cv_func_setvbuf_reversed=no
| ac_cv_sizeof__int64=${ac_cv_sizeof__int64=0}
| ac_cv_sizeof_char=${ac_cv_sizeof_char=1}
| ac_cv_sizeof_wchar_t=${ac_cv_sizeof_wchar_t=1}
| ac_cv_sizeof_unsigned_char=${ac_cv_sizeof_unsigned_char=1}
| ac_cv_sizeof_bool=${ac_cv_sizeof_bool=1}
| ac_cv_sizeof_char_p=${ac_cv_sizeof_int_p=4}
| ac_cv_sizeof_int=${ac_cv_sizeof_int=4}
| ac_cv_sizeof_int_p=${ac_cv_sizeof_int_p=4}
| ac_cv_sizeof_long=${ac_cv_sizeof_long=4}
| ac_cv_sizeof_long_int=${ac_cv_sizeof_long_int=4}

```

```

| ac_cv_sizeof_long_long=${ac_cv_sizeof_long_long=8}
| ac_cv_sizeof_off_t=${ac_cv_sizeof_off_t=4}
| ac_cv_sizeof_short=${ac_cv_sizeof_short=2}
| ac_cv_sizeof_short_int=${ac_cv_sizeof_short_int=2}
| ac_cv_sizeof_size_t=${ac_cv_sizeof_size_t=4}
| ac_cv_sizeof_void_p=${ac_cv_sizeof_void_p=4}
| ac_cv_sizeof_long_p=${ac_cv_sizeof_long_p=4}
| ac_cv_sizeof_float=${ac_cv_sizeof_float=4}
| ac_cv_sizeof_double=${ac_cv_sizeof_double=8}
| ac_cv_sizeof_long_double=${ac_cv_sizeof_long_double=8}
| ac_cv_sizeof_ptrdiff_t=${glib_cv_sizeof_ptrdiff_t=4}
| ac_cv_sizeof_unsigned_short=${ac_cv_sizeof_unsigned_short=2}
| ac_cv_sizeof_unsigned=${ac_cv_sizeof_unsigned=4}
| ac_cv_sizeof_unsigned_int=${ac_cv_sizeof_unsigned_int=4}
| ac_cv_sizeof_unsigned_long=${ac_cv_sizeof_unsigned_long=4}
| ac_cv_sizeof_unsigned_long_long=${ac_cv_sizeof_unsigned_long_long=8}
| ac_cv_sizeof_signed_char=${ac_cv_sizeof_signed_char=1}
|
| ac_cv_uchar=${ac_cv_uchar=no}
| ac_cv_uint=${ac_cv_uint=yes}
| ac_cv_ulong=${ac_cv_ulong=yes}
| ac_cv_ushort=${ac_cv_ushort=yes}
| ac_cv_time_r_type=${ac_cv_time_r_type=POSIX}
|
| # samba
|
samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=${samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=yes}
|
samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=${samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=no}
|
samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=${samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=no}
| samba_cv_HAVE_BROKEN_GETGROUPS=${samba_cv_HAVE_BROKEN_GETGROUPS=no}
|
samba_cv_HAVE_BROKEN_LINUX_SENDFILE=${samba_cv_HAVE_BROKEN_LINUX_SENDFILE=yes}
| samba_cv_HAVE_BROKEN_READDIR=${samba_cv_HAVE_BROKEN_READDIR=yes}
|
samba_cv_HAVE_BROKEN_READDIR_NAME=${samba_cv_HAVE_BROKEN_READDIR_NAME=no}
|
samba_cv_HAVE_C99_VSNPRINTF=${samba_cv_HAVE_C99_VSNPRINTF=yes}
| samba_cv_HAVE_DEV64_T=${samba_cv_HAVE_DEV64_T=no}
| samba_cv_HAVE_DEVICE_MAJOR_FN=${samba_cv_HAVE_DEVICE_MAJOR_FN=yes}
| samba_cv_HAVE_DEVICE_MINOR_FN=${samba_cv_HAVE_DEVICE_MINOR_FN=yes}
| samba_cv_HAVE_DQB_FSOFTLIMIT=${samba_cv_HAVE_DQB_FSOFTLIMIT=no}
|
samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=${samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=yes}
| samba_cv_HAVE_FAM_H=${samba_cv_HAVE_FAM_H=no}
| samba_cv_HAVE_FCNTL_LOCK=${samba_cv_HAVE_FCNTL_LOCK=yes}

```

```
| samba_cv_HAVE_FTRUNCATE_EXTEND=${samba_cv_HAVE_FTRUNCATE_EXTEND=yes}
| samba_cv_HAVE_FUNCTION_MACRO=${samba_cv_HAVE_FUNCTION_MACRO=yes}
| samba_cv_HAVE_GETTIMEOFDAY_TZ=${samba_cv_HAVE_GETTIMEOFDAY_TZ=yes}
| samba_cv_HAVE_INO64_T=${samba_cv_HAVE_INO64_T=no}
|
samba_cv_HAVE_INT16_FROM_RPC_RPC_H=${samba_cv_HAVE_INT16_FROM_RPC_RPC_
H=no}
|
samba_cv_HAVE_INT32_FROM_RPC_RPC_H=${samba_cv_HAVE_INT32_FROM_RPC_RPC_
H=no}
|
samba_cv_HAVE_KERNEL_CHANGE_NOTIFY=${samba_cv_HAVE_KERNEL_CHANGE_NOTIF
Y=yes}
|
samba_cv_HAVE_KERNEL_OPLOCKS_LINUX=${samba_cv_HAVE_KERNEL_OPLOCKS_LINU
X=yes}
|
samba_cv_HAVE_KERNEL_SHARE_MODES=${samba_cv_HAVE_KERNEL_SHARE_MODES=ye
s}
| samba_cv_HAVE_MAKEDEV=${samba_cv_HAVE_MAKEDEV=yes}
| samba_cv_HAVE_MMAP=${samba_cv_HAVE_MMAP=yes}
| samba_cv_HAVE_NATIVE_ICONV=${samba_cv_HAVE_NATIVE_ICONV=yes}
| samba_cv_HAVE_OFF64_T=${samba_cv_HAVE_OFF64_T=no}
| samba_cv_HAVE_ROOT=${samba_cv_HAVE_ROOT=yes}
|
samba_cv_HAVE_RPC_AUTH_ERROR_CONFLICT=${samba_cv_HAVE_RPC_AUTH_ERROR_C
ONFLICT=no}
| samba_cv_HAVE_SECURE_MKSTEMP=${samba_cv_HAVE_SECURE_MKSTEMP=yes}
| samba_cv_HAVE_SENDFILE=${samba_cv_HAVE_SENDFILE=yes}
| samba_cv_HAVE_SENDFILE64=${samba_cv_HAVE_SENDFILE64=yes}
| samba_cv_HAVE_SOCKET_SIN_LEN=${samba_cv_HAVE_SOCKET_SIN_LEN=no}
| samba_cv_HAVE_STAT_ST_BLKSIZE=${samba_cv_HAVE_STAT_ST_BLKSIZE=yes}
| samba_cv_HAVE_STAT_ST_BLOCKS=${samba_cv_HAVE_STAT_ST_BLOCKS=yes}
| samba_cv_HAVE_STRUCT_DIR64=${samba_cv_HAVE_STRUCT_DIR64=no}
| samba_cv_HAVE_STRUCT_DIRENT64=${samba_cv_HAVE_STRUCT_DIRENT64=yes}
| samba_cv_HAVE_STRUCT_FLOCK64=${samba_cv_HAVE_STRUCT_FLOCK64=yes}
| samba_cv_HAVE_TRUNCATED_SALT=${samba_cv_HAVE_TRUNCATED_SALT=no}
|
samba_cv_HAVE_UINT16_FROM_RPC_RPC_H=${samba_cv_HAVE_UINT16_FROM_RPC_RP
C_H=no}
|
samba_cv_HAVE_UINT32_FROM_RPC_RPC_H=${samba_cv_HAVE_UINT32_FROM_RPC_RP
C_H=no}
| samba_cv_HAVE_UNSIGNED_CHAR=${samba_cv_HAVE_UNSIGNED_CHAR=yes}
| samba_cv_HAVE_UTIMBUF=${samba_cv_HAVE_UTIMBUF=yes}
| samba_cv_HAVE_UT_UT_ADDR=${samba_cv_HAVE_UT_UT_ADDR=yes}
| samba_cv_HAVE_UT_UT_EXIT=${samba_cv_HAVE_UT_UT_EXIT=yes}
| samba_cv_HAVE_UT_UT_HOST=${samba_cv_HAVE_UT_UT_HOST=yes}
| samba_cv_HAVE_UT_UT_ID=${samba_cv_HAVE_UT_UT_ID=yes}
| samba_cv_HAVE_UT_UT_NAME=${samba_cv_HAVE_UT_UT_NAME=yes}
| samba_cv_HAVE_UT_UT_PID=${samba_cv_HAVE_UT_UT_PID=yes}
| samba_cv_HAVE_UT_UT_TIME=${samba_cv_HAVE_UT_UT_TIME=yes}
```



```

| samba_cv_HAVE_UT_UT_TV=${samba_cv_HAVE_UT_UT_TV=yes}
| samba_cv_HAVE_UT_UT_TYPE=${samba_cv_HAVE_UT_UT_TYPE=yes}
| samba_cv_HAVE_UT_UT_USER=${samba_cv_HAVE_UT_UT_USER=yes}
| samba_cv_HAVE_UX_UT_SYSLLEN=${samba_cv_HAVE_UX_UT_SYSLLEN=no}
| samba_cv_HAVE_VA_COPY=${samba_cv_HAVE_VA_COPY=yes}
| samba_cv_HAVE_WORKING_AF_LOCAL=${samba_cv_HAVE_WORKING_AF_LOCAL=yes}
| samba_cv_HAVE_Werror=${samba_cv_HAVE_Werror=yes}
|
samba_cv_PUTUTLINE_RETURNS_UTMP=${samba_cv_PUTUTLINE_RETURNS_UTMP=yes}
| samba_cv_QUOTA_WORKS=${samba_cv_QUOTA_WORKS=yes}
| samba_cv_REALPATH_TAKES_NULL=${samba_cv_REALPATH_TAKES_NULL=yes}
| samba_cv_REPLACE_GETPASS=${samba_cv_REPLACE_GETPASS=yes}
| samba_cv_REPLACE_INET_NTOA=${samba_cv_REPLACE_INET_NTOA=no}
| samba_cv_REPLACE_READDIR=${samba_cv_REPLACE_READDIR=no}
| samba_cv_RUN_QUOTA_TESTS=${samba_cv_RUN_QUOTA_TESTS=yes}
| samba_cv_SEEKDIR_RETURNS_VOID=${samba_cv_SEEKDIR_RETURNS_VOID=yes}
| samba_cv_SIZEOF_DEV_T=${samba_cv_SIZEOF_DEV_T=yes}
| samba_cv_SIZEOF_INO_T=${samba_cv_SIZEOF_INO_T=yes}
| samba_cv_SIZEOF_OFF_T=${samba_cv_SIZEOF_OFF_T=yes}
|
samba_cv_SYSCONF_SC_NGROUPS_MAX=${samba_cv_SYSCONF_SC_NGROUPS_MAX=yes}
|
samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=${samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=yes}
| samba_cv_SYSCONF_SC_NPROC_ONLN=${samba_cv_SYSCONF_SC_NPROC_ONLN=no}
| samba_cv_SYSCONF_SC_PAGESIZE=${samba_cv_SYSCONF_SC_PAGESIZE=yes}
| samba_cv_SYSQUOTA_FOUND=${samba_cv_SYSQUOTA_FOUND=yes}
| samba_cv_SYSQUOTA_WORKS=${samba_cv_SYSQUOTA_WORKS=yes}
| samba_cv_SYSQUOTA_WORKS_XFS=${samba_cv_SYSQUOTA_WORKS_XFS=yes}
| samba_cv_TRY_QUOTAS=${samba_cv_TRY_QUOTAS=no}
| samba_cv_TRY_SYS_QUOTAS=${samba_cv_TRY_SYS_QUOTAS=yes}
| samba_cv_USE_SETRESUID=${samba_cv_USE_SETRESUID=yes}
| samba_cv_WE_USE_SYS_QUOTAS=${samba_cv_WE_USE_SYS_QUOTAS=yes}
| samba_cv_WITH_AFS=${samba_cv_WITH_AFS=no}
| samba_cv_WITH_FAKE_KASERVER=${samba_cv_WITH_FAKE_KASERVER=no}
| samba_cv_WITH_QUOTAS=${samba_cv_WITH_QUOTAS=auto}
| samba_cv_WITH_SYS_QUOTAS=${samba_cv_WITH_SYS_QUOTAS=auto}
| samba_cv_WITH_VFS_AFSACL=${samba_cv_WITH_VFS_AFSACL=no}
| samba_cv_compiler_supports_ll=${samba_cv_compiler_supports_ll=yes}
| samba_cv_found_xfs_header=${samba_cv_found_xfs_header=yes}
| samba_cv_have_longlong=${samba_cv_have_longlong=yes}
| samba_cv_have_setresgid=${samba_cv_have_setresgid=yes}
| samba_cv_have_setresuid=${samba_cv_have_setresuid=yes}
| samba_cv_immediate_structures=${samba_cv_immediate_structures=yes}
|
samba_cv_optimize_out_funcation_calls=${samba_cv_optimize_out_funcation_calls=yes}
| samba_cv_sig_atomic_t=${samba_cv_sig_atomic_t=yes}
| samba_cv_socklen_t=${samba_cv_socklen_t=yes}
| samba_cv_struct_timespec=${samba_cv_struct_timespec=yes}

```

```

|
samba_cv_sysquotas_file=${samba_cv_sysquotas_file=lib/sysquotas_linux.
c}
| samba_cv_unixsocket=${samba_cv_unixsocket=yes}
| samba_cv_volatile=${samba_cv_volatile=yes}
|
| #older sambe defines
| samba_cv_USE_SETEUID=${samba_cv_USE_SETEUID=yes}
| samba_cv_USE_SETREUID=${samba_cv_USE_SETREUID=yes}
| samba_cv_USE_SETUIDX=${samba_cv_USE_SETUIDX=yes}
| samba_cv_LINUX_LFS_SUPPORT=${samba_cv_LINUX_LFS_SUPPORT=yes}
|
| # clamav
|
clamav_av_func_working_snprintf_long=${clamav_av_func_working_snprintf
_long=yes}
| clamav_av_have_in_port_t=${clamav_av_have_in_port_t=yes}
| clamav_av_have_in_addr_t=${clamav_av_have_in_addr_t=yes}
| ac_cv_func_mmap_fixed_mapped=${ac_cv_func_mmap_fixed_mapped=yes}
|
| #dbus
| ac_cv_have_abstract_sockets=${ac_cv_have_abstract_sockets=yes}
|
| # lftp
| ac_cv_file__dev_ptc_=yes
|
| # guile
|
ac_cv_func_pthread_attr_getstack=${ac_cv_func_pthread_attr_getstack=ye
s}
|
| # gnet
|
ac_cv_member_struct_sockaddr_sa_len=${ac_cv_member_struct_sockaddr_sa_
len=no}
|
ac_cv_gnet_have_abstract_sockets=${ac_cv_gnet_have_abstract_sockets=no
}
|
gnet_sockaddr_family_field_name=${gnet_sockaddr_family_field_name=ss_f
amily}
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/sources/oe-core/meta/site/common
| # libelf
| mr_cv_target_elf=${mr_cv_target_elf=yes}
|
| # Normally kernels have /dev/random enabled
| ac_cv_file__dev_random=${ac_cv_file__dev_random=yes}
|
| # Avoid sudo to assume void for unsetenv in cross environment, or
else it conflicts with

```

```
| # target stdlib.h prototype which follows POSIX compliance. Mark
for upstream.
| sudo_cv_func_unsetenv_void=no
|
| # shadow dir info, to avoid searching build system
| shadow_cv_maildir=${localstatedir}/spool/mail
| shadow_cv_mailfile=Mailbox
| shadow_cv_utmpdir=${localstatedir}/run
| shadow_cv_logdir=${localstatedir}/log
| shadow_cv_passwd_dir=${bindir}
|
| # python: deactivate a runtime check for ipv6-support in python
>=2.7.1 that fails when cross-compiling
| ac_cv_buggy_getaddrinfo=no
configure:2767: loading site script
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/share/arm-oe-linux-
gnueabi_config_site.d/ncurses_config
| ac_cv_header_X=${ac_cv_header_X=no}
| ac_cv_header__CC=${ac_cv_header__CC=no}
| ac_cv_header__CFLAGS=${ac_cv_header__CFLAGS=no}
| ac_cv_header__CPPFLAGS=${ac_cv_header__CPPFLAGS=no}
| ac_cv_header__5=${ac_cv_header__5=no}
| ac_cv_header__c=${ac_cv_header__c=no}
| ac_cv_header_confptest__ac_ext=${ac_cv_header_confptest__ac_ext=no}
| ac_cv_header_curses_h=${ac_cv_header_curses_h=yes}
| ac_cv_header_inttypes_h=${ac_cv_header_inttypes_h=yes}
| ac_cv_header_memory_h=${ac_cv_header_memory_h=yes}
| ac_cv_header_ncurses_curses_h=${ac_cv_header_ncurses_curses_h=no}
| ac_cv_header_ncurses_h=${ac_cv_header_ncurses_h=yes}
| ac_cv_header_ncurses_termcap_h=${ac_cv_header_ncurses_termcap_h=no}
| ac_cv_header_stdclib=${ac_cv_header_stdclib=yes}
| ac_cv_header_stdint_h=${ac_cv_header_stdint_h=yes}
| ac_cv_header_stdlib_h=${ac_cv_header_stdlib_h=yes}
| ac_cv_header_string_h=${ac_cv_header_string_h=yes}
| ac_cv_header_strings_h=${ac_cv_header_strings_h=yes}
| ac_cv_header_sys_stat_h=${ac_cv_header_sys_stat_h=yes}
| ac_cv_header_sys_types_h=${ac_cv_header_sys_types_h=yes}
| ac_cv_header_unistd_h=${ac_cv_header_unistd_h=yes}
configure:2897: checking build system type
configure:2911: result: i686-pc-linux-gnu
configure:2931: checking host system type
configure:2944: result: arm-oe-linux-gnueabi
configure:2985: checking for a BSD-compatible install
configure:3053: result: /usr/bin/install -c
configure:3064: checking whether build environment is sane
configure:3119: result: yes
configure:3178: checking for arm-oe-linux-gnueabi-strip
configure:3205: result: arm-linux-gnueabi-strip
configure:3270: checking for a thread-safe mkdir -p
configure:3309: result: /bin/mkdir -p
configure:3316: checking for gawk
```

```
configure:3332: found /usr/bin/gawk
configure:3343: result: gawk
configure:3354: checking whether make sets $(MAKE)
configure:3376: result: yes
configure:3457: checking how to create a ustar tar archive
configure:3470: tar --version
tar (GNU tar) 1.26
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Written by John Gilmore and Jay Fenlason.

```
configure:3473: $? = 0
configure:3513: tardir=confptest.dir && eval tar --format=ustar -chf -
"$stardir" >confptest.tar
configure:3516: $? = 0
configure:3520: tar -xf - <confptest.tar
configure:3523: $? = 0
configure:3536: result: gnutar
configure:3555: checking whether to enable maintainer-specific
portions of Makefiles
configure:3564: result: yes
configure:3589: checking whether make supports nested variables
configure:3606: result: yes
configure:3663: checking for arm-oe-linux-gnueabi-gcc
configure:3690: result: arm-linux-gnueabi-gcc -march=armv7-a -marm
-mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:3959: checking for C compiler version
configure:3968: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 --version >&5
arm-linux-gnueabi-gcc (crosstool-NG linaro-1.13.1-4.7-2013.03-
20130313 - Linaro GCC 2013.03) 4.7.3 20130226 (prerelease)
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

```
configure:3979: $? = 0
configure:3968: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -v >&5
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-gcc
```

```
COLLECT_LTO_WRAPPER=/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/bin/./libexec/gcc/arm-linux-gnueabi/4.7.3/lto-wrapper
Target: arm-linux-gnueabi
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/src/gcc-linaro-4.7-2013.03/configure --build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-linux-gnueabi --prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install --with-sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install/arm-linux-gnueabi/libc --enable-languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-pkgversion='crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03' --with-bugurl=https://bugs.launchpad.net/gcc-linaro --enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-libssp --with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-ppl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-cloog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static --with-host-libstdcxx='-L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/.build/arm-linux-gnueabi/build/static/lib -lpwl' --enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-id --enable-gold --with-local-prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-linux/install/arm-linux-gnueabi/libc --enable-c99 --enable-long-long --with-mode=thumb
Thread model: posix
gcc version 4.7.3 20130226 (prerelease) (crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03)
configure:3979: $? = 0
configure:3968: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -V >&5
arm-linux-gnueabi-gcc: error: unrecognized command line option '-V'
arm-linux-gnueabi-gcc: fatal error: no input files
compilation terminated.
configure:3979: $? = 1
configure:3968: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -qversion >&5
arm-linux-gnueabi-gcc: error: unrecognized command line option '-qversion'
```

```
arm-linux-gnueabi-gcc: fatal error: no input files
compilation terminated.
configure:3979: $? = 1
configure:3999: checking whether the C compiler works
configure:4021: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfp=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu
confptest.c >&5
configure:4025: $? = 0
configure:4073: result: yes
configure:4076: checking for C compiler default output file name
configure:4078: result: a.out
configure:4084: checking for suffix of executables
configure:4091: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfp=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c >&5
configure:4095: $? = 0
configure:4117: result:
configure:4139: checking whether we are cross compiling
configure:4177: result: yes
configure:4182: checking for suffix of object files
configure:4204: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfp=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector confptest.c >&5
configure:4208: $? = 0
configure:4229: result: o
configure:4233: checking whether we are using the GNU C compiler
```

```
configure:4252: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:4252: $? = 0
configure:4261: result: yes
configure:4270: checking whether arm-linux-gnueabihf-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 accepts -g
configure:4290: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -g -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector
conftest.c >&5
configure:4290: $? = 0
configure:4331: result: yes
configure:4348: checking for arm-linux-gnueabihf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to accept ISO C89
configure:4411: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:4411: $? = 0
configure:4424: result: none needed
configure:4455: checking for style of include used by make
configure:4483: result: GNU
configure:4509: checking dependency style of arm-linux-gnueabihf-gcc
-march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:4620: result: none
configure:4636: checking whether arm-linux-gnueabihf-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
```

```
linaro-toolchain/sysroots/delphi-vp1 and cc understand -c and -o
together
configure:4667: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c confptest.c -o confptest2.o >&5
configure:4671: $? = 0
configure:4677: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c confptest.c -o confptest2.o >&5
configure:4681: $? = 0
configure:4692: cc -c confptest.c >&5
configure:4696: $? = 0
configure:4704: cc -c confptest.c -o confptest2.o >&5
configure:4708: $? = 0
configure:4714: cc -c confptest.c -o confptest2.o >&5
configure:4718: $? = 0
configure:4736: result: yes
configure:4873: checking for C++ compiler version
configure:4882: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 --version >&5
arm-linux-gnueabi-hf-g++ (crosstool-NG linaro-1.13.1-4.7-2013.03-
20130313 - Linaro GCC 2013.03) 4.7.3 20130226 (prerelease)
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.

configure:4893: $? = 0
configure:4882: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -v >&5
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabi-hf-g++
COLLECT_LTO_WRAPPER=/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/bin/./libexec/gcc/arm-linux-
gnueabi-hf/4.7.3/lto-wrapper
Target: arm-linux-gnueabi-hf
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-hf-linux/.build/src/gcc-linaro-4.7-2013.03/configure --
build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --
target=arm-linux-gnueabi-hf --prefix=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-linux/install --with-
sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/install/arm-linux-gnueabi-hf/libc --enable-
languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-
tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard --with-
```



```
pkgversion='crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro
GCC 2013.03' --with-bugurl=https://bugs.launchpad.net/gcc-linaro --
enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-
libssp --with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-
gnueabi-hf-linux/.build/arm-linux-gnueabi-hf/build/static --with-
mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
ppl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
clog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-
libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static --with-host-libstdcxx='-
L/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabi-hf-
linux/.build/arm-linux-gnueabi-hf/build/static/lib -lpwl' --enable-
threads=posix --disable-libstdcxx-pch --enable-linker-build-id --
enable-gold --with-local-prefix=/cbuild/slaves/oorts/crosstool-
ng/builds/arm-linux-gnueabi-hf-linux/install/arm-linux-gnueabi-hf/libc -
enable-c99 --enable-long-long --with-mode=thumb
Thread model: posix
gcc version 4.7.3 20130226 (prerelease) (crosstool-NG linaro-1.13.1-
4.7-2013.03-20130313 - Linaro GCC 2013.03)
configure:4893: $? = 0
configure:4882: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -V >&5
arm-linux-gnueabi-hf-g++: error: unrecognized command line option '-V'
arm-linux-gnueabi-hf-g++: fatal error: no input files
compilation terminated.
configure:4893: $? = 1
configure:4882: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -qversion >&5
arm-linux-gnueabi-hf-g++: error: unrecognized command line option '-
qversion'
arm-linux-gnueabi-hf-g++: fatal error: no input files
compilation terminated.
configure:4893: $? = 1
configure:4897: checking whether we are using the GNU C++ compiler
configure:4916: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -fpermissive -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector confctest.cpp >&5
```

```
configure:4916: $? = 0
configure:4925: result: yes
configure:4934: checking whether arm-linux-gnueabi-g++ -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 accepts -g
configure:4954: arm-linux-gnueabi-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -g -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-g++-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-g++/include -fstack-protector
confptest.cpp >&5
configure:4954: $? = 0
configure:4995: result: yes
configure:5020: checking dependency style of arm-linux-gnueabi-g++
-march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:5131: result: none
configure:5152: checking how to run the C preprocessor
configure:5222: result: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8
configure:5242: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-g++-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-g++/include -fstack-protector
confptest.c
configure:5242: $? = 0
configure:5256: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-g++-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-g++/include -fstack-protector
confptest.c
confptest.c:13:28: fatal error: ac_nonexistent.h: No such file or
directory
compilation terminated.
configure:5256: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
```

```

| #define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| /* end confdefs.h. */
| #include <ac_nonexistent.h>
figure:5285: checking for grep that handles long lines and -e
figure:5343: result: /bin/grep
figure:5348: checking for egrep
figure:5410: result: /bin/grep -E
figure:5415: checking for ANSI C header files
figure:5519: result: yes
figure:5532: checking for sys/types.h
figure:5532: result: yes
figure:5532: checking for sys/stat.h
figure:5532: result: yes
figure:5532: checking for stdlib.h
figure:5532: result: yes
figure:5532: checking for string.h
figure:5532: result: yes
figure:5532: checking for memory.h
figure:5532: result: yes
figure:5532: checking for strings.h
figure:5532: result: yes
figure:5532: checking for inttypes.h
figure:5532: result: yes
figure:5532: checking for stdint.h
figure:5532: result: yes
figure:5532: checking for unistd.h
figure:5532: result: yes
figure:5545: checking minix/config.h usability
figure:5545: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
conftest.c:56:26: fatal error: minix/config.h: No such file or
directory
compilation terminated.
figure:5545: $? = 1
figure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"

```

```

| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| # include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| # include <memory.h>
| # endif
| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif

```

```
| #include <minix/config.h>
configure:5545: result: no
configure:5545: checking minix/config.h presence
configure:5545: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
confptest.c:23:26: fatal error: minix/config.h: No such file or
directory
compilation terminated.
configure:5545: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <minix/config.h>
configure:5545: result: no
configure:5545: checking for minix/config.h
configure:5545: result: no
configure:5566: checking whether it is safe to define __EXTENSIONS__
configure:5584: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confptest.c >&5
```

```
configure:5584: $? = 0
configure:5591: result: yes
configure:5605: checking for library containing strerror
configure:5636: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:5636: $? = 0
configure:5653: result: none required
configure:5661: checking for ANSI C header files
configure:5765: result: yes
configure:5773: checking for inline
configure:5789: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:5789: $? = 0
configure:5797: result: inline
configure:5861: checking how to print strings
configure:5888: result: printf
configure:5909: checking for a sed that does not truncate output
configure:5973: result: /bin/sed
configure:5991: checking for fgrep
configure:6053: result: /bin/grep -F
configure:6088: checking for ld used by arm-linux-gnueabi-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:6155: result: arm-linux-gnueabi-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:6162: checking if the linker (arm-linux-gnueabi-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1) is GNU ld
configure:6177: result: yes
configure:6189: checking for BSD- or MS-compatible name lister (nm)
configure:6238: result: arm-linux-gnueabi-nm
```

configure:6368: checking the name lister (arm-linux-gnueabi-hf-nm) interface
configure:6375: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector conftest.c >&5
configure:6378: arm-linux-gnueabi-hf-nm "conftest.o"
configure:6381: output
00000000 B some_variable
configure:6388: result: BSD nm
configure:6391: checking whether ln -s works
configure:6395: result: yes
configure:6403: checking the maximum length of command line arguments
configure:6533: result: 1572864
configure:6550: checking whether the shell understands some XSI constructs
configure:6560: result: yes
configure:6564: checking whether the shell understands "+="
configure:6570: result: yes
configure:6605: checking how to convert i686-pc-linux-gnu file names to arm-oe-linux-gnueabi format
configure:6645: result: func_convert_file_noop
configure:6652: checking how to convert i686-pc-linux-gnu file names to toolchain format
configure:6672: result: func_convert_file_noop
configure:6679: checking for arm-linux-gnueabi-hf-ld --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 option to reload object files
configure:6686: result: -r
configure:6720: checking for arm-oe-linux-gnueabi-objdump
configure:6747: result: arm-linux-gnueabi-hf-objdump
configure:6819: checking how to recognize dependent libraries
configure:7021: result: pass_all
configure:7066: checking for arm-oe-linux-gnueabi-dlltool
configure:7096: result: no
configure:7106: checking for dlltool
configure:7136: result: no
configure:7166: checking how to associate runtime and link libraries
configure:7193: result: printf %s\n
configure:7209: checking for arm-oe-linux-gnueabi-ar
configure:7236: result: arm-linux-gnueabi-hf-ar
configure:7317: checking for archiver @FILE support
configure:7334: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-

```
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector confptest.c >&5
figure:7334: $? = 0
figure:7337: arm-linux-gnueabihf-ar cru libconfptest.a @confptest.lst
>&5
figure:7340: $? = 0
figure:7345: arm-linux-gnueabihf-ar cru libconfptest.a @confptest.lst
>&5
arm-linux-gnueabihf-ar: confptest.o: No such file or directory
figure:7348: $? = 1
figure:7360: result: @
figure:7378: checking for arm-oe-linux-gnueabi-strip
figure:7405: result: arm-linux-gnueabihf-strip
figure:7477: checking for arm-oe-linux-gnueabi-ranlib
figure:7504: result: arm-linux-gnueabihf-ranlib
figure:7646: checking command to parse arm-linux-gnueabihf-nm
output from arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 object
figure:7766: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector confptest.c >&5
figure:7769: $? = 0
figure:7773: arm-linux-gnueabihf-nm confptest.o \|| sed -n -e 's/^[^*]
*\|([ABCDGIRSTW][ABCDGIRSTW]*)\| [ ]*\|([_A-Za-z][_A-Za-z0-9]*)\|$/\|1 \|2
\|/p' | sed '/__gnu_lto/d' \> confptest.nm
figure:7776: $? = 0
figure:7842: arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c conftstm.o >&5
figure:7845: $? = 0
figure:7883: result: ok
figure:7920: checking for sysroot
```



```
configure:7950: result:
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1
configure:8166: checking for arm-oe-linux-gnueabi-mt
configure:8196: result: no
configure:8206: checking for mt
configure:8222: found /bin/mt
configure:8233: result: mt
configure:8245: WARNING: using cross tools not prefixed with host
triplet
configure:8256: checking if mt is a manifest tool
configure:8262: mt '-?'
configure:8270: result: no
configure:8909: checking for dlfcn.h
configure:8909: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:8909: $? = 0
configure:8909: result: yes
configure:9127: checking for objdir
configure:9142: result: .libs
configure:9409: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 supports -fno-rtti -fno-
exceptions
configure:9427: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector -fno-rtti -fno-exceptions conftest.c >&5
ccl: warning: command line option '-fno-rtti' is valid for C++/ObjC++
but not for C [enabled by default]
configure:9431: $? = 0
configure:9444: result: no
configure:9771: checking for arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to produce PIC
configure:9778: result: -fPIC -DPIC
configure:9786: checking if arm-linux-gnueabi-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```

```
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl PIC flag -fPIC -DPIC works
configure:9804: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector -fPIC -DPIC -DPIC conftest.c >&5
configure:9808: $? = 0
configure:9821: result: yes
configure:9850: checking if arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl static flag -static works
configure:9878: result: yes
configure:9893: checking if arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl supports -c -o file.o
configure:9914: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector -o out/conftest2.o conftest.c >&5
configure:9918: $? = 0
configure:9940: result: yes
configure:9948: checking if arm-linux-gnueabi-hf-gcc -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl supports -c -o file.o
configure:9995: result: yes
configure:10028: checking whether the arm-linux-gnueabi-hf-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl linker (arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl) supports shared libraries
configure:11181: result: yes
configure:11218: checking whether -lc should be explicitly linked in
configure:11226: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
```

```
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:11229: $? = 0
configure:11244: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -shared -fPIC -DPIC conftest.o
-v -Wl,-soname -Wl,conftest -o conftest 2\>\&1 \ | /bin/grep -lc
\>/dev/null 2\>\&1
configure:11247: $? = 0
configure:11261: result: no
configure:11421: checking dynamic linker characteristics
configure:11932: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu -Wl,-rpath -Wl,/foo conftest.c >&5
configure:11932: $? = 0
configure:12154: result: GNU/Linux ld.so
configure:12261: checking how to hardcode library paths into programs
configure:12286: result: immediate
configure:12826: checking whether stripping libraries is possible
configure:12831: result: yes
configure:12866: checking if libtool supports shared libraries
configure:12868: result: yes
configure:12871: checking whether to build shared libraries
configure:12892: result: yes
configure:12895: checking whether to build static libraries
configure:12899: result: yes
configure:12922: checking how to run the C++ preprocessor
configure:12949: arm-linux-gnueabihf-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -E -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector conftest.cpp
configure:12949: $? = 0
configure:12963: arm-linux-gnueabihf-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -E -isystem/home/gangadhar/gcc-
```

```

linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector confptest.cpp
confptest.cpp:31:28: fatal error: ac_nonexistent.h: No such file or
directory
compilation terminated.
configure:12963: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| /* end confdefs.h.  */
| #include <ac_nonexistent.h>
configure:12988: result: arm-linux-gnueabi-g++ -march=armv7-a -marm
-mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -E
configure:13008: arm-linux-gnueabi-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -E -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector confptest.cpp
configure:13008: $? = 0
configure:13022: arm-linux-gnueabi-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --

```

```

sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -E -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector confctest.cpp
confctest.cpp:31:28: fatal error: ac_nonexistent.h: No such file or
directory
compilation terminated.
configure:13022: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| /* end confdefs.h. */
| #include <ac_nonexistent.h>
configure:13191: checking for ld used by arm-linux-gnueabi-hf-g++ -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:13258: result: arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1
configure:13265: checking if the linker (arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1) is GNU ld

```

```
configure:13280: result: yes
configure:13335: checking whether the arm-linux-gnueabi-hf-g++ -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 linker (arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1) supports shared libraries
configure:14340: result: yes
configure:14376: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -fpermissive -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector confctest.cpp >&5
configure:14379: $? = 0
configure:14899: checking for arm-linux-gnueabi-hf-g++ -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
-sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 option to produce PIC
configure:14906: result: -fPIC -DPIC
configure:14914: checking if arm-linux-gnueabi-hf-g++ -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 PIC flag -fPIC -DPIC works
configure:14932: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -fpermissive -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector -fPIC -DPIC -DPIC confctest.cpp >&5
configure:14936: $? = 0
configure:14949: result: yes
configure:14972: checking if arm-linux-gnueabi-hf-g++ -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 static flag -static works
configure:15000: result: yes
configure:15012: checking if arm-linux-gnueabi-hf-g++ -march=armv7-a -
marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 supports -c -o file.o
configure:15033: arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
```

```
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -fpermissive -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -o out/conftest2.o conftest.cpp >&5
configure:15037: $? = 0
configure:15059: result: yes
configure:15064: checking if arm-linux-gnueabi-g++ -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl supports -c -o file.o
configure:15111: result: yes
configure:15141: checking whether the arm-linux-gnueabi-g++ -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl linker (arm-linux-gnueabi-ld --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl) supports shared libraries
configure:15177: result: yes
configure:15318: checking dynamic linker characteristics
configure:15985: result: GNU/Linux ld.so
configure:16038: checking how to hardcode library paths into programs
configure:16063: result: immediate
configure:16147: checking for arm-oe-linux-gnueabi-pkg-config
configure:16180: result: no
configure:16190: checking for pkg-config
configure:16208: found
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config
configure:16220: result:
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config
configure:16245: checking pkg-config is at least version 0.9.0
configure:16248: result: yes
configure:16263: checking for arm-oe-linux-gnueabi-windres
configure:16293: result: no
configure:16303: checking for windres
configure:16333: result: no
configure:17410: checking for a Python interpreter with version >= 2.6
configure:17427: python -c import sys # split strings by '.' and convert to numeric. Append some zeros # because we need at least 4 digits for the hex conversion. # map returns an iterator in Python 3.0 and a list in 2.x minver = list(map(int, '2.6'.split('.'))) + [0, 0, 0] minverhex = 0 # xrange is not present in Python 3.0 and range returns an iterator for i in list(range(0, 4)): minverhex = (minverhex << 8) + minver[i] sys.exit(sys.hexversion < minverhex)
configure:17430: $? = 0
configure:17436: result: python
configure:17444: checking for python
configure:17462: found /usr/bin/python
configure:17474: result: /usr/bin/python
```

```
configure:17492: checking for python version
configure:17499: result: 2.7
configure:17513: checking for python platform
configure:17520: result: linux2
configure:17546: checking for python script directory
configure:17582: result: ${prefix}/lib/python2.7/dist-packages
configure:17591: checking for python extension module directory
configure:17627: result: ${exec_prefix}/lib/python2.7/dist-packages
configure:17699: checking whether we are using the GNU C Library 2.1
or newer
configure:17726: result: yes
configure:17735: checking size of char
configure:17754: result: 1
configure:17768: checking size of short
configure:17787: result: 2
configure:17801: checking size of long
configure:17820: result: 4
configure:17834: checking size of int
configure:17853: result: 4
configure:17867: checking size of void *
configure:17886: result: 4
configure:17900: checking size of long long
configure:17919: result: 8
configure:17933: checking size of __int64
configure:17952: result: 0
configure:17973: checking 64-bit integer type
configure:18047: result: long long
configure:18058: checking 32-bit integer type
configure:18078: result: int
configure:18085: checking 16-bit integer type
configure:18102: result: short
configure:18116: checking whether byte ordering is bigendian
configure:18322: result: no
configure:18343: checking for an implementation of va_copy()
configure:18366: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18366: $? = 0
configure:18375: result: yes
configure:18377: checking for an implementation of __va_copy()
configure:18400: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```



```
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18400: $? = 0
configure:18409: result: yes
configure:18433: checking whether va_lists can be copied by value
configure:18479: result: yes
configure:18497: checking whether arm-linux-gnueabi-gcc -
march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 knows __sync_sub_and_fetch()
configure:18514: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
conftest.c: In function 'main':
conftest.c:51:49: warning: incompatible implicit declaration of built-
in function 'exit' [enabled by default]
configure:18514: $? = 0
configure:18523: result: yes
configure:18539: checking for library containing socket
configure:18570: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
```

```
configure:18570: $? = 0
configure:18587: result: none required
configure:18595: checking for gethostbyname
configure:18595: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18595: $? = 0
configure:18595: result: yes
configure:18650: checking for vsnprintf
configure:18650: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
conftest.c:70:6: warning: conflicting types for built-in function
'vsnprintf' [enabled by default]
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for vasprintf
configure:18650: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
```

```
configure:18650: checking for nanosleep
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for usleep
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for setenv
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for clearenv
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
```

```
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for unsetenv
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for socketpair
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for getgrouplist
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
```

```
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for fpathconf
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl -o conftest -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for setrlimit
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl -o conftest -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for poll
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl -o conftest -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
```

```
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for setlocale
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for localeconv
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for strtoll
configure:18650: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
```

```
configure:18650: checking for strtoull
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:18650: $? = 0
configure:18650: result: yes
configure:18650: checking for issetugid
configure:18650: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
/tmp/cc70K9wy.o: In function `main':
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:97: undefined reference to
`issetugid'
collect2: error: ld returned 1 exit status
configure:18650: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
```

```

| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOOLL 1
| #define HAVE_STRTOULL 1
| /* end confdefs.h. */
| /* Define issetugid to an innocuous variant, in case <limits.h>
declares issetugid.
|     For example, HP-UX lli <limits.h> declares gettimeofday. */

```



```

| #define issetugid innocuous_issetugid
|
| /* System header to define __stub macros and hopefully few
prototypes,
|     which can conflict with char issetugid (); below.
|     Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
|     <limits.h> exists even on freestanding compilers. */
|
| #ifdef __STDC__
| # include <limits.h>
| #else
| # include <assert.h>
| #endif
|
| #undef issetugid
|
| /* Override any GCC internal prototype to avoid an error.
|     Use char because int might match the return type of a GCC
|     builtin and then its argument prototype would still apply. */
| #ifdef __cplusplus
| extern "C"
| #endif
| char issetugid ();
| /* The GNU C library defines this for functions which it implements
|     to always fail with ENOSYS. Some functions are actually named
|     something starting with __ and the normal name is an alias. */
| #if defined __stub_issetugid || defined __stub__issetugid
| choke me
| #endif
|
| int
| main ()
| {
| return issetugid ();
| ;
| return 0;
| }
configure:18650: result: no
configure:18650: checking for getresuid
configure:18650: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5

```

```
configure:18650: $? = 0
configure:18650: result: yes
configure:18662: checking syslog.h usability
configure:18662: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:18662: $? = 0
configure:18662: result: yes
configure:18662: checking syslog.h presence
configure:18662: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector
conftest.c
configure:18662: $? = 0
configure:18662: result: yes
configure:18662: checking for syslog.h
configure:18662: result: yes
configure:18673: checking whether LOG_ERROR is declared
configure:18673: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:18673: $? = 0
configure:18673: result: yes
configure:18689: checking for broken poll
configure:18728: result: no (cross compiling)
configure:18731: checking for dirfd
configure:18752: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
```

```
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-  
style=gnu conftest.c >&5  
configure:18752: $? = 0  
configure:18759: result: yes  
configure:18806: checking sys/resource.h usability  
configure:18806: arm-linux-gnueabihf-gcc -march=armv7-a -marm -  
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-  
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-  
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-  
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-  
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-  
protector conftest.c >&5  
configure:18806: $? = 0  
configure:18806: result: yes  
configure:18806: checking sys/resource.h presence  
configure:18806: arm-linux-gnueabihf-gcc -E --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-  
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -  
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-  
20130313_linux/arm-linux-gnueabihf/include -fstack-protector  
conftest.c  
configure:18806: $? = 0  
configure:18806: result: yes  
configure:18806: checking for sys/resource.h  
configure:18806: result: yes  
configure:18819: checking dirent.h usability  
configure:18819: arm-linux-gnueabihf-gcc -march=armv7-a -marm -  
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-  
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-  
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-  
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-  
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-  
protector conftest.c >&5  
configure:18819: $? = 0  
configure:18819: result: yes  
configure:18819: checking dirent.h presence  
configure:18819: arm-linux-gnueabihf-gcc -E --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-  
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -  
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-  
20130313_linux/arm-linux-gnueabihf/include -fstack-protector  
conftest.c  
configure:18819: $? = 0  
configure:18819: result: yes  
configure:18819: checking for dirent.h  
configure:18819: result: yes
```



```
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
configure:18881: $? = 0
configure:18881: result: yes
configure:18881: checking for locale.h
configure:18881: result: yes
configure:18894: checking byteswap.h usability
configure:18894: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confptest.c >&5
configure:18894: $? = 0
configure:18894: result: yes
configure:18894: checking byteswap.h presence
configure:18894: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
configure:18894: $? = 0
configure:18894: result: yes
configure:18894: checking for byteswap.h
configure:18894: result: yes
configure:18907: checking for unistd.h
configure:18907: result: yes
configure:18920: checking ws2tcpip.h usability
configure:18920: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confptest.c >&5
confptest.c:109:22: fatal error: ws2tcpip.h: No such file or directory
compilation terminated.
configure:18920: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
```

```
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
```

```
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| # include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| # include <memory.h>
| # endif
| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
```



```
| #endif
| #include <ws2tcpip.h>
configure:18920: result: no
configure:18920: checking ws2tcpip.h presence
configure:18920: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector
confptest.c
confptest.c:76:22: fatal error: ws2tcpip.h: No such file or directory
compilation terminated.
configure:18920: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define __ALL_SOURCE 1
| #define __GNU_SOURCE 1
| #define __POSIX_PTHREAD_SEMANTICS 1
| #define __TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
```

```

| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF___INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <ws2tcpip.h>
configure:18920: result: no
configure:18920: checking for ws2tcpip.h
configure:18920: result: no
configure:18933: checking wspiapi.h usability
configure:18933: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-

```

```
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
conftest.c:109:21: fatal error: wspiapi.h: No such file or directory
compilation terminated.
configure:18933: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF___INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
```

```

| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| # include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| # include <memory.h>
| # endif
| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H

```

```

| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif
| #include <wspiapi.h>
configure:18933: result: no
configure:18933: checking wspiapi.h presence
configure:18933: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector
confptest.c
confptest.c:76:21: fatal error: wspiapi.h: No such file or directory
compilation terminated.
configure:18933: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1

```

```

| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| /* end confdefs.h. */
| #include <wspiapi.h>

```

```

configure:18933: result: no
configure:18933: checking for wsapi.h

```

```
configure:18933: result: no
configure:18962: checking for posix getpwnam_r
configure:19004: result: yes
configure:19054: checking whether socklen_t is defined
configure:19074: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:19074: $? = 0
configure:19080: result: yes
configure:19091: checking sys/uio.h usability
configure:19091: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:19091: $? = 0
configure:19091: result: yes
configure:19091: checking sys/uio.h presence
configure:19091: arm-linux-gnueabihf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector
conftest.c
configure:19091: $? = 0
configure:19091: result: yes
configure:19091: checking for sys/uio.h
configure:19091: result: yes
configure:19098: checking for writev
configure:19098: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
```

```
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-  
style=gnu conftest.c >&5  
configure:19098: $? = 0  
configure:19098: result: yes  
configure:19114: checking sys/syslimits.h usability  
configure:19114: arm-linux-gnueabi-gcc -march=armv7-a -marm -  
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-  
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-  
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-  
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-  
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-  
protector conftest.c >&5  
conftest.c:113:27: fatal error: sys/syslimits.h: No such file or  
directory  
compilation terminated.  
configure:19114: $? = 1  
configure: failed program was:  
| /* confdefs.h */  
| #define PACKAGE_NAME "dbus"  
| #define PACKAGE_TARNAME "dbus"  
| #define PACKAGE_VERSION "1.6.8"  
| #define PACKAGE_STRING "dbus 1.6.8"  
| #define PACKAGE_BUGREPORT  
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"  
| #define PACKAGE_URL ""  
| #define PACKAGE "dbus"  
| #define VERSION "1.6.8"  
| #define GETTEXT_PACKAGE "dbus-1"  
| #define DBUS_DAEMON_NAME "dbus-daemon"  
| #define STDC_HEADERS 1  
| #define HAVE_SYS_TYPES_H 1  
| #define HAVE_SYS_STAT_H 1  
| #define HAVE_STDLIB_H 1  
| #define HAVE_STRING_H 1  
| #define HAVE_MEMORY_H 1  
| #define HAVE_STRINGS_H 1  
| #define HAVE_INTTYPES_H 1  
| #define HAVE_STDINT_H 1  
| #define HAVE_UNISTD_H 1  
| #define __EXTENSIONS__ 1  
| #define _ALL_SOURCE 1  
| #define _GNU_SOURCE 1  
| #define _POSIX_PTHREAD_SEMANTICS 1  
| #define _TANDEM_SOURCE 1  
| #define STDC_HEADERS 1  
| #define HAVE_DLFCN_H 1  
| #define LT_OBJDIR ".libs/"  
| #define DBUS_UNIX 1  
| #define DBUS_WITH_GLIB 1  
| #define DBUS_DISABLE_ASSERT 1
```



```
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
```

```

| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| #   include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| #   include <memory.h>
| # endif
| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif
| #include <sys/syslimits.h>
configure:19114: result: no
configure:19114: checking sys/syslimits.h presence
configure:19114: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
confptest.c
confptest.c:80:27: fatal error: sys/syslimits.h: No such file or
directory
compilation terminated.
configure:19114: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"

```

```
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
```

```

| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| /* end confdefs.h. */
| #include <sys/syslimits.h>
configure:19114: result: no
configure:19114: checking for sys/syslimits.h
configure:19114: result: no
configure:19125: checking whether MSG_NOSIGNAL is declared
configure:19125: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:19125: $? = 0
configure:19125: result: yes
configure:19139: checking for ISO C99 varargs macros in C
configure:19156: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:19156: $? = 0
configure:19162: result: yes
configure:19165: checking for GNUC varargs macros
configure:19182: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-

```

```
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector conftest.c >&5
configure:19182: $? = 0
configure:19188: result: yes
configure:19202: checking for struct cmsgcred
configure:19222: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector conftest.c >&5
conftest.c: In function 'main':
conftest.c:91:17: error: storage size of 'cred' isn't known
configure:19222: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
```

```
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| /* end confdefs.h. */
```

```

|
| #include <sys/types.h>
| #include <sys/socket.h>
|
| int
| main ()
| {
|
| struct cmsgcred cred;
|
| cred.cmcred_pid = 0;
|
| ;
| return 0;
| }
configure:19228: result: no
configure:19240: checking for getpeerucred
configure:19240: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
/tmp/ccxqXAkN.o: In function `main':
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:117: undefined reference to
`getpeerucred'
collect2: error: ld returned 1 exit status
configure:19240: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1

```

```
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
```



```

| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| /* end confdefs.h.  */
| /* Define getpeerucrd to an innocuous variant, in case <limits.h>
declares getpeerucrd.
|   For example, HP-UX 11i <limits.h> declares gettimeofday.  */
| #define getpeerucrd innocuous_getpeerucrd
|
| /* System header to define __stub macros and hopefully few
prototypes,
|   which can conflict with char getpeerucrd (); below.
|   Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
|   <limits.h> exists even on freestanding compilers.  */
|
| #ifdef __STDC__
| # include <limits.h>
| #else
| # include <assert.h>
| #endif
|
| #undef getpeerucrd
|
| /* Override any GCC internal prototype to avoid an error.
|   Use char because int might match the return type of a GCC
|   builtin and then its argument prototype would still apply.  */
| #ifdef __cplusplus
| extern "C"
| #endif
| char getpeerucrd ();
| /* The GNU C library defines this for functions which it implements
|   to always fail with ENOSYS.  Some functions are actually named
|   something starting with __ and the normal name is an alias.  */
| #if defined __stub_getpeerucrd || defined __stub__getpeerucrd
| choke me
| #endif
|
| int
| main ()

```

```

| {
| return getpeerucred ();
| ;
| return 0;
| }
configure:19240: result: no
configure:19240: checking for getpeereid
configure:19240: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
/tmp/ccafMNX1.o: In function `main':
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:117: undefined reference to
`getpeereid'
collect2: error: ld returned 1 exit status
configure:19240: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1

```

```
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
```

```

| #define HAVE_SOCKLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| /* end confdefs.h.  */
| /* Define getpeereid to an innocuous variant, in case <limits.h>
declares getpeereid.
|   For example, HP-UX 11i <limits.h> declares gettimeofday.  */
| #define getpeereid innocuous_getpeereid
|
| /* System header to define __stub macros and hopefully few
prototypes,
|   which can conflict with char getpeereid (); below.
|   Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
|   <limits.h> exists even on freestanding compilers.  */
|
| #ifdef __STDC__
| # include <limits.h>
| #else
| # include <assert.h>
| #endif
|
| #undef getpeereid
|
| /* Override any GCC internal prototype to avoid an error.
|   Use char because int might match the return type of a GCC
|   builtin and then its argument prototype would still apply.  */
| #ifdef __cplusplus
| extern "C"
| #endif
| char getpeereid ();
| /* The GNU C library defines this for functions which it implements
|   to always fail with ENOSYS.  Some functions are actually named
|   something starting with __ and the normal name is an alias.  */
| #if defined __stub_getpeereid || defined __stub__getpeereid
| choke me
| #endif
|
| int
| main ()
| {
| return getpeereid ();
| ;
| return 0;
| }

```

configure:19240: result: no
configure:19253: checking for pipe2
configure:19253: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-

```
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
figure:19253: $? = 0
figure:19253: result: yes
figure:19253: checking for accept4
figure:19253: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
figure:19253: $? = 0
figure:19253: result: yes
figure:19273: checking abstract socket namespace
figure:19344: result: yes
figure:19387: checking for arm-oe-linux-gnueabi-pkg-config
figure:19417: result:
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/i686-linux/usr/bin/pkg-config
figure:19485: checking pkg-config is at least version 0.9.0
figure:19488: result: yes
figure:19501: checking for XML_ParserCreate_MM in -lexpat
figure:19526: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lexpat >&5
figure:19526: $? = 0
figure:19535: result: yes
```

```
configure:19540: checking expat.h usability
configure:19540: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector conftest.c >&5
configure:19540: $? = 0
configure:19540: result: yes
configure:19540: checking expat.h presence
configure:19540: arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
conftest.c
configure:19540: $? = 0
configure:19540: result: yes
configure:19540: checking for expat.h
configure:19540: result: yes
configure:19679: checking for pthread_cond_timedwait
configure:19679: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
configure:19679: $? = 0
configure:19679: result: yes
configure:19681: checking for pthread_cond_timedwait in -lpthread
configure:19706: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
```

```
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lpthread >&5
configure:19706: $? = 0
configure:19715: result: yes
configure:19725: checking for pthread_condattr_setclock
configure:19725: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lpthread >&5
configure:19725: $? = 0
configure:19725: result: yes
configure:19733: checking for library containing clock_getres
configure:19764: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lpthread >&5
/tmp/ccJ8UkJv.o: In function `main':
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:98: undefined reference to
`clock_getres'
collect2: error: ld returned 1 exit status
configure:19764: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
```

```
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
```



```

| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| /* end confdefs.h.  */
|
| /* Override any GCC internal prototype to avoid an error.
|    Use char because int might match the return type of a GCC
|    builtin and then its argument prototype would still apply.  */
| #ifdef __cplusplus
| extern "C"
| #endif
| char clock_getres ();
| int
| main ()
| {
| return clock_getres ();
| ;
| return 0;
| }
configure:19764: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-

```

```
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lrt -lpthread >&5
configure:19764: $? = 0
configure:19781: result: -lrt
configure:19789: checking for CLOCK_MONOTONIC
configure:19810: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-
protector conftest.c >&5
configure:19810: $? = 0
configure:19817: result: found
configure:19836: checking for is_selinux_enabled in -lselinux
configure:19861: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c -lselinux >&5
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/bin/../../lib/gcc/arm-linux-
gnueabihf/4.7.3/../../../../arm-linux-gnueabihf/bin/ld: cannot find -
lselinux
collect2: error: ld returned 1 exit status
configure:19861: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
```

```
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
```

```

| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| /* end confdefs.h.  */
|
| /* Override any GCC internal prototype to avoid an error.
|    Use char because int might match the return type of a GCC
|    builtin and then its argument prototype would still apply.  */
| #ifdef __cplusplus
| extern "C"
| #endif
| char is_selinux_enabled ();
| int
| main ()
| {
| return is_selinux_enabled ();
| ;
| return 0;
| }

```

configure:19870: result: no

configure:19909: WARNING: Sufficiently new SELinux library not found

configure:19992: checking sys/inotify.h usability

configure:19992: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confctest.c >&5

configure:19992: \$? = 0

configure:19992: result: yes

configure:19992: checking sys/inotify.h presence

```
configure:19992: arm-linux-gnueabihf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector
confptest.c
configure:19992: $? = 0
configure:19992: result: yes
configure:19992: checking for sys/inotify.h
configure:19992: result: yes
configure:20012: checking for inotify_init1
configure:20012: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c >&5
configure:20012: $? = 0
configure:20012: result: yes
configure:20071: checking for Linux epoll(4)
configure:20089: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -o confptest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-
2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-
style=gnu confptest.c >&5
configure:20089: $? = 0
configure:20096: result: yes
configure:20121: checking sys/event.h usability
configure:20121: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
```

```
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-  
protector conftest.c >&5  
conftest.c:125:23: fatal error: sys/event.h: No such file or directory  
compilation terminated.  
configure:20121: $? = 1  
configure: failed program was:  
| /* confdefs.h */  
| #define PACKAGE_NAME "dbus"  
| #define PACKAGE_TARNAME "dbus"  
| #define PACKAGE_VERSION "1.6.8"  
| #define PACKAGE_STRING "dbus 1.6.8"  
| #define PACKAGE_BUGREPORT  
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"  
| #define PACKAGE_URL ""  
| #define PACKAGE "dbus"  
| #define VERSION "1.6.8"  
| #define GETTEXT_PACKAGE "dbus-1"  
| #define DBUS_DAEMON_NAME "dbus-daemon"  
| #define STDC_HEADERS 1  
| #define HAVE_SYS_TYPES_H 1  
| #define HAVE_SYS_STAT_H 1  
| #define HAVE_STDLIB_H 1  
| #define HAVE_STRING_H 1  
| #define HAVE_MEMORY_H 1  
| #define HAVE_STRINGS_H 1  
| #define HAVE_INTTYPES_H 1  
| #define HAVE_STDINT_H 1  
| #define HAVE_UNISTD_H 1  
| #define __EXTENSIONS__ 1  
| #define _ALL_SOURCE 1  
| #define _GNU_SOURCE 1  
| #define _POSIX_PTHREAD_SEMANTICS 1  
| #define _TANDEM_SOURCE 1  
| #define STDC_HEADERS 1  
| #define HAVE_DLFCN_H 1  
| #define LT_OBJDIR ".libs/"  
| #define DBUS_UNIX 1  
| #define DBUS_WITH_GLIB 1  
| #define DBUS_DISABLE_ASSERT 1  
| #define DBUS_DISABLE_CHECKS 1  
| #define G_DISABLE_CHECKS 1  
| #define DBUS_ENABLE_USERDB_CACHE 1  
| #define SIZEOF_CHAR 1  
| #define SIZEOF_SHORT 2  
| #define SIZEOF_LONG 4  
| #define SIZEOF_INT 4  
| #define SIZEOF_VOID_P 4  
| #define SIZEOF_LONG_LONG 8  
| #define SIZEOF__INT64 0  
| #define DBUS_INT64_PRINTF_MODIFIER "ll"  
| #define DBUS_VA_COPY va_copy  
| #define DBUS_USE_SYNC 1
```

```
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
```

```

| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| #  include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| #  include <memory.h>
| # endif
| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif
| #include <sys/event.h>
configure:20121: result: no
configure:20121: checking sys/event.h presence
configure:20121: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector
confptest.c
confptest.c:92:23: fatal error: sys/event.h: No such file or directory
compilation terminated.
configure:20121: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"

```



```
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
```

```

| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h.  */
| #include <sys/event.h>
configure:20121: result: no
configure:20121: checking for sys/event.h
configure:20121: result: no
configure:20129: checking for kqueue
configure:20129: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -o conftest -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
/tmp/ccK6hC9P.o: In function `main':
/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:126: undefined reference to `kqueue'
collect2: error: ld returned 1 exit status

```

```
configure:20129: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
```

```

| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h. */
| /* Define kqueue to an innocuous variant, in case <limits.h>
declares kqueue.
|   For example, HP-UX 11i <limits.h> declares gettimeofday. */
| #define kqueue innocuous_kqueue
|
| /* System header to define __stub macros and hopefully few
prototypes,
|   which can conflict with char kqueue (); below.
|   Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
|   <limits.h> exists even on freestanding compilers. */

```

```

|
| #ifndef __STDC__
| # include <limits.h>
| #else
| # include <assert.h>
| #endif
|
| #undef kqueue
|
| /* Override any GCC internal prototype to avoid an error.
|    Use char because int might match the return type of a GCC
|    builtin and then its argument prototype would still apply. */
| #ifndef __cplusplus
| extern "C"
| #endif
| char kqueue ();
| /* The GNU C library defines this for functions which it implements
|    to always fail with ENOSYS. Some functions are actually named
|    something starting with __ and the normal name is an alias. */
| #if defined __stub_kqueue || defined __stub___kqueue
| choke me
| #endif
|
| int
| main ()
| {
| return kqueue ();
| ;
| return 0;
| }

```

configure:20129: result: no

configure:20162: checking launch.h usability

```

configure:20162: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector conftest.c >&5

```

conftest.c:125:20: fatal error: launch.h: No such file or directory
compilation terminated.

configure:20162: \$? = 1

configure: failed program was:

```

| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"

```

```
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
```

```

| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| # include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| # include <memory.h>
| # endif

```

```

| # include <string.h>
| #endif
| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif
| #include <launch.h>
configure:20162: result: no
configure:20162: checking launch.h presence
configure:20162: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc/include -fstack-protector
confptest.c
confptest.c:92:20: fatal error: launch.h: No such file or directory
compilation terminated.
configure:20162: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1

```



```
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
```

```

| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h. */
| #include <launch.h>
configure:20162: result: no
configure:20162: checking for launch.h
configure:20162: result: no
configure:20172: checking for launchctl
configure:20205: result: no
configure:20461: checking for ADT API
configure:20477: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
conftest.c:93:21: fatal error: bsm/adt.h: No such file or directory
compilation terminated.
configure:20477: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1

```

```
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
```

```

| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| /* end confdefs.h.  */

```

```

| #include <bsm/adt.h>
| adt_user_context = ADT_USER;
|
| int
| main ()
| {
|
|     ;
|     return 0;
| }

```

```

configure:20494: result: no
configure:20500: checking for SCM_RIGHTS
configure:20518: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vpl -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-
protector conftest.c >&5
configure:20518: $? = 0
configure:20519: result: supported
configure:20666: checking for X
configure:20768: result: disabled
configure:21339: checking to see if compiler understands

```

```
configure:21358: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector confptest.c >&5
configure:21358: $? = 0
configure:21374: result: yes
configure:21393: checking to see if compiler understands -Wall
configure:21412: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wall -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -Wextra
configure:21412: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wextra -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -Wchar-
subscripts
configure:21412: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wchar-subscripts -isystem/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include
-fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -Wmissing-
declarations
configure:21412: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
```



```
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wcast-align -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -
fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -Wno-address
configure:21412: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-address -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -
fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -Wfloat-equal
configure:21412: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wfloat-equal -isystem/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include
-fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21393: checking to see if compiler understands -
Wdeclaration-after-statement
configure:21412: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wdeclaration-after-statement -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector confptest.c >&5
configure:21412: $? = 0
configure:21428: result: yes
configure:21435: checking to see if compiler understands -Werror
configure:21454: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-
```



```
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-error=missing-field-initializers -
-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/include -fstack-protector
confptest.c >&5
configure:21540: $? = 0
configure:21556: result: yes
configure:21482: checking to see if compiler understands -Wno-unused-
parameter
configure:21501: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-unused-parameter -isystem/home/gangadhar/gcc-linaro-
arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector confptest.c >&5
configure:21501: $? = 0
configure:21517: result: yes
configure:21521: checking to see if compiler understands -Wno-
error=unused-parameter
configure:21540: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-error=unused-parameter -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector confptest.c >&5
configure:21540: $? = 0
configure:21556: result: yes
configure:21482: checking to see if compiler understands -Wno-sign-
compare
configure:21501: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-
gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-sign-compare -isystem/home/gangadhar/gcc-linaro-arm-
linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include
-fstack-protector confptest.c >&5
configure:21501: $? = 0
configure:21517: result: yes
configure:21521: checking to see if compiler understands -Wno-
error=sign-compare
configure:21540: arm-linux-gnueabihf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
```

```
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -Wno-error=sign-compare -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector confptest.c >&5
configure:21540: $? = 0
configure:21556: result: yes
configure:21482: checking to see if compiler understands -Wno-pointer-sign
configure:21501: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -Wno-pointer-sign -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector confptest.c >&5
configure:21501: $? = 0
configure:21517: result: yes
configure:21521: checking to see if compiler understands -Wno-error=pointer-sign
configure:21540: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -Wno-error=pointer-sign -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector confptest.c >&5
configure:21540: $? = 0
configure:21556: result: yes
configure:21482: checking to see if compiler understands -Wno-type-limits
configure:21501: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -Wno-type-limits -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector confptest.c >&5
configure:21501: $? = 0
configure:21517: result: yes
configure:21521: checking to see if compiler understands -Wno-error=type-limits
configure:21540: arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
```

```
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -Wno-error=type-limits -isystem/home/gangadhar/gcc-linaro-
arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector confptest.c >&5
configure:21540: $? = 0
configure:21556: result: yes
configure:21582: checking to see if compiler understands -fno-common
configure:21601: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -fno-common -isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector confptest.c >&5
configure:21601: $? = 0
configure:21617: result: yes
configure:21625: checking to see if compiler understands -fno-strict-
aliasing
configure:21644: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-
debug-types -fno-strict-aliasing -isystem/home/gangadhar/gcc-linaro-
arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector confptest.c >&5
configure:21644: $? = 0
configure:21660: result: yes
configure:21727: checking for doxygen
configure:21745: found /usr/bin/doxygen
configure:21758: result: /usr/bin/doxygen
configure:21767: checking whether to build Doxygen documentation
configure:21798: result: no
configure:21805: checking for xsltproc
configure:21821: found /usr/bin/xsltproc
configure:21832: result: xsltproc
configure:21856: checking for xmlto
configure:21887: result: no
configure:21896: checking whether to build XML documentation
configure:21927: result: no
configure:21932: checking for man2html
configure:21965: result: no
configure:22500: checking crt_externs.h usability
configure:22500: arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdtk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -c -Wall -Wextra -Wchar-
subscripts -Wmissing-declarations -Wmissing-prototypes -Wnested-
```

```
externs -Wpointer-arith -Wcast-align -Wno-address -Wfloat-equal -
Wdeclaration-after-statement -Wno-unused-label -Wno-missing-field-
initializers -Wno-unused-parameter -Wno-sign-compare -Wno-pointer-sign
-Wno-type-limits -fno-common -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-
gnueabi-hf/include -fstack-protector confptest.c >&5
confptest.c:144:25: fatal error: crt_externs.h: No such file or
directory
compilation terminated.
configure:22500: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
```

```
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
```

```

| #define DBUS_HAVE_LINUX_EPOLL 1
| #define HAVE_UNIX_FD_PASSING 1
| #define X_DISPLAY_MISSING 1
| #define DBUS_SYSTEM_SOCKET "/var/run/dbus/system_bus_socket"
| #define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"unix:path=/var/run/dbus/system_bus_socket"
| #define DBUS_CONSOLE_AUTH_DIR "/var/run/console/"
| #define DBUS_CONSOLE_OWNER_FILE ""
| #define DBUS_USER "messagebus"
| #define DBUS_PREFIX "/usr"
| #define DBUS_DATADIR "/usr/share"
| #define DBUS_DAEMONDIR "/usr/bin"
| #define DBUS_BINDIR "/usr/bin"
| #define DBUS_LIBEXECDIR "/usr/lib/dbus"
| #define DBUS_TEST_EXEC
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/test"
| #define DBUS_EXEEXT ""
| #define TEST_BUS_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon"
| #define DBUS_TEST_LAUNCH_HELPER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-launch-helper-test"
| #define DBUS_TEST_SOCKET_DIR "/tmp"
| #define TEST_LISTEN "unix:tmpdir=/tmp"
| #define DBUS_SESSION_SOCKET_DIR "/tmp"
| /* end confdefs.h. */
| #include <stdio.h>
| #ifdef HAVE_SYS_TYPES_H
| # include <sys/types.h>
| #endif
| #ifdef HAVE_SYS_STAT_H
| # include <sys/stat.h>
| #endif
| #ifdef STDC_HEADERS
| # include <stdlib.h>
| # include <stddef.h>
| #else
| # ifdef HAVE_STDLIB_H
| # include <stdlib.h>
| # endif
| #endif
| #ifdef HAVE_STRING_H
| # if !defined STDC_HEADERS && defined HAVE_MEMORY_H
| # include <memory.h>
| # endif
| # include <string.h>
| #endif

```

```

| #ifdef HAVE_STRINGS_H
| # include <strings.h>
| #endif
| #ifdef HAVE_INTTYPES_H
| # include <inttypes.h>
| #endif
| #ifdef HAVE_STDINT_H
| # include <stdint.h>
| #endif
| #ifdef HAVE_UNISTD_H
| # include <unistd.h>
| #endif
| #include <crt_externs.h>
configure:22500: result: no
configure:22500: checking crt_externs.h presence
configure:22500: arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector
confptest.c
confptest.c:111:25: fatal error: crt_externs.h: No such file or
directory
compilation terminated.
configure:22500: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1

```

```
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
```



```

| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| #define HAVE_UNIX_FD_PASSING 1
| #define X_DISPLAY_MISSING 1
| #define DBUS_SYSTEM_SOCKET "/var/run/dbus/system_bus_socket"
| #define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"unix:path=/var/run/dbus/system_bus_socket"
| #define DBUS_CONSOLE_AUTH_DIR "/var/run/console/"
| #define DBUS_CONSOLE_OWNER_FILE ""
| #define DBUS_USER "messagebus"
| #define DBUS_PREFIX "/usr"
| #define DBUS_DATADIR "/usr/share"
| #define DBUS_DAEMONDIR "/usr/bin"
| #define DBUS_BINDIR "/usr/bin"
| #define DBUS_LIBEXECDIR "/usr/lib/dbus"
| #define DBUS_TEST_EXEC
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/test"
| #define DBUS_EXEEXT ""
| #define TEST_BUS_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon"
| #define DBUS_TEST_LAUNCH_HELPER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-launch-helper-test"
| #define DBUS_TEST_SOCKET_DIR "/tmp"
| #define TEST_LISTEN "unix:tmpdir=/tmp"
| #define DBUS_SESSION_SOCKET_DIR "/tmp"
| /* end confdefs.h. */
| #include <crt_externs.h>
configure:22500: result: no
configure:22500: checking for crt_externs.h
configure:22500: result: no
configure:22510: checking for _NSGetEnviron
configure:22510: arm-linux-gnueabi-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-

```

```

linaro-toolchain/sysroots/delphi-vp1 -o conftest -Wall -Wextra -
Wchar-subscripts -Wmissing-declarations -Wmissing-prototypes -Wnested-
externs -Wpointer-arith -Wcast-align -Wno-address -Wfloat-equal -
Wdeclaration-after-statement -Wno-unused-label -Wno-missing-field-
initializers -Wno-unused-parameter -Wno-sign-compare -Wno-pointer-sign
-Wno-type-limits -fno-common -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -isystem/home/gangadhar/gcc-
linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-
gnueabi/include -fstack-protector -L/home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -
Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-
style=gnu conftest.c >&5
/tmp/ccQa7JEg.o: In function `main':
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/conftest.c:145: undefined reference to
`_NSGetEnviron'
collect2: error: ld returned 1 exit status
configure:22510: $? = 1
configure: failed program was:
| /* confdefs.h */
| #define PACKAGE_NAME "dbus"
| #define PACKAGE_TARNAME "dbus"
| #define PACKAGE_VERSION "1.6.8"
| #define PACKAGE_STRING "dbus 1.6.8"
| #define PACKAGE_BUGREPORT
| "https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
| #define PACKAGE_URL ""
| #define PACKAGE "dbus"
| #define VERSION "1.6.8"
| #define GETTEXT_PACKAGE "dbus-1"
| #define DBUS_DAEMON_NAME "dbus-daemon"
| #define STDC_HEADERS 1
| #define HAVE_SYS_TYPES_H 1
| #define HAVE_SYS_STAT_H 1
| #define HAVE_STDLIB_H 1
| #define HAVE_STRING_H 1
| #define HAVE_MEMORY_H 1
| #define HAVE_STRINGS_H 1
| #define HAVE_INTTYPES_H 1
| #define HAVE_STDINT_H 1
| #define HAVE_UNISTD_H 1
| #define __EXTENSIONS__ 1
| #define _ALL_SOURCE 1
| #define _GNU_SOURCE 1
| #define _POSIX_PTHREAD_SEMANTICS 1
| #define _TANDEM_SOURCE 1
| #define STDC_HEADERS 1
| #define HAVE_DLFCN_H 1

```

```
| #define LT_OBJDIR ".libs/"
| #define DBUS_UNIX 1
| #define DBUS_WITH_GLIB 1
| #define DBUS_DISABLE_ASSERT 1
| #define DBUS_DISABLE_CHECKS 1
| #define G_DISABLE_CHECKS 1
| #define DBUS_ENABLE_USERDB_CACHE 1
| #define SIZEOF_CHAR 1
| #define SIZEOF_SHORT 2
| #define SIZEOF_LONG 4
| #define SIZEOF_INT 4
| #define SIZEOF_VOID_P 4
| #define SIZEOF_LONG_LONG 8
| #define SIZEOF__INT64 0
| #define DBUS_INT64_PRINTF_MODIFIER "ll"
| #define DBUS_VA_COPY va_copy
| #define DBUS_USE_SYNC 1
| #define HAVE_VSNPRINTF 1
| #define HAVE_VASPRINTF 1
| #define HAVE_NANOSLEEP 1
| #define HAVE_USLEEP 1
| #define HAVE_SETENV 1
| #define HAVE_CLEARENV 1
| #define HAVE_UNSETENV 1
| #define HAVE_SOCKETPAIR 1
| #define HAVE_GETGROUPLIST 1
| #define HAVE_FPATHCONF 1
| #define HAVE_SETRLIMIT 1
| #define HAVE_POLL 1
| #define HAVE_SETLOCALE 1
| #define HAVE_LOCALECONV 1
| #define HAVE_STRTOLL 1
| #define HAVE_STRTOULL 1
| #define HAVE_GETRESUID 1
| #define HAVE_SYSLOG_H 1
| #define HAVE_DECL_LOG_PERROR 1
| #define HAVE_DIRFD 1
| #define HAVE_SYS_RESOURCE_H 1
| #define HAVE_DIRENT_H 1
| #define HAVE_EXECINFO_H 1
| #define HAVE_BACKTRACE 1
| #define HAVE_ERRNO_H 1
| #define HAVE_SIGNAL_H 1
| #define HAVE_LOCALE_H 1
| #define HAVE_BYTESWAP_H 1
| #define HAVE_UNISTD_H 1
| #define HAVE_POSIX_GETPWNAM_R 1
| #define HAVE_SOCKETLEN_T 1
| #define HAVE_SYS_UIO_H 1
| #define HAVE_WRITEV 1
| #define HAVE_DECL_MSG_NOSIGNAL 1
| #define HAVE_ISO_VARARGS 1
```

```

| #define HAVE_GNUC_VARARGS 1
| #define HAVE_PIPE2 1
| #define HAVE_ACCEPT4 1
| #define HAVE_ABSTRACT_SOCKETS 1
| #define HAVE_EXPAT_H 1
| #define HAVE_MONOTONIC_CLOCK 1
| #define HAVE_SYS_INOTIFY_H 1
| #define DBUS_BUS_ENABLE_INOTIFY 1
| #define HAVE_INOTIFY_INIT1 1
| #define DBUS_HAVE_LINUX_EPOLL 1
| #define HAVE_UNIX_FD_PASSING 1
| #define X_DISPLAY_MISSING 1
| #define DBUS_SYSTEM_SOCKET "/var/run/dbus/system_bus_socket"
| #define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"unix:path=/var/run/dbus/system_bus_socket"
| #define DBUS_CONSOLE_AUTH_DIR "/var/run/console/"
| #define DBUS_CONSOLE_OWNER_FILE ""
| #define DBUS_USER "messagebus"
| #define DBUS_PREFIX "/usr"
| #define DBUS_DATADIR "/usr/share"
| #define DBUS_DAEMONDIR "/usr/bin"
| #define DBUS_BINDIR "/usr/bin"
| #define DBUS_LIBEXECDIR "/usr/lib/dbus"
| #define DBUS_TEST_EXEC
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/test"
| #define DBUS_EXEEXT ""
| #define TEST_BUS_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon"
| #define DBUS_TEST_LAUNCH_HELPER_BINARY
"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-launch-helper-test"
| #define DBUS_TEST_SOCKET_DIR "/tmp"
| #define TEST_LISTEN "unix:tmpdir=/tmp"
| #define DBUS_SESSION_SOCKET_DIR "/tmp"
| /* end confdefs.h. */
| /* Define _NSGetEnviron to an innocuous variant, in case <limits.h>
declares _NSGetEnviron.
|     For example, HP-UX 11i <limits.h> declares gettimeofday. */
| #define _NSGetEnviron innocuous__NSGetEnviron
|
| /* System header to define __stub macros and hopefully few
prototypes,
|     which can conflict with char _NSGetEnviron (); below.
|     Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
|     <limits.h> exists even on freestanding compilers. */
|
| #ifdef __STDC__

```

```

| # include <limits.h>
| #else
| # include <assert.h>
| #endif
|
| #undef _NSGetEnviron
|
| /* Override any GCC internal prototype to avoid an error.
|    Use char because int might match the return type of a GCC
|    builtin and then its argument prototype would still apply.  */
| #ifdef __cplusplus
| extern "C"
| #endif
| char _NSGetEnviron ();
| /* The GNU C library defines this for functions which it implements
|    to always fail with ENOSYS.  Some functions are actually named
|    something starting with __ and the normal name is an alias.  */
| #if defined __stub__NSGetEnviron || defined __stub___NSGetEnviron
| choke me
| #endif
|
| int
| main ()
| {
| return _NSGetEnviron ();
| ;
| return 0;
| }

```

```

configure:22510: result: no
configure:22643: checking that generated files are newer than
configure
configure:22649: result: done
configure:22793: creating ./config.status

```

```

## ----- ##
## Running config.status. ##
## ----- ##

```

This file was extended by dbus config.status 1.6.8, which was generated by GNU Autoconf 2.69. Invocation command line was

```

CONFIG_FILES =
CONFIG_HEADERS =
CONFIG_LINKS =
CONFIG_COMMANDS =
$ ./config.status

```

on pc45

```

config.status:1486: creating Doxyfile
config.status:1486: creating dbus/versioninfo.rc
config.status:1486: creating dbus/dbus-arch-deps.h

```

```
config.status:1486: creating bus/system.conf
config.status:1486: creating bus/session.conf
config.status:1486: creating bus/messagebus
config.status:1486: creating bus/messagebus-config
config.status:1486: creating bus/org.freedesktop.dbus-session.plist
config.status:1486: creating bus/rc.messagebus
config.status:1486: creating bus/dbus.service
config.status:1486: creating bus/dbus.socket
config.status:1486: creating Makefile
config.status:1486: creating dbus/Makefile
config.status:1486: creating bus/Makefile
config.status:1486: creating tools/Makefile
config.status:1486: creating test/Makefile
config.status:1486: creating test/name-test/Makefile
config.status:1486: creating doc/Makefile
config.status:1486: creating doc/dbus-daemon.1
config.status:1486: creating dbus-1.pc
config.status:1486: creating dbus-1-uninstalled.pc
config.status:1486: creating test/data/valid-config-files/debug-allow-
all.conf
config.status:1486: creating test/data/valid-config-files/debug-allow-
all-sha1.conf
config.status:1486: creating test/data/valid-config-files-
system/debug-allow-all-pass.conf
config.status:1486: creating test/data/valid-config-files-
system/debug-allow-all-fail.conf
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
config.status:1486: creating test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
config.status:1486: creating test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service
config.status:1486: creating test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service
config.status:1486: creating test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
config.status:1486: creating test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
config.status:1486: creating test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service
config.status:1486: creating test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service
config.status:1486: creating test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service
```

```
config.status:1486: creating config.h
config.status:1715: executing depfiles commands
config.status:1715: executing libtool commands
configure:25554: WARNING: unrecognized options: --enable-nls
```

```
## ----- ##
## Cache variables. ##
## ----- ##
```

```
ac_cv___va_copy=yes
ac_cv_af_unix_large_dgram=yes
ac_cv_alignof_CORBA_boolean=1
ac_cv_alignof_CORBA_char=1
ac_cv_alignof_CORBA_double=8
ac_cv_alignof_CORBA_float=4
ac_cv_alignof_CORBA_long=4
ac_cv_alignof_CORBA_long_double=8
ac_cv_alignof_CORBA_long_long=8
ac_cv_alignof_CORBA_octet=1
ac_cv_alignof_CORBA_pointer=4
ac_cv_alignof_CORBA_short=2
ac_cv_alignof_CORBA_struct=1
ac_cv_alignof_CORBA_wchar=2
ac_cv_alignof_guint32=4
ac_cv_alignof_guint64=8
ac_cv_alignof_unsigned_long=4
ac_cv_buggy_getaddrinfo=no
ac_cv_build=i686-pc-linux-gnu
ac_cv_c_bigendian=no
ac_cv_c_compiler_gnu=yes
ac_cv_c_endian=little
ac_cv_c_inline=inline
ac_cv_c_littleendian=yes
ac_cv_c_long_double=yes
ac_cv_check_sjlj=ssjlj
ac_cv_conv_longlong_to_float=yes
ac_cv_cxx_compiler_gnu=yes
ac_cv_dirent_have_space_d_name=yes
ac_cv_env_CCC_set=
ac_cv_env_CCC_value=
ac_cv_env_CC_set=set
ac_cv_env_CC_value='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
ac_cv_env_CFLAGS_set=set
ac_cv_env_CFLAGS_value='-isystem/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -
fstack-protector -O2 -pipe -g -feliminate-unused-debug-types'
ac_cv_env_CPPFLAGS_set=set
```

```
ac_cv_env_CPPFLAGS_value='-isystem/home/gangadhar/gcc-linaro-arm-  
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include  
-fstack-protector'  
ac_cv_env_CPP_set=set  
ac_cv_env_CPP_value='arm-linux-gnueabi-gcc -E --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-  
interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8'  
ac_cv_env_CXXCPP_set=  
ac_cv_env_CXXCPP_value=  
ac_cv_env_CXXFLAGS_set=set  
ac_cv_env_CXXFLAGS_value='-isystem/home/gangadhar/gcc-linaro-arm-  
linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include  
-fstack-protector -O2 -pipe -g -feliminate-unused-debug-types -  
fpermissive'  
ac_cv_env_CXX_set=set  
ac_cv_env_CXX_value='arm-linux-gnueabi-g++ -march=armv7-a -marm -  
mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --  
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-  
linaro-toolchain/sysroots/delphi-vp1'  
ac_cv_env_DBUS_GLIB_CFLAGS_set=  
ac_cv_env_DBUS_GLIB_CFLAGS_value=  
ac_cv_env_DBUS_GLIB_LIBS_set=  
ac_cv_env_DBUS_GLIB_LIBS_value=  
ac_cv_env_GLIB_CFLAGS_set=  
ac_cv_env_GLIB_CFLAGS_value=  
ac_cv_env_GLIB_LIBS_set=  
ac_cv_env_GLIB_LIBS_value=  
ac_cv_env_LDFLAGS_set=set  
ac_cv_env_LDFLAGS_value='-L/home/gangadhar/gcc-linaro-arm-linux-  
gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/lib -Wl,-  
rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-  
20130313_linux/arm-linux-gnueabi/lib -Wl,-O1 -Wl,--hash-style=gnu'  
ac_cv_env_LIBS_set=  
ac_cv_env_LIBS_value=  
ac_cv_env_LIBXML_CFLAGS_set=  
ac_cv_env_LIBXML_CFLAGS_value=  
ac_cv_env_LIBXML_LIBS_set=  
ac_cv_env_LIBXML_LIBS_value=  
ac_cv_env_MAN2HTML_set=  
ac_cv_env_MAN2HTML_value=  
ac_cv_env_PKG_CONFIG_set=  
ac_cv_env_PKG_CONFIG_value=  
ac_cv_env_PYTHON_set=  
ac_cv_env_PYTHON_value=  
ac_cv_env_SYSTEMD_CFLAGS_set=  
ac_cv_env_SYSTEMD_CFLAGS_value=  
ac_cv_env_SYSTEMD_LIBS_set=  
ac_cv_env_SYSTEMD_LIBS_value=  
ac_cv_env_VALGRIND_CFLAGS_set=  
ac_cv_env_VALGRIND_CFLAGS_value=  
ac_cv_env_VALGRIND_LIBS_set=
```


ac_cv_env_VALGRIND_LIBS_value=
ac_cv_env_build_alias_set=set
ac_cv_env_build_alias_value=i686-linux
ac_cv_env_host_alias_set=set
ac_cv_env_host_alias_value=arm-oe-linux-gnueabi
ac_cv_env_target_alias_set=set
ac_cv_env_target_alias_value=arm-oe-linux-gnueabi
ac_cv_epoll_works=yes
ac_cv_file__dev_ptc_=yes
ac_cv_file__dev_random=yes
ac_cv_file__dev_zero=yes
ac_cv_func__NSGetEnviron=no
ac_cv_func__va_copy=yes
ac_cv_func_accept4=yes
ac_cv_func_backtrace=yes
ac_cv_func_clearenv=yes
ac_cv_func_fnmatch_works=yes
ac_cv_func_fpathconf=yes
ac_cv_func_getaddrinfo=yes
ac_cv_func_getgrgid_r=yes
ac_cv_func_getgrouplist=yes
ac_cv_func_gethostbyname=yes
ac_cv_func_getpagesize=yes
ac_cv_func_getpeereid=no
ac_cv_func_getpeerucred=no
ac_cv_func_getpgrp_void=yes
ac_cv_func_getpwuid_r=yes
ac_cv_func_getresuid=yes
ac_cv_func_inotify_init1=yes
ac_cv_func_issetugid=no
ac_cv_func_kqueue=no
ac_cv_func_localeconv=yes
ac_cv_func_lstat_dereferences_slashed_symlink=yes
ac_cv_func_lstat_empty_string_bug=no
ac_cv_func_malloc_0_nonnull=yes
ac_cv_func_malloc_works=yes
ac_cv_func_memcmp_working=yes
ac_cv_func_memcpy=yes
ac_cv_func_mmap_fixed_mapped=yes
ac_cv_func_nanosleep=yes
ac_cv_func_pipe2=yes
ac_cv_func_poll=yes
ac_cv_func_posix_getgrgid_r=yes
ac_cv_func_posix_getpwnam_r=yes
ac_cv_func_posix_getpwuid_r=yes
ac_cv_func_printf_unix98=yes
ac_cv_func_pthread_attr_getstack=yes
ac_cv_func_pthread_cond_timedwait=yes
ac_cv_func_pthread_condattr_setclock=yes
ac_cv_func_pthread_key_delete=yes
ac_cv_func_realloc_0_nonnull=yes
ac_cv_func_realloc_works=yes

ac_cv_func_setenv=yes
ac_cv_func_setgrent_void=yes
ac_cv_func_setlocale=yes
ac_cv_func_setpgrp_void=yes
ac_cv_func_setresuid=yes
ac_cv_func_setrlimit=yes
ac_cv_func_setvbuf_reversed=no
ac_cv_func_snprintf_c99=yes
ac_cv_func_socketpair=yes
ac_cv_func_stat_empty_string_bug=no
ac_cv_func_stat_ignores_trailing_slash=no
ac_cv_func_strerror=yes
ac_cv_func_strtoll=yes
ac_cv_func_strtoull=yes
ac_cv_func_unsetenv=yes
ac_cv_func_usleep=yes
ac_cv_func_va_copy=yes
ac_cv_func_vasprintf=yes
ac_cv_func_vsnprintf=yes
ac_cv_func_vsnprintf_c99=yes
ac_cv_func_writev=yes
ac_cv_gnet_have_abstract_sockets=no
ac_cv_gnu_library_2_1=yes
ac_cv_have_abstract_sockets=yes
ac_cv_have_accrights_in_msghdr=no
ac_cv_have_broken_snprintf=no
ac_cv_have_control_in_msghdr=yes
ac_cv_have_decl_LOG_PERROR=yes
ac_cv_have_decl_MSG_NOSIGNAL=yes
ac_cv_have_decl_sys_siglist=yes
ac_cv_have_long_long_format=yes
ac_cv_have_openpty_ctty_bug=yes
ac_cv_have_space_d_name_in_struct_dirent=yes
ac_cv_header_X=no
ac_cv_header__CC=no
ac_cv_header__CFLAGS=no
ac_cv_header__CPPFLAGS=no
ac_cv_header__5=no
ac_cv_header__c=no
ac_cv_header_byteswap_h=yes
ac_cv_header_conftest__ac_ext=no
ac_cv_header_crt_externs_h=no
ac_cv_header_curses_h=yes
ac_cv_header_dirent_h=yes
ac_cv_header_dlfcn_h=yes
ac_cv_header_errno_h=yes
ac_cv_header_execinfo_h=yes
ac_cv_header_expats_h=yes
ac_cv_header_inttypes_h=yes
ac_cv_header_launch_h=no
ac_cv_header_locale_h=yes
ac_cv_header_memory_h=yes

```
ac_cv_header_minix_config_h=no
ac_cv_header_ncurses_curses_h=no
ac_cv_header_ncurses_h=yes
ac_cv_header_ncurses_termcap_h=no
ac_cv_header_netinet_sctp_h=no
ac_cv_header_netinet_sctp_uio_h=no
ac_cv_header_signal_h=yes
ac_cv_header_stdcl=yes
ac_cv_header_stdint_h=yes
ac_cv_header_stdlib_h=yes
ac_cv_header_string_h=yes
ac_cv_header_strings_h=yes
ac_cv_header_sys_event_h=no
ac_cv_header_sys_inotify_h=yes
ac_cv_header_sys_resource_h=yes
ac_cv_header_sys_stat_h=yes
ac_cv_header_sys_syslimits_h=no
ac_cv_header_sys_types_h=yes
ac_cv_header_sys_uio_h=yes
ac_cv_header_syslog_h=yes
ac_cv_header_unistd_h=yes
ac_cv_header_ws2tcpip_h=no
ac_cv_header_wspiapi_h=no
ac_cv_host=arm-oe-linux-gnueabi
ac_cv_ispeed_offset=13
ac_cv_lib_expat_XML_ParserCreate_MM=yes
ac_cv_lib_png_png_create_info_struct=yes
ac_cv_lib_pthread_pthread_cond_timedwait=yes
ac_cv_lib_selinux_is_selinux_enabled=no
ac_cv_libiconv_utf8=yes
ac_cv_libnet_endianness=lil
ac_cv_linux_vers=2
ac_cv_member_struct_sockaddr_sa_len=no
ac_cv_need_trio=no
ac_cv_o_nonblock_inherited=no
ac_cv_objext=o
ac_cv_path_DOXYGEN=/usr/bin/doxygen
ac_cv_path_EGREP='/bin/grep -E'
ac_cv_path_FGREP='/bin/grep -F'
ac_cv_path_GREP=/bin/grep
ac_cv_path_PKG_CONFIG=/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config
ac_cv_path_PYTHON=/usr/bin/python
ac_cv_path_SED=/bin/sed
ac_cv_path_XMLTO=no
ac_cv_path_ac_pt_PKG_CONFIG=/home/gangadhar/newyoctobuild/tisdk/build/
arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-
config
ac_cv_path_install='/usr/bin/install -c'
ac_cv_path_mkdir=/bin/mkdir
ac_cv_pread=no
ac_cv_prog_AR=arm-linux-gnueabi-hf-ar
```

```
ac_cv_prog_AWK=gawk
ac_cv_prog_CC='arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
ac_cv_prog_CPP='arm-linux-gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
ac_cv_prog_CXXCPP='arm-linux-gnueabi-g++ -march=armv7-a -marm -
mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -E'
ac_cv_prog_OBJDUMP=arm-linux-gnueabi-objdump
ac_cv_prog_RANLIB=arm-linux-gnueabi-ranlib
ac_cv_prog_STRIP=arm-linux-gnueabi-strip
ac_cv_prog_XSLTPROC=xsltproc
ac_cv_prog_ac_ct_MANIFEST_TOOL=mt
ac_cv_prog_cc_arm_linux_gnueabi_gcc_c_o=yes
ac_cv_prog_cc_c89=
ac_cv_prog_cc_g=yes
ac_cv_prog_cxx_g=yes
ac_cv_prog_make_make_set=yes
ac_cv_pwrite=no
ac_cv_regex_exec_segfault_emptystr=no
ac_cv_safe_to_define___extensions___=yes
ac_cv_sctp=no
ac_cv_search_clock_getres=-lrt
ac_cv_search_socket='none required'
ac_cv_search_strerror='none required'
ac_cv_sizeof___int64=0
ac_cv_sizeof_bool=1
ac_cv_sizeof_char=1
ac_cv_sizeof_char_p=4
ac_cv_sizeof_double=8
ac_cv_sizeof_float=4
ac_cv_sizeof_int=4
ac_cv_sizeof_int_p=4
ac_cv_sizeof_long=4
ac_cv_sizeof_long_double=8
ac_cv_sizeof_long_int=4
ac_cv_sizeof_long_long=8
ac_cv_sizeof_long_p=4
ac_cv_sizeof_off_t=4
ac_cv_sizeof_pid_t=4
ac_cv_sizeof_ptrdiff_t=4
ac_cv_sizeof_short=2
ac_cv_sizeof_short_int=2
ac_cv_sizeof_signed_char=1
ac_cv_sizeof_size_t=4
ac_cv_sizeof_ssize_t=4
ac_cv_sizeof_unsigned=4
```

```
ac_cv_sizeof_unsigned_char=1
ac_cv_sizeof_unsigned_int=4
ac_cv_sizeof_unsigned_long=4
ac_cv_sizeof_unsigned_long_long=8
ac_cv_sizeof_unsigned_short=2
ac_cv_sizeof_void_p=4
ac_cv_sizeof_wchar_t=4
ac_cv_strerror_r_SUSv3=no
ac_cv_sys_restartable_syscalls=yes
ac_cv_time_r_type=POSIX
ac_cv_type_uid_t='{ac_cv_type_uid_t=yes}'
ac_cv_uchar=no
ac_cv_uint=yes
ac_cv_ulong=yes
ac_cv_ushort=yes
ac_cv_va_copy=yes
ac_cv_va_val_copy=yes
am_cv_CC_dependencies_compiler_type=none
am_cv_CXX_dependencies_compiler_type=none
am_cv_func_working_getline=yes
am_cv_make_support_nested_variables=yes
am_cv_pathless_PYTHON=python
am_cv_prog_tar_ustar=gnutar
am_cv_python_platform=linux2
am_cv_python_pyexecdir='${exec_prefix}/lib/python2.7/dist-packages'
am_cv_python_pythondir='${prefix}/lib/python2.7/dist-packages'
am_cv_python_version=2.7
am_cv_scanf_llt=yes
apr_cv_process_shared_works=no
apr_cv_tcp_nodelay_with_cork=yes
as_cv_unaligned_access=no
bash_cv_dup2_broken=no
bash_cv_func_sigsetjmp=missing
bash_cv_func_strcoll_broken=no
bash_cv_getcwd_calls_popen=no
bash_cv_getenv_redef=yes
bash_cv_have_mbstate_t=yes
bash_cv_job_control_missing=present
bash_cv_must_reinstall_sighandlers=no
bash_cv_opendir_not_robust=no
bash_cv_pgrp_pipe=no
bash_cv_printf_a_format=yes
bash_cv_sys_named_pipes=present
bash_cv_sys_siglist=yes
bash_cv_type_rlimit=rlim_t
bash_cv_ulimit_maxfds=yes
bash_cv_under_sys_siglist=yes
bash_cv_unusable_rtsigs=no
compat_cv_func_basename_works=no
compat_cv_func_dirname_works=no
compat_cv_func_snprintf_works=yes
cvs_cv_func_printf_ptr=yes
```

```
db_cv_align_t='unsigned long long'  
db_cv_alignp_t='unsigned long'  
db_cv_fcntl_f_setfd=yes  
db_cv_path_ar=/usr/bin/ar  
db_cv_path_chmod=/bin/chmod  
db_cv_path_cp=/bin/cp  
db_cv_path_ln=/bin/ln  
db_cv_path_mkdir=/bin/mkdir  
db_cv_path_ranlib=/usr/bin/ranlib  
db_cv_path_rm=/bin/rm  
db_cv_path_sh=/bin/sh  
db_cv_path_strip=/usr/bin/strip  
db_cv_spinlocks=no  
db_cv_sprintf_count=yes  
dbus_cv__va_copy=yes  
dbus_cv_sync_sub_and_fetch=yes  
dbus_cv_va_copy=yes  
dbus_cv_va_val_copy=yes  
ettercap_cv_type_socklen_t=yes  
fu_cv_sys_stat_statfs2_bsize=yes  
gl_cv_func_btowc_eof=yes  
gl_cv_func_fstatat_zero_flag=yes  
gl_cv_func_mbrtowc_incomplete_state=yes  
gl_cv_func_mbrtowc_nul_retval=yes  
gl_cv_func_mbrtowc_null_arg=yes  
gl_cv_func_mbrtowc_retval=yes  
gl_cv_func_mbrtowc_sanitycheck=yes  
gl_cv_func_wcrtomb_retval=yes  
gl_cv_func_wctob_works=yes  
glib_cv__va_copy=yes  
glib_cv_compliant_posix_memalign=1  
glib_cv_has__inline=yes  
glib_cv_has__inline__=yes  
glib_cv_hasinline=yes  
glib_cv_have_qsort_r=yes  
glib_cv_long_long_format=ll  
glib_cv_rtdglobal_broken=no  
glib_cv_sane_realloc=yes  
glib_cv_sizeof_gmutex=24  
glib_cv_sizeof_intmax_t=8  
glib_cv_sizeof_ptrdiff_t=4  
glib_cv_sizeof_size_t=4  
glib_cv_sizeof_system_thread=4  
glib_cv_stack_grows=no  
glib_cv_strncpy=no  
glib_cv_sys_pthread_cond_timedwait_posix=yes  
glib_cv_sys_pthread_getspecific_posix=yes  
glib_cv_sys_pthread_mutex_trylock_posix=yes  
glib_cv_sys_use_pid_niceness_surrogate=yes  
glib_cv_uscore=no  
glib_cv_use_pid_surrogate=yes  
glib_cv_va_copy=yes
```

```

glib_cv_va_val_copy=yes
gt_cv_locale_fr=fr_FR
jm_cv_func_working_readdir=yes
lf_cv_sane_realloc=yes
lftp_cv_va_val_copy=yes
libIDL_cv_long_long_format=ll
lt_cv_ar_at_file=@
lt_cv_archive_cmds_need_lc=no
lt_cv_deplibs_check_method=pass_all
lt_cv_file_magic_cmd='$MAGIC_CMD'
lt_cv_file_magic_test_file=
lt_cv_ld_reload_flag=-r
lt_cv_nm_interface='BSD nm'
lt_cv_objdir=.libs
lt_cv_path_LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
lt_cv_path_LDCXX='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
lt_cv_path_NM=arm-linux-gnueabi-hf-nm
lt_cv_path_mainfest_tool=no
lt_cv_prog_compiler_c_o=yes
lt_cv_prog_compiler_c_o_CXX=yes
lt_cv_prog_compiler_c_o_RC=yes
lt_cv_prog_compiler_pic='-fPIC -DPIC'
lt_cv_prog_compiler_pic_CXX='-fPIC -DPIC'
lt_cv_prog_compiler_pic_works=yes
lt_cv_prog_compiler_pic_works_CXX=yes
lt_cv_prog_compiler_rtti_exceptions=no
lt_cv_prog_compiler_static_works=yes
lt_cv_prog_compiler_static_works_CXX=yes
lt_cv_prog_gnu_ld=yes
lt_cv_prog_gnu_ldcxx=yes
lt_cv_sharedlib_from_linklib_cmd='printf %s\n'
lt_cv_shlibpath_overrides_runpath=no
lt_cv_sys_global_symbol_pipe='sed -n -e '\''s/^.*[
]\([ABCDGIRSTW][ABCDGIRSTW]*\) [ ]* \([ ]*\)\([_A-Za-z0-
9]*\)$/\1 \2 \2/p'\'' | sed '\''/ __gnu_lto/d'\''
lt_cv_sys_global_symbol_to_c_name_address='sed -n -e '\''s/^: \([
]*\) [ ]*$/ { "\1", (void *) 0 },/p'\'' -e '\''s/^[ABCDGIRSTW]* \([
]*\) \([ ]*\)\$/ {"\2", (void *) \&\2},/p'\''
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix='sed -n -e
'\''s/^: \([ ]*\) [ ]*$/ {"\1", (void *) 0},/p'\'' -e
'\''s/^[ABCDGIRSTW]* \([ ]*\) \([ ]*\)\$/ {"\2", (void *)
\&\2},/p'\'' -e '\''s/^[ABCDGIRSTW]* \([ ]*\) \([ ]*\)\$/ {"lib\2",
(void *) \&\2},/p'\''
lt_cv_sys_global_symbol_to_cdecl='sed -n -e '\''s/^T .* \(.*)$/extern
int \1();/p'\'' -e '\''s/^[ABCDGIRSTW]* .* \(.*)$/extern char
\1;/p'\''
lt_cv_sys_lib_dlsearch_path_spec='/usr/lib /lib'
lt_cv_sys_max_cmd_len=1572864

```

lt_cv_to_host_file_cmd=func_convert_file_noop
lt_cv_to_tool_file_cmd=func_convert_file_noop
mr_cv_target_elf=yes
mysql_cv_func_atomic_add=no
mysql_cv_func_atomic_sub=no
nano_cv_func_regexec_segv_emptystr=no
php_cv_lib_cookie_io_functions_use_off64_t=yes
rppoe_cv_pack_bitfields=rev
rsync_cv_HAVE_BROKEN_LARGEFILE=no
rsync_cv_HAVE_BROKEN_READDIR=no
rsync_cv_HAVE_C99_VSNPRINTF=yes
rsync_cv_HAVE_GETTIMEOFDAY_TZ=yes
rsync_cv_HAVE_LONGLONG=yes
rsync_cv_HAVE_OFF64_T=no
rsync_cv_HAVE_SECURE_MKSTEMP=yes
rsync_cv_HAVE_SHORT_INO_T=no
rsync_cv_HAVE_SOCKETPAIR=yes
rsync_cv_HAVE_UNSIGNED_CHAR=no
rsync_cv_REPLACE_INET_ATON=no
rsync_cv_REPLACE_INET_NTOA=no
samba_cv_BROKEN_NISPLUS_INCLUDE_FILES=yes
samba_cv_BROKEN_REDHAT_7_SYSTEM_HEADERS=no
samba_cv_HAVE_BROKEN_FCNTL64_LOCKS=no
samba_cv_HAVE_BROKEN_GETGROUPS=no
samba_cv_HAVE_BROKEN_LINUX_SENDFILE=yes
samba_cv_HAVE_BROKEN_READDIR=yes
samba_cv_HAVE_BROKEN_READDIR_NAME=no
samba_cv_HAVE_C99_VSNPRINTF=yes
samba_cv_HAVE_DEV64_T=no
samba_cv_HAVE_DEVICE_MAJOR_FN=yes
samba_cv_HAVE_DEVICE_MINOR_FN=yes
samba_cv_HAVE_DQB_FSOFTLIMIT=no
samba_cv_HAVE_EXPLICIT_LARGEFILE_SUPPORT=yes
samba_cv_HAVE_FAM_H=no
samba_cv_HAVE_FCNTL_LOCK=yes
samba_cv_HAVE_FTRUNCATE_EXTEND=yes
samba_cv_HAVE_FUNCTION_MACRO=yes
samba_cv_HAVE_GETTIMEOFDAY_TZ=yes
samba_cv_HAVE_IFACE_AIX=no
samba_cv_HAVE_IFACE_IFCONF=yes
samba_cv_HAVE_IFACE_IFREQ=yes
samba_cv_HAVE_INO64_T=no
samba_cv_HAVE_INT16_FROM_RPC_RPC_H=no
samba_cv_HAVE_INT32_FROM_RPC_RPC_H=no
samba_cv_HAVE_IRIX_SPECIFIC_CAPABILITIES=no
samba_cv_HAVE_KERNEL_CHANGE_NOTIFY=yes
samba_cv_HAVE_KERNEL_OPLOCKS_LINUX=yes
samba_cv_HAVE_KERNEL_SHARE_MODES=yes
samba_cv_HAVE_MAKEDEV=yes
samba_cv_HAVE_MMAP=yes
samba_cv_HAVE_NATIVE_ICONV=yes
samba_cv_HAVE_OFF64_T=no

samba_cv_HAVE_ROOT=yes
samba_cv_HAVE_RPC_AUTH_ERROR_CONFLICT=no
samba_cv_HAVE_SECURE_MKSTEMP=yes
samba_cv_HAVE_SENDFILE64=yes
samba_cv_HAVE_SENDFILE=yes
samba_cv_HAVE_SOCKET_SIN_LEN=no
samba_cv_HAVE_STAT_ST_BLKSIZE=yes
samba_cv_HAVE_STAT_ST_BLOCKS=yes
samba_cv_HAVE_STRUCT_DIR64=no
samba_cv_HAVE_STRUCT_DIRENT64=yes
samba_cv_HAVE_STRUCT_FLOCK64=yes
samba_cv_HAVE_TRUNCATED_SALT=no
samba_cv_HAVE_UINT16_FROM_RPC_RPC_H=no
samba_cv_HAVE_UINT32_FROM_RPC_RPC_H=no
samba_cv_HAVE_UNSIGNED_CHAR=yes
samba_cv_HAVE_UTIMBUF=yes
samba_cv_HAVE_UT_UT_ADDR=yes
samba_cv_HAVE_UT_UT_EXIT=yes
samba_cv_HAVE_UT_UT_HOST=yes
samba_cv_HAVE_UT_UT_ID=yes
samba_cv_HAVE_UT_UT_NAME=yes
samba_cv_HAVE_UT_UT_PID=yes
samba_cv_HAVE_UT_UT_TIME=yes
samba_cv_HAVE_UT_UT_TV=yes
samba_cv_HAVE_UT_UT_TYPE=yes
samba_cv_HAVE_UT_UT_USER=yes
samba_cv_HAVE_UX_UT_SYSLLEN=no
samba_cv_HAVE_VA_COPY=yes
samba_cv_HAVE_WORKING_AF_LOCAL=yes
samba_cv_HAVE_Werror=yes
samba_cv_LINUX_LFS_SUPPORT=yes
samba_cv_PUTUTLINE_RETURNS_UTMP=yes
samba_cv_QUOTA_WORKS=yes
samba_cv_REALPATH_TAKES_NULL=yes
samba_cv_REPLACE_GETPASS=yes
samba_cv_REPLACE_INET_NTOA=no
samba_cv_REPLACE_READDIR=no
samba_cv_RUN_QUOTA_TESTS=yes
samba_cv_SEEKDIR_RETURNS_VOID=yes
samba_cv_SIZEOF_DEV_T=yes
samba_cv_SIZEOF_INO_T=yes
samba_cv_SIZEOF_OFF_T=yes
samba_cv_SYSCONF_SC_NGROUPS_MAX=yes
samba_cv_SYSCONF_SC_NPROCESSORS_ONLN=yes
samba_cv_SYSCONF_SC_NPROC_ONLN=no
samba_cv_SYSCONF_SC_PAGESIZE=yes
samba_cv_SYSQUOTA_FOUND=yes
samba_cv_SYSQUOTA_WORKS=yes
samba_cv_SYSQUOTA_WORKS_XFS=yes
samba_cv_TRY_QUOTAS=no
samba_cv_TRY_SYS_QUOTAS=yes
samba_cv_USE_SETEUID=yes

```
samba_cv_USE_SETRESUID=yes
samba_cv_USE_SETREUID=yes
samba_cv_USE_SETUIDX=yes
samba_cv_WE_USE_SYS_QUOTAS=yes
samba_cv_WITH_AFS=no
samba_cv_WITH_FAKE_KASERVER=no
samba_cv_WITH_QUOTAS=auto
samba_cv_WITH_SYS_QUOTAS=auto
samba_cv_WITH_VFS_AFSACL=no
samba_cv_compiler_supports_ll=yes
samba_cv_found_xfs_header=yes
samba_cv_have_longlong=yes
samba_cv_have_setresgid=yes
samba_cv_have_setresuid=yes
samba_cv_immediate_structures=yes
samba_cv_optimize_out_funcation_calls=yes
samba_cv_sig_atomic_t=yes
samba_cv_socklen_t=yes
samba_cv_struct_timespec=yes
samba_cv_sysquotas_file=lib/sysquotas_linux.c
samba_cv_unixsocket=yes
samba_cv_volatile=yes
sc_cv_termios_ispeed=yes
screen_cv_sys_bcopy_overlap=no
screen_cv_sys_fifo_broken_impl=yes
screen_cv_sys_fifo_usable=yes
screen_cv_sys_memcpy_overlap=no
screen_cv_sys_memmove_overlap=no
screen_cv_sys_select_broken_retval=no
screen_cv_sys_sockets_nofs=no
screen_cv_sys_sockets_usable=yes
screen_cv_sys_terminfo_used=yes
shadow_cv_logdir=/var/log
shadow_cv_maildir=/var/spool/mail
shadow_cv_mailfile=Mailbox
shadow_cv_passwd_dir=/usr/bin
shadow_cv_utmpdir=/var/run
slrn_cv_va_val_copy=yes
sudo_cv_func_unsetenv_void=no
tcl_cv_api_serial=termios
utils_cv_sys_open_max=1019
```

```
## ----- ##
## Output variables. ##
## ----- ##
```

```
ACLOCAL='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/missing --run aclocal-1.12'
ADT_LIBS=''
AMDEPBACKSLASH=''
AMDEP_FALSE=''
```

```
AMDEP_TRUE='#'
AMTAR='${TAR-tar}'
AM_BACKSLASH='\ '
AM_DEFAULT_V='$ (AM_DEFAULT_VERBOSITY) '
AM_DEFAULT_VERBOSITY='1'
AM_V='$ (V) '
AR='arm-linux-gnueabi-hf-ar'
AUTOCONF='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/missing --run autoconf'
AUTOHEADER='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/missing --run autoheader'
AUTOMAKE='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/missing --run automake-1.12'
AWK='gawk'
BUILD_FILEVERSION=''
BUILD_TIMESTAMP=''
CC='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
CCDEPMODE='depmode=none'
CFLAGS=' -Wall -Wextra -Wchar-subscripts -Wmissing-declarations -
Wmissing-prototypes -Wnested-externs -Wpointer-arith -Wcast-align -
Wno-address -Wfloat-equal -Wdeclaration-after-statement -Wno-unused-
label -Wno-missing-field-initializers -Wno-unused-parameter -Wno-sign-
compare -Wno-pointer-sign -Wno-type-limits -fno-common -fno-strict-
aliasing -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types'
CPP='arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
CPPFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector'
CXX='arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
CXXCPP='arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -E'
CXXDEPMODE='depmode=none'
CXXFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types -fpermissive'
CYGPATH_W='echo'
DBUS_BINDIR='/usr/bin'
```

```
DBUS_BUILD_TESTS_FALSE=''
DBUS_BUILD_TESTS_TRUE='#'
DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE=''
DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE='#'
DBUS_BUS_ENABLE_INOTIFY_FALSE='#'
DBUS_BUS_ENABLE_INOTIFY_TRUE=''
DBUS_BUS_ENABLE_KQUEUE_FALSE=''
DBUS_BUS_ENABLE_KQUEUE_TRUE='#'
DBUS_CAN_UPLOAD_DOCS_FALSE=''
DBUS_CAN_UPLOAD_DOCS_TRUE='#'
DBUS_CONSOLE_AUTH_DIR='/var/run/console/'
DBUS_CONSOLE_OWNER_FILE=''
DBUS_CYGWIN_FALSE=''
DBUS_CYGWIN_TRUE='#'
DBUS_DAEMONDIR='/usr/bin'
DBUS_DATADIR='/usr/share'
DBUS_DOXYGEN_DOCS_ENABLED_FALSE=''
DBUS_DOXYGEN_DOCS_ENABLED_TRUE='#'
DBUS_ENABLE_EMBEDDED_TESTS_FALSE=''
DBUS_ENABLE_EMBEDDED_TESTS_TRUE='#'
DBUS_ENABLE_INSTALLED_TESTS_FALSE=''
DBUS_ENABLE_INSTALLED_TESTS_TRUE='#'
DBUS_ENABLE_LAUNCHD_FALSE=''
DBUS_ENABLE_LAUNCHD_TRUE='#'
DBUS_ENABLE_MODULAR_TESTS_FALSE=''
DBUS_ENABLE_MODULAR_TESTS_TRUE='#'
DBUS_GLIB_CFLAGS=''
DBUS_GLIB_LIBS=''
DBUS_HAVE_INT64='1'
DBUS_HAVE_MAN2HTML_FALSE=''
DBUS_HAVE_MAN2HTML_TRUE='#'
DBUS_HAVE_XSLTPROC_FALSE='#'
DBUS_HAVE_XSLTPROC_TRUE=''
DBUS_INIT_SCRIPTS_CYGWIN_FALSE=''
DBUS_INIT_SCRIPTS_CYGWIN_TRUE='#'
DBUS_INIT_SCRIPTS_RED_HAT_FALSE=''
DBUS_INIT_SCRIPTS_RED_HAT_TRUE='#'
DBUS_INIT_SCRIPTS_SLACKWARE_FALSE=''
DBUS_INIT_SCRIPTS_SLACKWARE_TRUE='#'
DBUS_INT16_TYPE='short'
DBUS_INT32_TYPE='int'
DBUS_INT64_CONSTANT='(val##LL) '
DBUS_INT64_TYPE='long long'
DBUS_LIBEXECDIR='/usr/lib/dbus'
DBUS_MAJOR_VERSION='1'
DBUS_MICRO_VERSION='8'
DBUS_MINOR_VERSION='6'
DBUS_PATH_OR_ABSTRACT='abstract'
DBUS_PREFIX='/usr'
DBUS_SESSION_BUS_DEFAULT_ADDRESS='unix:tmpdir=/tmp'
DBUS_SESSION_SOCKET_DIR='/tmp'
DBUS_STATIC_BUILD_CPPFLAGS=''
```

```
DBUS_SYSTEM_BUS_DEFAULT_ADDRESS='unix:path=/var/run/dbus/system_bus_socket'
DBUS_SYSTEM_PID_FILE='/var/run/dbus/pid'
DBUS_SYSTEM_SOCKET='/var/run/dbus/system_bus_socket'
DBUS_TEST_DATA='/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/test/data'
DBUS_TEST_EXEC='/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/test'
DBUS_UINT64_CONSTANT='(val##ULL)'
DBUS_UNIX_FALSE='#'
DBUS_UNIX_TRUE=''
DBUS_USER='messagebus'
DBUS_USE_EXPAT_FALSE='#'
DBUS_USE_EXPAT_TRUE=''
DBUS_USE_LIBXML_FALSE=''
DBUS_USE_LIBXML_TRUE='#'
DBUS_VERSION='1.6.8'
DBUS_WINCE_FALSE=''
DBUS_WINCE_TRUE='#'
DBUS_WIN_FALSE=''
DBUS_WIN_TRUE='#'
DBUS_WITH_GLIB_FALSE='#'
DBUS_WITH_GLIB_TRUE=''
DBUS_XML_DOCS_ENABLED_FALSE=''
DBUS_XML_DOCS_ENABLED_TRUE='#'
DBUS_X_CFLAGS=''
DBUS_X_LIBS=''
DEFS='-DHAVE_CONFIG_H'
DEPDIR='.deps'
DLLTOOL='false'
DOXYGEN='/usr/bin/doxygen'
DSYMUTIL=''
DUMPBIN=''
ECHO_C=''
ECHO_N='-n'
ECHO_T=''
EGREP='/bin/grep -E'
EXEEXT=''
EXPANDED_BINDIR='/usr/bin'
EXPANDED_DATADIR='/usr/share'
EXPANDED_LIBDIR='/usr/lib'
EXPANDED_LIBEXECDIR='/usr/lib/dbus'
EXPANDED_LOCALSTATEDIR='/var'
EXPANDED_PREFIX='/usr'
EXPANDED_SYSCONFDIR='/etc'
FGREP='/bin/grep -F'
GETTEXT_PACKAGE='dbus-1'
GLIB_CFLAGS=''
GLIB_LIBS=''
GREP='/bin/grep'
```

```
HAVE_CONSOLE_OWNER_FILE_FALSE=''
HAVE_CONSOLE_OWNER_FILE_TRUE='#'
HAVE_LIBAUDIT_FALSE=''
HAVE_LIBAUDIT_TRUE='#'
HAVE_LINUX_EPOLL_FALSE='#'
HAVE_LINUX_EPOLL_TRUE=''
HAVE_SELINUX_FALSE=''
HAVE_SELINUX_TRUE='#'
HAVE_SYSTEMD_FALSE=''
HAVE_SYSTEMD_TRUE='#'
INSTALL_DATA='${INSTALL} -m 644'
INSTALL_PROGRAM='${INSTALL}'
INSTALL_SCRIPT='${INSTALL}'
INSTALL_STRIP_PROGRAM='${install_sh} -c -s'
LAUNCHCTL=''
LAUNCHD_AGENT_DIR='/Library/LaunchAgents'
LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
LDFLAGS='-L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-style=gnu'
LIBDBUS_LIBS='-lpthread -lrt '
LIBBOJS=''
LIBS=''
LIBTOOL='${top_builddir}/arm-oe-linux-gnueabi-libtool'
LIBXML_CFLAGS=''
LIBXML_LIBS=''
LIPO=''
LN_S='ln -s'
LTLIBBOJS=''
LT_AGE='7'
LT_CURRENT='10'
LT_REVISION='2'
MAINT=''
MAINTAINER_MODE_FALSE='#'
MAINTAINER_MODE_TRUE=''
MAKEINFO='${SHELL} /home/gangadhar/newyoctobuild/tisdsk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/missing --run makeinfo'
MAN2HTML=''
MANIFEST_TOOL=':'
MKDIR_P='/bin/mkdir -p'
NETWORK_libs=''
NM='arm-linux-gnueabi-hf-nm'
NMEDIT=''
OBJDUMP='arm-linux-gnueabi-hf-objdump'
OBJEXT='o'
OTOOL64=''
OTOOL=''
PACKAGE='dbus'
```

```
PACKAGE_BUGREPORT='https://bugs.freedesktop.org/enter_bug.cgi?product=
dbus'
PACKAGE_NAME='dbus'
PACKAGE_STRING='dbus 1.6.8'
PACKAGE_TARNAME='dbus'
PACKAGE_URL=''
PACKAGE_VERSION='1.6.8'
PATH_SEPARATOR=':'
PKG_CONFIG='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config'
PYTHON='/usr/bin/python'
PYTHON_EXEC_PREFIX='${exec_prefix}'
PYTHON_LIB_PREFIX='${libdir}'
PYTHON_PLATFORM='linux2'
PYTHON_PREFIX='${prefix}'
PYTHON_VERSION='2.7'
RANLIB='arm-linux-gnueabi-hf-ranlib'
RC=''
R_DYNAMIC_LDFLAG=''
SED='/bin/sed'
SELINUX_LIBS=''
SET_MAKE=''
SHELL='/bin/sh'
STRIP='arm-linux-gnueabi-hf-strip'
SYSTEMD_CFLAGS=''
SYSTEMD_LIBS=''
TEST_BUS_BINARY=''
TEST_LAUNCH_HELPER_BINARY='/home/gangadhar/newyoctobuild/tisdk/build/a
rago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-
linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-
launch-helper-test'
TEST_LISTEN='unix:tmpdir=/tmp'
TEST_SOCKET_DIR='/tmp'
THREAD_LIBS='-lpthread -lrt'
VALGRIND_CFLAGS=''
VALGRIND_LIBS=''
VERSION='1.6.8'
WINDRES=''
XMLTO='no'
XML_CFLAGS=''
XML_LIBS='-lexpat'
XSLTPROC='xsltproc'
X_CFLAGS=''
X_EXTRA_LIBS=''
X_LIBS=''
X_PRE_LIBS=''
ac_ct_AR=''
ac_ct_CC=''
ac_ct_CXX=''
ac_ct_DUMPBIN=''
am_EXEEXT_FALSE=''
am_EXEEXT_TRUE='#'
```

```
am__fastdepCC_FALSE=''
am__fastdepCC_TRUE='#'
am__fastdepCXX_FALSE=''
am__fastdepCXX_TRUE='#'
am__include='include'
am__isrc=''
am__leading_dot='.'
am__nodep=''
am__quote=''
am__tar='tar --format=ustar -chf - "$$stardir"'
am__untar='tar -xf -'
bindir='/usr/bin'
build='i686-pc-linux-gnu'
build_alias='i686-linux'
build_cpu='i686'
build_os='linux-gnu'
build_vendor='pc'
datadir='/usr/share'
datarootdir='${prefix}/share'
docdir='${datarootdir}/doc/${PACKAGE_TARNAME}'
dvidir='${docdir}'
exec_prefix='/usr'
host='arm-oe-linux-gnueabi'
host_alias='arm-oe-linux-gnueabi'
host_cpu='arm'
host_os='linux-gnueabi'
host_vendor='oe'
htmldir='${docdir}'
includedir='/usr/include'
infodir='/usr/share/info'
install_sh='${SHELL} /home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/install-sh'
libdir='/usr/lib'
libexecdir='/usr/lib/dbus'
localedir='${datarootdir}/locale'
localstatedir='/var'
mandir='/usr/share/man'
mkdir_p='${MKDIR_P}'
oldincludedir='/usr/include'
pdfdir='${docdir}'
pkgpyexecdir='${pyexecdir}/dbus'
pkgpythondir='${pythondir}/dbus'
prefix='/usr'
program_transform_name='s,x,x,'
psdir='${docdir}'
pyexecdir='${exec_prefix}/lib/python2.7/dist-packages'
pythondir='${prefix}/lib/python2.7/dist-packages'
sbindir='/usr/sbin'
sharedstatedir='/com'
sysconfdir='/etc'
systemdsystemunitdir=''
```



```

target_alias='arm-oe-linux-gnueabi'

## ----- ##
## confdefs.h. ##
## ----- ##

/* confdefs.h */
#define PACKAGE_NAME "dbus"
#define PACKAGE_TARNAME "dbus"
#define PACKAGE_VERSION "1.6.8"
#define PACKAGE_STRING "dbus 1.6.8"
#define PACKAGE_BUGREPORT
"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
#define PACKAGE_URL ""
#define PACKAGE "dbus"
#define VERSION "1.6.8"
#define GETTEXT_PACKAGE "dbus-1"
#define DBUS_DAEMON_NAME "dbus-daemon"
#define STDC_HEADERS 1
#define HAVE_SYS_TYPES_H 1
#define HAVE_SYS_STAT_H 1
#define HAVE_STDLIB_H 1
#define HAVE_STRING_H 1
#define HAVE_MEMORY_H 1
#define HAVE_STRINGS_H 1
#define HAVE_INTTYPES_H 1
#define HAVE_STDINT_H 1
#define HAVE_UNISTD_H 1
#define __EXTENSIONS__ 1
#define _ALL_SOURCE 1
#define _GNU_SOURCE 1
#define _POSIX_PTHREAD_SEMANTICS 1
#define _TANDEM_SOURCE 1
#define STDC_HEADERS 1
#define HAVE_DLFCN_H 1
#define LT_OBJDIR ".libs/"
#define DBUS_UNIX 1
#define DBUS_WITH_GLIB 1
#define DBUS_DISABLE_ASSERT 1
#define DBUS_DISABLE_CHECKS 1
#define G_DISABLE_CHECKS 1
#define DBUS_ENABLE_USERDB_CACHE 1
#define SIZEOF_CHAR 1
#define SIZEOF_SHORT 2
#define SIZEOF_LONG 4
#define SIZEOF_INT 4
#define SIZEOF_VOID_P 4
#define SIZEOF_LONG_LONG 8
#define SIZEOF___INT64 0
#define DBUS_INT64_PRINTF_MODIFIER "ll"
#define DBUS_VA_COPY va_copy
#define DBUS_USE_SYNC 1

```

```
#define HAVE_VSNPRINTF 1
#define HAVE_VASPRINTF 1
#define HAVE_NANOSLEEP 1
#define HAVE_USLEEP 1
#define HAVE_SETENV 1
#define HAVE_CLEARENV 1
#define HAVE_UNSETENV 1
#define HAVE_SOCKETPAIR 1
#define HAVE_GETGROUPLIST 1
#define HAVE_FPATHCONF 1
#define HAVE_SETRLIMIT 1
#define HAVE_POLL 1
#define HAVE_SETLOCALE 1
#define HAVE_LOCALECONV 1
#define HAVE_STRTOULL 1
#define HAVE_STRTOULL 1
#define HAVE_GETRESUID 1
#define HAVE_SYSLOG_H 1
#define HAVE_DECL_LOG_PERROR 1
#define HAVE_DIRFD 1
#define HAVE_SYS_RESOURCE_H 1
#define HAVE_DIRENT_H 1
#define HAVE_EXECINFO_H 1
#define HAVE_BACKTRACE 1
#define HAVE_ERRNO_H 1
#define HAVE_SIGNAL_H 1
#define HAVE_LOCALE_H 1
#define HAVE_BYTESWAP_H 1
#define HAVE_UNISTD_H 1
#define HAVE_POSIX_GETPWNAM_R 1
#define HAVE_SOCKLEN_T 1
#define HAVE_SYS_UIO_H 1
#define HAVE_WRITEV 1
#define HAVE_DECL_MSG_NOSIGNAL 1
#define HAVE_ISO_VARARGS 1
#define HAVE_GNUC_VARARGS 1
#define HAVE_PIPE2 1
#define HAVE_ACCEPT4 1
#define HAVE_ABSTRACT_SOCKETS 1
#define HAVE_EXPAT_H 1
#define HAVE_MONOTONIC_CLOCK 1
#define HAVE_SYS_INOTIFY_H 1
#define DBUS_BUS_ENABLE_INOTIFY 1
#define HAVE_INOTIFY_INIT1 1
#define DBUS_HAVE_LINUX_EPOLL 1
#define HAVE_UNIX_FD_PASSING 1
#define X_DISPLAY_MISSING 1
#define DBUS_SYSTEM_SOCKET "/var/run/dbus/system_bus_socket"
#define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"unix:path=/var/run/dbus/system_bus_socket"
#define DBUS_CONSOLE_AUTH_DIR "/var/run/console/"
#define DBUS_CONSOLE_OWNER_FILE ""
```

```

#define DBUS_USER "messagebus"
#define DBUS_PREFIX "/usr"
#define DBUS_DATADIR "/usr/share"
#define DBUS_DAEMONDIR "/usr/bin"
#define DBUS_BINDIR "/usr/bin"
#define DBUS_LIBEXECDIR "/usr/lib/dbus"
#define DBUS_TEST_EXEC
"/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/test"
#define DBUS_EXEEXT ""
#define TEST_BUS_BINARY
"/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon"
#define DBUS_TEST_LAUNCH_HELPER_BINARY
"/home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-launch-helper-test"
#define DBUS_TEST_SOCKET_DIR "/tmp"
#define TEST_LISTEN "unix:tmpdir=/tmp"
#define DBUS_SESSION_SOCKET_DIR "/tmp"

```

configure: exit 0

File = config.status

```

#!/bin/sh
# Generated by configure.
# Run this file to recreate the current configuration.
# Compiler output produced by configure, useful for debugging
# configure, is in config.log if it exists.

debug=false
ac_cs_recheck=false
ac_cs_silent=false

SHELL=${CONFIG_SHELL-/bin/sh}
export SHELL
## ----- ##
## M4sh Initialization. ##
## ----- ##

# Be more Bourne compatible
DUALCASE=1; export DUALCASE # for MKS sh
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
:
  emulate sh
  NULLCMD=:
  # Pre-4.2 versions of Zsh do word splitting on ${1+"$@"}, which

```



```

    as_echo='sh -c $as_echo_body as_echo'
fi

# The user is always right.
if test "${PATH_SEPARATOR+set}" != set; then
    PATH_SEPARATOR=:
    (PATH='/bin;/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 && {
        (PATH='/bin:/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 ||
            PATH_SEPARATOR=';'
    }
fi

# IFS
# We need space, tab and new line, in precisely that order. Quoting
# is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " $as_nl

# Find who we are. Look in the path if we contain no directory
# separator.
as_myself=
case $0 in #((
    *[\ \/]* ) as_myself=$0 ;;
    *) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done
IFS=$as_save_IFS

    ;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
    as_myself=$0
fi
if test ! -f "$as_myself"; then
    $as_echo "$as_myself: error: cannot find myself; rerun with an
absolute file name" >&2
    exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
# 1"
# suppresses any "Segmentation fault" message there. '(( could

```

```

# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH
do eval test x\${$as_var+set} = xset \
  && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

# NLS nuisances.
LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
# are
# provided, also output the error to LOG_FD, referencing LINENO. Then
# exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
  as_status=$1; test $as_status -eq 0 && as_status=1
  if test "$4"; then
    as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
  fi
  $as_echo "$as_me: error: $2" >&2
  as_fn_exit $as_status
} # as_fn_error

# as_fn_set_status STATUS
# -----
# Set $? to STATUS, without forking.
as_fn_set_status ()
{
  return $1
} # as_fn_set_status

# as_fn_exit STATUS
# -----
# Exit the shell with STATUS, even in a "trap 0" or "set -e" context.
as_fn_exit ()

```

```

{
  set +e
  as_fn_set_status $1
  exit $1
} # as_fn_exit

# as_fn_unset VAR
# -----
# Portably unset VAR.
as_fn_unset ()
{
  { eval $1=; unset $1;}
}
as_unset=as_fn_unset
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
# VAR. Take
# advantage of any shell optimizations that allow amortized linear
# growth over
# repeated appends, instead of the typical quadratic growth present in
# naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\${as_var} = x12") 2>/dev/null;
then :
  eval 'as_fn_append ()
  {
    eval $1+=\${2}
  }'
else
  as_fn_append ()
  {
    eval $1=\${$1}\${2}
  }
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
# the
# global $as_val. Take advantage of shells that can avoid forks. The
# arguments
# must be portable across $(( )) and expr.
if (eval "test \${(( 1 + 1 ))} = 2") 2>/dev/null; then :
  eval 'as_fn_arith ()
  {
    as_val=$(( $* ))
  }'
else
  as_fn_arith ()
  {
    as_val=`expr "$@" || test $? -eq 1`
  }
fi

```

```

    }
fi # as_fn_arith

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
"X/"; then
    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\/\([^\/]*\)/*$' \|| \
X"$0" : 'X\(/\)\$' \|| \
X"$0" : 'X\(/\)\' \|| . 2>/dev/null ||
$as_echo X/"$0" |
sed '/^.*\/\([^\/]*\)\/*$/ {
    s//\1/
    q
}
/^X\/\(\(\)\)/ {
    s//\1/
    q
}
/^X\/\(\(\)\).*/ {
    s//\1/
    q
}
s/.*\/./; q'`

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyz'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

ECHO_C= ECHO_N= ECHO_T=

```



```

case `echo -n x` in #((((
-n*)
  case `echo 'xy\c'` in
  *c*) ECHO_T=' ';; # ECHO_T is single tab character.
  xy) ECHO_C='\c';;
  *) echo `echo ksh88 bug on AIX 6.1` > /dev/null
     ECHO_T=' ';;
  esac;;
*)
  ECHO_N='-n';;
esac

rm -f conf$$ conf$$exe conf$$file
if test -d conf$$dir; then
  rm -f conf$$dir/conf$$file
else
  rm -f conf$$dir
  mkdir conf$$dir 2>/dev/null
fi
if (echo >conf$$file) 2>/dev/null; then
  if ln -s conf$$file conf$$ 2>/dev/null; then
    as_ln_s='ln -s'
    # ... but there are two gotchas:
    # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
    # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
    executable.
    # In both cases, we have to default to `cp -pR'.
    ln -s conf$$file conf$$dir 2>/dev/null && test ! -f conf$$exe
  ||
    as_ln_s='cp -pR'
  elif ln conf$$file conf$$ 2>/dev/null; then
    as_ln_s=ln
  else
    as_ln_s='cp -pR'
  fi
else
  as_ln_s='cp -pR'
fi
rm -f conf$$ conf$$exe conf$$dir/conf$$file conf$$file
rmdir conf$$dir 2>/dev/null

# as_fn_mkdir_p
# -----
# Create "$as_dir" as a directory, including parents if necessary.
as_fn_mkdir_p ()
{
  case $as_dir in #(
  -*) as_dir=./$as_dir;;
  esac
  test -d "$as_dir" || eval $as_mkdir_p || {

```



```

    test -f "$1" && test -x "$1"
} # as_fn_executable_p
as_test_x='test -x'
as_executable_p=as_fn_executable_p

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed
'y%*$as_cr_letters%P$as_cr_LETTERS%;s%[^_$as_cr_alnum]%%_g'"

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_$as_cr_alnum]%%_g'"

exec 6>&1
## ----- ##
## Main body of $CONFIG_STATUS script. ##
## ----- ##
# Save the log message, to keep $0 and so on meaningful, and to
# report actual input values of CONFIG_FILES etc. instead of their
# values after options handling.
ac_log="
This file was extended by dbus-glib $as_me 0.100.2, which was
generated by GNU Autoconf 2.69.  Invocation command line was

    CONFIG_FILES    = $CONFIG_FILES
    CONFIG_HEADERS  = $CONFIG_HEADERS
    CONFIG_LINKS    = $CONFIG_LINKS
    CONFIG_COMMANDS = $CONFIG_COMMANDS
$ $0 $@

on `(hostname || uname -n) 2>/dev/null | sed 1q`
"

# Files that config.status was made for.
config_files=" Makefile m4/Makefile doc/Makefile
doc/reference/Makefile doc/reference/version.xml dbus/Makefile
dbus/examples/Makefile dbus/examples/statemachine/Makefile
test/Makefile test/core/Makefile test/interfaces/Makefile
test/data/valid-service-files/debug-glib.service test/data/valid-
service-files/debug-echo.service test/data/valid-service-
files/interfaces-test.service test/lib/Makefile test/manual/Makefile
tools/Makefile dbus-glib-1.pc dbus-glib-1-uninstalled.pc"
config_headers=" config.h"
config_commands=" depfiles libtool"

ac_cs_usage="\
`$as_me' instantiates files and other configuration actions
from templates according to the current configuration.  Unless the
files
and actions are specified as TAGs, all are instantiated by default.

Usage: $0 [OPTION]... [TAG]..."

```

-h, --help print this help, then exit
-V, --version print version number and configuration settings,
then exit
--config print configuration, then exit
-q, --quiet, --silent do not print progress messages
-d, --debug don't remove temporary files
--recheck update \$as_me by reconfiguring in the same
conditions
--file=FILE[:TEMPLATE] instantiate the configuration file FILE
--header=FILE[:TEMPLATE] instantiate the configuration header FILE

Configuration files:
\$config_files

Configuration headers:
\$config_headers

Configuration commands:
\$config_commands

Report bugs to
<https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLib>."

```
ac_cs_config="--build=i686-linux' '--host=arm-oe-linux-gnueabi' '--target=arm-oe-linux-gnueabi' '--prefix=/usr' '--exec_prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib/dbus-glib' '--datadir=/usr/share' '--sysconfdir=/etc' '--sharedstatedir=/com' '--localstatedir=/var' '--libdir=/usr/lib' '--includedir=/usr/include' '--oldincludedir=/usr/include' '--infodir=/usr/share/info' '--mandir=/usr/share/man' '--disable-silent-rules' '--disable-dependency-tracking' '--with-libtool-sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' '--with-introspect-xml=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/share/dbus/dbus-bus-introspect.xml' '--with-dbus-binding-tool=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/dbus-binding-tool' '--enable-nls' 'build_alias=i686-linux' 'host_alias=arm-oe-linux-gnueabi' 'target_alias=arm-oe-linux-gnueabi' 'CC=arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' 'CFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types' 'LDFLAGS=-
```

```

L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-style=gnu'
'CPPFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector'
'CPP=arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
'PKG_CONFIG_PATH=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig:/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-
vp1/usr/share/pkgconfig'
'PKG_CONFIG_LIBDIR=/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1//usr/lib/pkgconfig'"
ac_cs_version="\
dbus-glib config.status 0.100.2
configured by /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/configure, generated by
GNU Autoconf 2.69,
  with options `"$ac_cs_config`"

```

Copyright (C) 2012 Free Software Foundation, Inc.
This config.status script is free software; the Free Software
Foundation
gives unlimited permission to copy, distribute and modify it."

```

ac_pwd='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100.2-r0/dbus-glib-0.100.2'
srcdir='.'
INSTALL='/usr/bin/install -c'
MKDIR_P='/bin/mkdir -p'
AWK='gawk'
test -n "$AWK" || AWK=awk
# The default lists apply if the user does not specify any file.
ac_need_defaults=:
while test $# != 0
do
  case $1 in
    --*=?*)
      ac_option=`expr "X$1" : 'X\[^\]=*\)`
      ac_optarg=`expr "X$1" : 'X\[^\]=*\ (.*)`'
      ac_shift=:
      ;;
    --*=)
      ac_option=`expr "X$1" : 'X\[^\]=*\)`
      ac_optarg=
      ac_shift=:

```

```

;;
*)
  ac_option=$1
  ac_optarg=$2
  ac_shift=shift
  ;;
esac

case $ac_option in
# Handling of the options.
-recheck | --recheck | --recheck | --reche | --rech | --rec | --re |
--r)
  ac_cs_recheck=: ;;
--version | --versio | --versi | --vers | --ver | --ve | --v | -V )
  $as_echo "$ac_cs_version"; exit ;;
--config | --confi | --conf | --con | --co | --c )
  $as_echo "$ac_cs_config"; exit ;;
--debug | --debu | --deb | --de | --d | -d )
  debug=: ;;
--file | --fil | --fi | --f )
  $ac_shift
  case $ac_optarg in
*'*) ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\'/g"`
;;
') as_fn_error $? "missing file argument" ;;
esac
as_fn_append CONFIG_FILES " '$ac_optarg'"
ac_need_defaults=false;;
--header | --heade | --head | --hea )
  $ac_shift
  case $ac_optarg in
*'*) ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\'/g"`
;;
esac
as_fn_append CONFIG_HEADERS " '$ac_optarg'"
ac_need_defaults=false;;
--he | --h)
  # Conflict between --help and --header
  as_fn_error $? "ambiguous option: \`$1'"
Try \`$0 --help' for more information.";;
--help | --hel | -h )
  $as_echo "$ac_cs_usage"; exit ;;
-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil | --si | --s)
  ac_cs_silent=: ;;

# This is an error.
-*) as_fn_error $? "unrecognized option: \`$1'"
Try \`$0 --help' for more information." ;;

*) as_fn_append ac_config_targets " $1"
  ac_need_defaults=false ;;

```

```

    esac
    shift
done

ac_configure_extra_args=

if $ac_cs_silent; then
    exec 6>/dev/null
    ac_configure_extra_args="$ac_configure_extra_args --silent"
fi

if $ac_cs_recheck; then
    set X /bin/sh '/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/configure' '--build=i686-linux' '--host=arm-oe-linux-gnueabi' '--target=arm-oe-linux-gnueabi' '--prefix=/usr' '--exec_prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib/dbus-glib' '--datadir=/usr/share' '--sysconfdir=/etc' '--sharedstatedir=/com' '--localstatedir=/var' '--libdir=/usr/lib' '--includedir=/usr/include' '--oldincludedir=/usr/include' '--infodir=/usr/share/info' '--mandir=/usr/share/man' '--disable-silent-rules' '--disable-dependency-tracking' '--with-libtool-sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' '--with-introspect-xml=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/share/dbus/dbus-bus-introspect.xml' '--with-dbus-binding-tool=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/dbus-binding-tool' '--enable-nls' 'build_alias=i686-linux' 'host_alias=arm-oe-linux-gnueabi' 'target_alias=arm-oe-linux-gnueabi' 'CC=arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' 'CFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types' 'LDFLAGS=-L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-O1 -Wl,--hash-style=gnu' 'CPPFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector' 'CPP=arm-linux-gnueabi-gcc -E --sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8' 'PKG_CONFIG_PATH=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-

```

```

vp1//usr/lib/pkgconfig:/home/gangadhar/newyoctobuild/tisdk/build/arago-
-tmp-external-linaro-toolchain/sysroots/delphi-
vp1/usr/share/pkgconfig'
'PKG_CONFIG_LIBDIR=/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1//usr/lib/pkgconfig'
$ac_configure_extra_args --no-create --no-recursion
  shift
  $as_echo "running CONFIG_SHELL=/bin/sh $" >&6
  CONFIG_SHELL='/bin/sh'
  export CONFIG_SHELL
  exec "$@"
fi

exec 5>>config.log
{
  echo
  sed 'h;s/./-/g;s/^\.../## /;s/...$/ ##/;p;x;p;x' <<_ASBOX
## Running $as_me. ##
  _ASBOX
  $as_echo "$ac_log"
} >&5

#
# INIT-COMMANDS
#
AMDEP_TRUE="#" ac_aux_dir="."

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

sed_quote_subst='s/\([\"`$\\]\)/\\\1/g'
double_quote_subst='s/\([\"`\\]\)/\\\1/g'
delay_variable_subst='s/\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\/$/\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\/g'
macro_version='2.4.2'
macro_revision='1.3337'
enable_shared='yes'
enable_static='yes'
pic_mode='default'
enable_fast_install='yes'
SHELL='/bin/sh'
ECHO='printf %s\n'
PATH_SEPARATOR=':'
host_alias='arm-oe-linux-gnueabi'
host='arm-oe-linux-gnueabi'
host_os='linux-gnueabi'
build_alias='i686-linux'
build='i686-pc-linux-gnu'
build_os='linux-gnu'
SED='/bin/sed'
Xsed='/bin/sed -e 1s/^X//'

```



```
GREP='/bin/grep'
EGREP='/bin/grep -E'
FGREP='/bin/grep -F'
LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
NM='arm-linux-gnueabi-hf-nm'
LN_S='ln -s'
max_cmd_len='1572864'
ac_objext='o'
exeext=''
lt_unset='unset'
lt_SP2NL='tr \040 \012'
lt_NL2SP='tr \015\012 \040\040'
lt_cv_to_host_file_cmd='func_convert_file_noop'
lt_cv_to_tool_file_cmd='func_convert_file_noop'
reload_flag=' -r'
reload_cmds='$LD$reload_flag -o $output$reload_objs'
OBJDUMP='arm-linux-gnueabi-hf-objdump'
deplibs_check_method='pass_all'
file_magic_cmd='$MAGIC_CMD'
file_magic_glob=''
want_nocaseglob='no'
DLLTOOL='false'
sharedlib_from_linklib_cmd='printf %s\n'
AR='arm-linux-gnueabi-hf-ar'
AR_FLAGS='cru'
archiver_list_spec='@'
STRIP='arm-linux-gnueabi-hf-strip'
RANLIB='arm-linux-gnueabi-hf-ranlib'
old_postinstall_cmds='chmod 644 $oldlib~$RANLIB $tool_oldlib'
old_postuninstall_cmds=''
old_archive_cmds='$AR $AR_FLAGS $oldlib$oldobjs~$RANLIB $tool_oldlib'
lock_old_archive_extraction='no'
CC='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
CFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types -Wall -Wchar-subscripts -
Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-
arith -Wcast-align -Wfloat-equal -Wsign-compare -fno-strict-aliasing'
compiler='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
GCC='yes'
lt_cv_sys_global_symbol_pipe='sed -n -e '\''s/^.*[
]\([ABCDGIRSTW][ABCDGIRSTW]*\) [ ] [ ] * \([_A-Za-z][_A-Za-z0-
9]*\) $\1 \2 \2/p'\'' | sed '\''/_gnu_lto/d'\''
```

```

lt_cv_sys_global_symbol_to_cdecl='sed -n -e '\''s/^T .* \(.*\)$/extern
int \1();/p\' -e '\''s/^[ABCDGIRSTW]* .* \(.*\)$/extern char
\1;/p\''''
lt_cv_sys_global_symbol_to_c_name_address='sed -n -e '\''s/^: \([^
]*\) [ ]*$/ {"\1", (void *) 0},/p\' -e '\''s/^[ABCDGIRSTW]* \([^
]*\) \([^ ]*\)$/ {"\2", (void *) \&\2},/p\''''
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix='sed -n -e
'\''s/^: \([^ ]*\) [ ]*$/ {"\1", (void *) 0},/p\' -e
'\''s/^[ABCDGIRSTW]* \([^ ]*\) \(\lib[^ ]*\)$/ {"\2", (void *)
\&\2},/p\' -e '\''s/^[ABCDGIRSTW]* \([^ ]*\) \([^ ]*\)$/ {"lib\2",
(void *) \&\2},/p\''''
nm_file_list_spec='@'
lt_sysroot='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-vpl'
objdir='.libs'
MAGIC_CMD='file'
lt_prog_compiler_no_builtin_flag=' -fno-builtin'
lt_prog_compiler_pic=' -fPIC -DPIC'
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_static='-static'
lt_cv_prog_compiler_c_o='yes'
need_locks='no'
MANIFEST_TOOL=':'
DSYMUTIL=''
NMEDIT=''
LIPO=''
OTOOL=''
OTOOL64=''
libext='a'
shrext_cmds='.so'
extract_expsyms_cmds=''
archive_cmds_need_lc='no'
enable_shared_with_static_runtimes='no'
export_dynamic_flag_spec='${wl}--export-dynamic'
whole_archive_flag_spec='${wl}--whole-archive$convenience ${wl}--no-
whole-archive'
compiler_needs_object='no'
old_archive_from_new_cmds=''
old_archive_from_expsyms_cmds=''
archive_cmds='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
archive_expsym_cmds='echo "{ global:" > $output_objdir/$libname.ver~
cat $export_symbols | sed -e "s/\(.*\)/\1;/" >>
$output_objdir/$libname.ver~
echo "local: *; };" >> $output_objdir/$libname.ver~
$CC -shared $pic_flag $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname ${wl}-version-script
${wl}$output_objdir/$libname.ver -o $lib'
module_cmds=''
module_expsym_cmds=''
with_gnu_ld='yes'
allow_undefined_flag=''

```

```
no_undefined_flag=''
hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
hardcode_libdir_separator=''
hardcode_direct='no'
hardcode_direct_absolute='no'
hardcode_minus_L='no'
hardcode_shlibpath_var='unsupported'
hardcode_automatic='no'
inherit_rpath='no'
link_all_deplibs='unknown'
always_export_symbols='no'
export_symbols_cmds='$NM $libobjs $convenience | $global_symbol_pipe |
$SED '\''s/.* //' | sort | uniq > $export_symbols'
exclude_expsyms='_GLOBAL_OFFSET_TABLE_|_GLOBAL__[FID]_.*'
include_expsyms=''
prelink_cmds=''
postlink_cmds=''
file_list_spec=''
variables_saved_for_relink='PATH LD_LIBRARY_PATH LD_RUN_PATH
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH'
need_lib_prefix='no'
need_version='no'
version_type='linux'
runpath_var='LD_RUN_PATH'
shlibpath_var='LD_LIBRARY_PATH'
shlibpath_overrides_runpath='no'
libname_spec='lib$name'
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
install_override_mode=''
postinstall_cmds=''
postuninstall_cmds=''
finish_cmds='PATH="$PATH:/sbin" ldconfig -n $libdir'
finish_eval=''
hardcode_into_libs='yes'
sys_lib_search_path_spec='/home/gangadhar/gcc-linaro-arm-linux-
gnueabihf-4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf/4.7.3
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/lib/gcc /home/gangadhar/gcc-linaro-arm-linux-gnueabihf-
4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabihf
/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-
20130313_linux/arm-linux-gnueabihf/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib '
sys_lib_dlsearch_path_spec='/usr/lib /lib'
hardcode_action='immediate'
enable_dlopen='unknown'
enable_dlopen_self='unknown'
enable_dlopen_self_static='unknown'
```



```

    "doc/reference/Makefile") CONFIG_FILES="$CONFIG_FILES
doc/reference/Makefile" ;;
    "doc/reference/version.xml") CONFIG_FILES="$CONFIG_FILES
doc/reference/version.xml" ;;
    "dbus/Makefile") CONFIG_FILES="$CONFIG_FILES dbus/Makefile" ;;
    "dbus/examples/Makefile") CONFIG_FILES="$CONFIG_FILES
dbus/examples/Makefile" ;;
    "dbus/examples/statemachine/Makefile") CONFIG_FILES="$CONFIG_FILES
dbus/examples/statemachine/Makefile" ;;
    "test/Makefile") CONFIG_FILES="$CONFIG_FILES test/Makefile" ;;
    "test/core/Makefile") CONFIG_FILES="$CONFIG_FILES
test/core/Makefile" ;;
    "test/interfaces/Makefile") CONFIG_FILES="$CONFIG_FILES
test/interfaces/Makefile" ;;
    "test/data/valid-service-files/debug-glib.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/debug-
glib.service" ;;
    "test/data/valid-service-files/debug-echo.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/debug-
echo.service" ;;
    "test/data/valid-service-files/interfaces-test.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/interfaces-
test.service" ;;
    "test/lib/Makefile") CONFIG_FILES="$CONFIG_FILES
test/lib/Makefile" ;;
    "test/manual/Makefile") CONFIG_FILES="$CONFIG_FILES
test/manual/Makefile" ;;
    "tools/Makefile") CONFIG_FILES="$CONFIG_FILES tools/Makefile" ;;
    "dbus-glib-1.pc") CONFIG_FILES="$CONFIG_FILES dbus-glib-1.pc" ;;
    "dbus-glib-1-uninstalled.pc") CONFIG_FILES="$CONFIG_FILES dbus-
glib-1-uninstalled.pc" ;;

    *) as_fn_error $? "invalid argument: \`${ac_config_target}'" "$LINENO"
5;;
    esac
done

```

```

# If the user did not use the arguments to specify the items to
# instantiate,
# then the envvar interface is used. Set only those that are not.
# We use the long form for the default assignment because of an
# extremely
# bizarre bug on SunOS 4.1.3.
if $ac_need_defaults; then
    test "${CONFIG_FILES+set}" = set || CONFIG_FILES=$config_files
    test "${CONFIG_HEADERS+set}" = set || CONFIG_HEADERS=$config_headers
    test "${CONFIG_COMMANDS+set}" = set ||
CONFIG_COMMANDS=$config_commands
fi

```

```

# Have a temporary directory for convenience.  Make it in the build
tree
# simply because there is no reason against having it here, and in
addition,
# creating and moving files from /tmp can sometimes cause problems.
# Hook for its removal unless debugging.
# Note that there is a small window in which the directory will not be
cleaned:
# after its creation but before its name has been assigned to `$tmp'.
$debug ||
{
    tmp= ac_tmp=
    trap 'exit_status=$?
: "${ac_tmp:=$tmp}"
{ test ! -d "$ac_tmp" || rm -fr "$ac_tmp"; } && exit $exit_status
' 0
    trap 'as_fn_exit 1' 1 2 13 15
}
# Create a (secure) tmp directory for tmp files.

{
    tmp=`(umask 077 && mktemp -d "./confXXXXXX") 2>/dev/null` &&
    test -d "$tmp"
} ||
{
    tmp=./conf$$-$RANDOM
    (umask 077 && mkdir "$tmp")
} || as_fn_error $? "cannot create a temporary directory in ."
"$LINENO" 5
ac_tmp=$tmp

# Set up the scripts for CONFIG_FILES section.
# No need to generate them if there are no CONFIG_FILES.
# This happens for instance with `./config.status config.h'.
if test -n "$CONFIG_FILES"; then

ac_cr=`echo X | tr X '\015'`
# On cygwin, bash can eat \r inside `` if the user requested igncr.
# But we know of no other shell where ac_cr would be empty at this
# point, so we can use a bashism as a fallback.
if test "x$ac_cr" = x; then
    eval ac_cr=\$\'\r\'
fi
ac_cs_awk_cr=`$AWK 'BEGIN { print "a\rb" }' </dev/null 2>/dev/null`
if test "$ac_cs_awk_cr" = "a${ac_cr}b"; then
    ac_cs_awk_cr='\r'
else
    ac_cs_awk_cr=$ac_cr
fi

echo 'BEGIN {' >"$ac_tmp/subs1.awk" &&

```

```
cat >>"$ac_tmp/subs1.awk" <<\_ACAWK &&
S["am__EXEEXT_FALSE"]=""
S["am__EXEEXT_TRUE"]="#"
S["LTLIBOBJS"]=""
S["LIBOBJS"]=""
S["TEST_SOCKET_DIR"]=""
S["ABSOLUTE_TOP_BUILDDIR"]="/home/gangadhar/newyoctobuild/tisdk/build/
arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-
linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2"
S["TEST_SLEEP_FOREVER_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/bu
ild/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/test-sleep-forever"
S["TEST_SEGFAULT_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/build/a
rago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-
linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/test-segfault"
S["TEST_EXIT_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/build/arago
-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/test-exit"
S["TEST_INTERFACES_SERVICE_BINARY"]="/home/gangadhar/newyoctobuild/tis
dk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-
3.8-oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/interfaces/test-service"
S["TEST_CORE_SERVICE_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/bui
ld/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/core/test-service-glib"
S["TEST_SHELL_SERVICE_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/bu
ild/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/test-shell-service"
S["TEST_SERVICE_BINARY"]="/home/gangadhar/newyoctobuild/tisdk/build/ar
ago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-
linux-gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/test-service"
S["TEST_SERVICE_DIR"]="/home/gangadhar/newyoctobuild/tisdk/build/arago
-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus-glib/0.100.2-r0/dbu"\
"s-glib-0.100.2/test/data/valid-service-files"
S["EXPANDED_DATADIR"]="/usr/share"
S["EXPANDED_LIBDIR"]="/usr/lib"
S["EXPANDED_BINDIR"]="/usr/bin"
S["EXPANDED_SYSCONFDIR"]="/etc"
S["EXPANDED_LOCALSTATEDIR"]="/var"
S["GTK_DOC_USE_REBASE_FALSE"]="#"
S["GTK_DOC_USE_REBASE_TRUE"]=""
S["GTK_DOC_USE_LIBTOOL_FALSE"]="#"
S["GTK_DOC_USE_LIBTOOL_TRUE"]=""
S["GTK_DOC_BUILD_PDF_FALSE"]=""
```



```
S["GTK_DOC_BUILD_PDF_TRUE"]="#"
S["GTK_DOC_BUILD_HTML_FALSE"]="#"
S["GTK_DOC_BUILD_HTML_TRUE"]=" "
S["ENABLE_GTK_DOC_FALSE"]=" "
S["ENABLE_GTK_DOC_TRUE"]="#"
S["GTKDOC_DEPS_LIBS"]=" "
S["GTKDOC_DEPS_CFLAGS"]=" "
S["HTML_DIR"]="${datadir}/gtk-doc/html"
S["GTKDOC_MKPDF"]=" "
S["GTKDOC_REBASE"]="true"
S["GTKDOC_CHECK"]=" "
S["DBUS_GLIB_TOOL_LIBS"]="-lexpat"
S["DBUS_GLIB_TOOL_CFLAGS"]=" "
S["GLIB_GENMARSHAL"]="glib-genmarshal"
S["HAVE_GLIB_THREADS_FALSE"]="#"
S["HAVE_GLIB_THREADS_TRUE"]=" "
S["DBUS_GLIB_THREADS_LIBS"]="-pthread -lgthread-2.0 -lrt -lglib-2.0  "
S["DBUS_GLIB_THREADS_CFLAGS"]="-pthread -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -
I/home/gangadhar/"\
"newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include  "
S["DBUS_GLIB_LIBS"]="-lgio-2.0 -lgobject-2.0 -lglib-2.0  "
S["DBUS_GLIB_CFLAGS"]="-pthread -
I/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/include/glib-2.0 -
I/home/gangadhar/"\
"newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/glib-2.0/include  "
S["DBUS_LIBS"]="-ldbus-1  "
S["DBUS_CFLAGS"]="-I/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1/usr/include/dbus-1.0
-I/home/gangadhar/newyoctob\
uild/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-
vp1/usr/lib/dbus-1.0/include  "
S["PKG_CONFIG_LIBDIR"]="/home/gangadhar/newyoctobuild/tisdk/build/arag
o-tmp-external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig"
S["PKG_CONFIG_PATH"]="/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-
vp1//usr/lib/pkgconfig:/home/gangadhar/newyoctobuild/t"\
"isdsk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-
vp1/usr/share/pkgconfig"
S["PKG_CONFIG"]="/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config"
S["DBUS_PATH_OR_ABSTRACT"]="abstract"
S["OTOOL64"]=" "
S["OTOOL"]=" "
S["LIPO"]=" "
S["NMEDIT"]=" "
S["DSYMUTIL"]=" "
```

```
S["MANIFEST_TOOL"]=":"
S["RANLIB"]="arm-linux-gnueabi-hf-ranlib"
S["ac_ct_AR"]=""
S["AR"]="arm-linux-gnueabi-hf-ar"
S["DLLTOOL"]="false"
S["OBJDUMP"]="arm-linux-gnueabi-hf-objdump"
S["LN_S"]="ln -s"
S["NM"]="arm-linux-gnueabi-hf-nm"
S["ac_ct_DUMPBIN"]=""
S["DUMPBIN"]=""
S["LD"]="arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"
S["FGREP"]="bin/grep -F"
S["SED"]="bin/sed"
S["LIBTOOL"]="$(top_builddir)/arm-oe-linux-gnueabi-libtool"
S["DBUS_BUILD_TESTS_FALSE"]=""
S["DBUS_BUILD_TESTS_TRUE"]="#"
S["DBUS_BINDING_TOOL"]="home/gangadhar/newyoctobuild/tisdk/build/arag
o-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/dbus-
binding-tool"
S["DBUS_BASH_COMPLETION_FALSE"]="#"
S["DBUS_BASH_COMPLETION_TRUE"]=""
S["EGREP"]="bin/grep -E"
S["GREP"]="bin/grep"
S["CPP"]="arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7"
-a -marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-
a8"
S["am__fastdepCC_FALSE"]=""
S["am__fastdepCC_TRUE"]="#"
S["CCDEPMODE"]="depmode=none"
S["am__nodep"]=""
S["AMDEPBACKSLASH"]=""
S["AMDEP_FALSE"]=""
S["AMDEP_TRUE"]="#"
S["am__quote"]=""
S["am__include"]="include"
S["DEPDIR"]="deps"
S["OBJEXT"]="o"
S["EXEEXT"]=""
S["ac_ct_CC"]=""
S["CPPFLAGS"]="-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector"
S["LDFLAGS"]="-L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-l"
inux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-O1 -Wl,--hash-style=gnu"
```

```

S["CFLAGS"]="-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector -O2 -pipe -g -felim"\
"inate-unused-debug-types -Wall -Wchar-subscripts -Wmissing-declarations -Wmissing-prototypes -Wnested-externs -Wpointer-arith -Wcast-align -Wfloat-e"\
"qual -Wsign-compare -fno-strict-aliasing"
S["CC"]="arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild"\
"/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1"
S["LT_AGE"]="2"
S["LT_REVISION"]="2"
S["LT_CURRENT"]="4"
S["AM_BACKSLASH"]="\\"
S["AM_DEFAULT_VERBOSITY"]="1"
S["AM_DEFAULT_V"]="$ (AM_DEFAULT_VERBOSITY) "
S["AM_V"]="$ (V) "
S["MAINT"]="#"
S["MAINTAINER_MODE_FALSE"]=""
S["MAINTAINER_MODE_TRUE"]="#"
S["am__untar"]="$ ${TAR-tar} xf -"
S["am__tar"]="$ ${TAR-tar} chof - \"${stardir}\""
S["AMTAR"]="$ ${TAR-tar}"
S["am__leading_dot"]="."
S["SET_MAKE"]=""
S["AWK"]="gawk"
S["mkdir_p"]="$ (MKDIR_P) "
S["MKDIR_P"]="/bin/mkdir -p"
S["INSTALL_STRIP_PROGRAM"]="$ (install_sh) -c -s"
S["STRIP"]="arm-linux-gnueabi-strip"
S["install_sh"]="$ {SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-glib/0.100"\
".2-r0/dbus-glib-0.100.2/install-sh"
S["MAKEINFO"]="$ {SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-glib/0.100"\
".2-r0/dbus-glib-0.100.2/missing --run makeinfo"
S["AUTOHEADER"]="$ {SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-glib/0.100"\
".2-r0/dbus-glib-0.100.2/missing --run autoheader"
S["AUTOMAKE"]="$ {SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-glib/0.100"\
".2-r0/dbus-glib-0.100.2/missing --run automake-1.12"

```

```
S["AUTOCONF"]="${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100"\
".2-r0/dbus-glib-0.100.2/missing --run autoconf"
S["ACLOCAL"]="${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus-
glib/0.100"\
".2-r0/dbus-glib-0.100.2/missing --run aclocal-1.12 "
S["VERSION"]="0.100.2"
S["PACKAGE"]="dbus-glib"
S["CYGPATH_W"]="echo"
S["am__isrc"]=""
S["INSTALL_DATA"]="${INSTALL} -m 644"
S["INSTALL_SCRIPT"]="${INSTALL}"
S["INSTALL_PROGRAM"]="${INSTALL}"
S["host_os"]="linux-gnueabi"
S["host_vendor"]="oe"
S["host_cpu"]="arm"
S["host"]="arm-oe-linux-gnueabi"
S["build_os"]="linux-gnu"
S["build_vendor"]="pc"
S["build_cpu"]="i686"
S["build"]="i686-pc-linux-gnu"
S["target_alias"]="arm-oe-linux-gnueabi"
S["host_alias"]="arm-oe-linux-gnueabi"
S["build_alias"]="i686-linux"
S["LIBS"]=""
S["ECHO_T"]=""
S["ECHO_N"]="-n"
S["ECHO_C"]=""
S["DEFS"]="-DHAVE_CONFIG_H"
S["mandir"]="/usr/share/man"
S["localedir"]="${datarootdir}/locale"
S["libdir"]="/usr/lib"
S["psdir"]="${docdir}"
S["pdfdir"]="${docdir}"
S["dvidir"]="${docdir}"
S["htmldir"]="${docdir}"
S["infodir"]="/usr/share/info"
S["docdir"]="${datarootdir}/doc/${PACKAGE_TARNAME}"
S["oldincludedir"]="/usr/include"
S["includedir"]="/usr/include"
S["localstatedir"]="/var"
S["sharedstatedir"]="/com"
S["sysconfdir"]="/etc"
S["datadir"]="/usr/share"
S["datarootdir"]="${prefix}/share"
S["libexecdir"]="/usr/lib/dbus-glib"
S["sbindir"]="/usr/sbin"
S["bindir"]="/usr/bin"
```

```

S["program_transform_name"]="s,x,x,"
S["prefix"]="/usr"
S["exec_prefix"]="/usr"
S["PACKAGE_URL"]=""
S["PACKAGE_BUGREPORT"]="https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=Glib"
S["PACKAGE_STRING"]="dbus-glib 0.100.2"
S["PACKAGE_VERSION"]="0.100.2"
S["PACKAGE_TARNAME"]="dbus-glib"
S["PACKAGE_NAME"]="dbus-glib"
S["PATH_SEPARATOR"]=":"
S["SHELL"]="/bin/sh"
_ACAWK
cat >>"$ac_tmp/subs1.awk" <<_ACAWK &&
  for (key in S) S_is_set[key] = 1
  FS = " "
}
{
  line = $ 0
  nfields = split(line, field, "@")
  subst = 0
  len = length(field[1])
  for (i = 2; i < nfields; i++) {
    key = field[i]
    keylen = length(key)
    if (S_is_set[key]) {
      value = S[key]
      line = substr(line, 1, len) " " value " " substr(line, len +
keylen + 3)
      len += length(value) + length(field[++i])
      subst = 1
    } else
      len += 1 + keylen
  }

  print line
}

_ACAWK
if sed "s/$ac_cr//" < /dev/null > /dev/null 2>&1; then
  sed "s/$ac_cr\\$//; s/$ac_cr/$ac_cs_awk_cr/g"
else
  cat
fi < "$ac_tmp/subs1.awk" > "$ac_tmp/subs.awk" \
  || as_fn_error $? "could not setup config files machinery" "$LINENO"
5
fi # test -n "$CONFIG_FILES"

# Set up the scripts for CONFIG_HEADERS section.
# No need to generate them if there are no CONFIG_HEADERS.
# This happens for instance with `./config.status Makefile'.

```

```

if test -n "$CONFIG_HEADERS"; then
cat >"$ac_tmp/defines.awk" <<\_ACAWK ||
BEGIN {
D["PACKAGE_NAME"]=" \\"dbus-glib\\""
D["PACKAGE_TARNAME"]=" \\"dbus-glib\\""
D["PACKAGE_VERSION"]=" \\"0.100.2\\""
D["PACKAGE_STRING"]=" \\"dbus-glib 0.100.2\\""
D["PACKAGE_BUGREPORT"]="
\\"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GL
ib\\""
D["PACKAGE_URL"]=" \\"\""
D["PACKAGE"]=" \\"dbus-glib\\""
D["VERSION"]=" \\"0.100.2\\""
D["STDC_HEADERS"]=" 1"
D["DBUS_BASH_COMPLETION"]=" 1"
D["DBUS_DISABLE_ASSERT"]=" 1"
D["G_DISABLE_ASSERT"]=" 1"
D["HAVE_SYS_TYPES_H"]=" 1"
D["HAVE_SYS_STAT_H"]=" 1"
D["HAVE_STDLIB_H"]=" 1"
D["HAVE_STRING_H"]=" 1"
D["HAVE_MEMORY_H"]=" 1"
D["HAVE_STRINGS_H"]=" 1"
D["HAVE_INTTYPES_H"]=" 1"
D["HAVE_STDINT_H"]=" 1"
D["HAVE_UNISTD_H"]=" 1"
D["HAVE_DLFCN_H"]=" 1"
D["LT_OBJDIR"]=" \\".libs/\\""
D["HAVE_SOCKLEN_T"]=" 1"
D["HAVE_ABSTRACT_SOCKETS"]=" 1"
D["HAVE_EXPAT_H"]=" 1"
D["TEST_SERVICE_DIR"]="
\\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueab"
"i/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/data/valid-service-
files\\""
D["TEST_SERVICE_BINARY"]="
\\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnu"
"eabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-service\\""
D["TEST_SHELL_SERVICE_BINARY"]="
\\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-lin"
"ux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-shell-
service\\""
D["TEST_CORE_SERVICE_BINARY"]="
\\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linu"
"x-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/core/test-
service-glib\\""

```

```

D["TEST_INTERFACES_SERVICE_BINARY"]="
\ /home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-o" \
"e-linux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-
0.100.2/test/interfaces/test-service\" ""
D["TEST_EXIT_BINARY"]="
\ /home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueab" \
"i/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-exit\" ""
D["TEST_SEGFAULT_BINARY"]="
\ /home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gn" \
"ueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-segfault\" ""
D["TEST_SLEEP_FOREVER_BINARY"]="
\ /home/gangadhar/newyocbuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-lin" \
"ux-gnueabi/dbus-glib/0.100.2-r0/dbus-glib-0.100.2/test/test-sleep-
forever\" ""
D["DBUS_TEST_SOCKET_DIR"]=" \\" ""
  for (key in D) D_is_set[key] = 1
  FS = " "
}
/^[\t ]*#[\t ]*(define|undef)[\t
]+[_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ][_abcdefghijklmnopqrstu
vwxyz0123456789]*([\t (]|$)/ {
  line = $ 0
  split(line, arg, " ")
  if (arg[1] == "#") {
    defundef = arg[2]
    mac1 = arg[3]
  } else {
    defundef = substr(arg[1], 2)
    mac1 = arg[2]
  }
  split(mac1, mac2, "(") #)
  macro = mac2[1]
  prefix = substr(line, 1, index(line, defundef) - 1)
  if (D_is_set[macro]) {
    # Preserve the white space surrounding the "#".
    print prefix "define", macro P[macro] D[macro]
    next
  } else {
    # Replace #undef with comments. This is necessary, for example,
    # in the case of _POSIX_SOURCE, which is predefined and required
    # on some systems where configure will not decide to define it.
    if (defundef == "undef") {
      print "/*", prefix defundef, macro, "*/"
      next
    }
  }
}
}
{ print }

```



```

# Let's still pretend it is `configure' which instantiates (i.e.,
don't
# use $as_me), people would be surprised to read:
# /* config.h.  Generated by config.status. */
configure_input='Generated from '`
  $as_echo "$*" | sed 's|^[^:]*|/|;s|:[^:]*|/, |g'
  ` ' by configure.'
if test x"$ac_file" != x-; then
  configure_input="$ac_file. $configure_input"
  { $as_echo "$as_me:${as_lineno-$LINENO}: creating $ac_file" >&5
$as_echo "$as_me: creating $ac_file" >&6;}
fi
# Neutralize special characters interpreted by sed in replacement
strings.
case $configure_input in #(
*\&* | *\\|* | *\\*)
  ac_sed_conf_input=`$as_echo "$configure_input" |
  sed 's/[\\|&|]/\\\\&/g'`; # (
*) ac_sed_conf_input=$configure_input;;
esac

case $ac_tag in
*:-:* | *:-) cat >"$ac_tmp/stdin" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5 ;;
esac
;;
esac

ac_dir=`$as_dirname -- "$ac_file" ||
$as_expr X"$ac_file" : 'X\([^/]\)\/*[^/]\/*$' \| \
  X"$ac_file" : 'X\(/\/)\[^/]' \| \
  X"$ac_file" : 'X\(/\/)\$' \| \
  X"$ac_file" : 'X\(/\/)' \| . 2>/dev/null ||
$as_echo X"$ac_file" |
  sed '/^X\([^/]\)\/*[^/]\/*$/{
    s//\1/
    q
  }
/^X\(/\/\/)\[^/].*/{
  s//\1/
  q
}
/^X\(/\/\/)\$/{
  s//\1/
  q
}
/^X\(/\/)\.*/{
  s//\1/
  q
}
s/././; q'`

```

```

as_dir="$ac_dir"; as_fn_mkdir_p
ac_builddir=.

case "$ac_dir" in
.) ac_dir_suffix= ac_top_builddir_sub=. ac_top_build_prefix= ;;
*)
  ac_dir_suffix=`$as_echo "$ac_dir" | sed 's|^\.([\//]|||)`
  # A ".." for each directory in $ac_dir_suffix.
  ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/[^\\/]*/|/..|g;s|/|||`
  case $ac_top_builddir_sub in
  "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
  *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
  esac ;;
esac
ac_abs_top_builddir=$ac_pwd
ac_abs_builddir=$ac_pwd$ac_dir_suffix
# for backward compatibility:
ac_top_builddir=$ac_top_build_prefix

case $srcdir in
.) # We are building in place.
  ac_srcdir=.
  ac_top_srcdir=$ac_top_builddir_sub
  ac_abs_top_srcdir=$ac_pwd ;;
[\\/] * | ?:[\\/] * ) # Absolute name.
  ac_srcdir=$srcdir$ac_dir_suffix;
  ac_top_srcdir=$srcdir
  ac_abs_top_srcdir=$srcdir ;;
*) # Relative name.
  ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
  ac_top_srcdir=$ac_top_build_prefix$srcdir
  ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

case $ac_mode in
:F)
#
# CONFIG_FILE
#

case $INSTALL in
[\\/$] * | ?:[\\/] * ) ac_INSTALL=$INSTALL ;;
*) ac_INSTALL=$ac_top_build_prefix$INSTALL ;;
esac
ac_MKDIR_P=$MKDIR_P
case $MKDIR_P in
[\\/$] * | ?:[\\/] * ) ;;
*/ *) ac_MKDIR_P=$ac_top_build_prefix$MKDIR_P ;;
esac

```

```

# If the template does not know about datarootdir, expand it.
# FIXME: This hack should be removed a few years after 2.60.
ac_datarootdir_hack=; ac_datarootdir_seen=
ac_sed_dataroot='
/datarootdir/ {
    p
    q
}
/@datadir@/p
/@docdir@/p
/@infodir@/p
/@localedir@/p
/@mandir@/p'
case `eval "sed -n \"\$ac_sed_dataroot\" \$ac_file_inputs" ` in
*datarootdir*) ac_datarootdir_seen=yes;;
*@datadir@*|*@docdir@*|*@infodir@*|*@localedir@*|*@mandir@*)
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file_inputs
seems to ignore the --datarootdir setting" >&5
$as_echo "$as_me: WARNING: $ac_file_inputs seems to ignore the --
datarootdir setting" >&2;}
    ac_datarootdir_hack='
s@datadir@&/usr/share&g
s@docdir@&${datarootdir}/doc/${PACKAGE_TARNAME}&g
s@infodir@&/usr/share/info&g
s@localedir@&${datarootdir}/locale&g
s@mandir@&/usr/share/man&g
s\${datarootdir}&${prefix}/share&g' ;;
esac
ac_sed_extra="/^[          ]*VPATH[          ]*=[          ]*/{
h
s///
s/^\:/
s/[          ]*$/:/
s/:\$(srcdir)::/g
s/:\${srcdir}::/g
s/:@srcdir@::/g
s/^\:*//
s/:\:*$//
x
s/\(=[          ]*\)\.*/\1/
G
s/\n//
s/^[^=]*=[          ]*$//
}

:t
/@[a-zA-Z_][a-zA-Z_0-9]*@/!b
s|@configure_input@|${ac_sed_conf_input}|;t t
s&@top_build_dir@&${ac_top_build_dir_sub}&;t t
s&@top_build_prefix@&${ac_top_build_prefix}&;t t
s&@srcdir@&${ac_srcdir}&;t t
s&@abs_srcdir@&${ac_abs_srcdir}&;t t

```

```

s&@top_srcdir@&&$ac_top_srcdir&;t t
s&@abs_top_srcdir@&&$ac_abs_top_srcdir&;t t
s&@builddir@&&$ac_builddir&;t t
s&@abs_builddir@&&$ac_abs_builddir&;t t
s&@abs_top_builddir@&&$ac_abs_top_builddir&;t t
s&@INSTALL@&&$ac_INSTALL&;t t
s&@MKDIR_P@&&$ac_MKDIR_P&;t t
$ac_datarootdir_hack
"
eval sed "\\"$ac_sed_extra\" \"$ac_file_inputs" | $AWK -f
"$ac_tmp/subs.awk" \
  >$ac_tmp/out || as_fn_error $? "could not create $ac_file" "$LINENO"
5

test -z "$ac_datarootdir_hack$ac_datarootdir_seen" &&
{ ac_out=`sed -n '/\${datarootdir}/p' "$ac_tmp/out`; test -n
"$ac_out"; } &&
{ ac_out=`sed -n '/^[ ]*datarootdir[ ]*:=/p' \
  "$ac_tmp/out`; test -z "$ac_out"; } &&
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file contains
a reference to the variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&5
$as_echo "$as_me: WARNING: $ac_file contains a reference to the
variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&2;}

rm -f "$ac_tmp/stdin"
case $ac_file in
-) cat "$ac_tmp/out" && rm -f "$ac_tmp/out";;
*) rm -f "$ac_file" && mv "$ac_tmp/out" "$ac_file";;
esac \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
;;
:H)
#
# CONFIG_HEADER
#
if test x"$ac_file" != x-; then
{
  $as_echo "/* $configure_input */" \
  && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs"
} >"$ac_tmp/config.h" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5
if diff "$ac_file" "$ac_tmp/config.h" >/dev/null 2>&1; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: $ac_file is unchanged"
>&5
$as_echo "$as_me: $ac_file is unchanged" >&6;}
else
rm -f "$ac_file"
mv "$ac_tmp/config.h" "$ac_file" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5
fi

```

```

else
  $sas_echo "/* $configure_input */" \
    && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs" \
    || as_fn_error $? "could not create -" "$LINENO" 5
fi
# Compute "$ac_file"'s index in $config_headers.
_am_arg="$ac_file"
_am_stamp_count=1
for _am_header in $config_headers ;; do
  case $_am_header in
    $_am_arg | $_am_arg:* )
      break ;;
    * )
      _am_stamp_count=`expr $_am_stamp_count + 1` ;;
  esac
done
echo "timestamp for $_am_arg" >`$as_dirname -- "$_am_arg" ||
$as_expr X"$_am_arg" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \| \
  X"$_am_arg" : 'X\(//\)[^/]' \| \
  X"$_am_arg" : 'X\(//\)$' \| \
  X"$_am_arg" : 'X\(//\)' \| . 2>/dev/null ||
$as_echo X"$_am_arg" |
  sed '/^X\(.*[^/]\)\/*[^/][^/]*/*$/{
    s//\1/
    q
  }
/^X\(\\/\)\)[^/].*${
  s//\1/
  q
}
/^X\(\\/\)\)$/{
  s//\1/
  q
}
/^X\(\\/\).*${
  s//\1/
  q
}
s/.*/./; q'`/stamp-h$_am_stamp_count
;;

:C) { $sas_echo "$as_me:${as_lineno-$LINENO}: executing $ac_file
commands" >&5
$as_echo "$as_me: executing $ac_file commands" >&6;}
;;
esac

case $ac_file$ac_mode in
  "depfiles":C) test x"$SAMDEP_TRUE" != x"" || {
# Autoconf 2.62 quotes --file arguments for eval, but not when files

```

```

# are listed without --file.  Let's play safe and only enable the
eval
# if we detect the quoting.
case $CONFIG_FILES in
*\'*) eval set x "$CONFIG_FILES" ;;
*) set x $CONFIG_FILES ;;
esac
shift
for mf
do
# Strip MF so we end up with the name of the file.
mf=`echo "$mf" | sed -e 's/.*$//'`
# Check whether this is an Automake generated Makefile or not.
# We used to match only the files named 'Makefile.in', but
# some people rename them; so instead we look at the file content.
# Grep'ing the first line is not enough: some people post-process
# each Makefile.in and add a new line on top of each file to say
so.
# Grep'ing the whole file is not good either: AIX grep has a line
# limit of 2048, but all sed's we know have understand at least
4000.
if sed -n 's,^#.*generated by automake.*,X,p' "$mf" | grep X
>/dev/null 2>&1; then
dirpart=`$as_dirname -- "$mf" ||
$as_expr X"$mf" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \| \
X"$mf" : 'X\(//\)[^/]' \| \
X"$mf" : 'X\(//\)$' \| \
X"$mf" : 'X\(/\)' \| . 2>/dev/null ||
$as_echo X"$mf" |
sed '/^X\(.*[^/]\)\/*[^/][^/]*/*$/{
s//\1/
q
}
/^X\(\\/\)\)[^/].*/{
s//\1/
q
}
/^X\(\\/\)\)$/{
s//\1/
q
}
/^X\(\\/\).*/{
s//\1/
q
}'
s/.*/./; q'`
else
continue
fi
# Extract the definition of DEPDIR, am__include, and am__quote
# from the Makefile without running 'make'.
DEPDIR=`sed -n 's/^DEPDIR = //p' < "$mf"`

```

```

test -z "$DEPDIR" && continue
am__include=`sed -n 's/^am__include = //p' < "$mf"`
test -z "am__include" && continue
am__quote=`sed -n 's/^am__quote = //p' < "$mf"`
# Find all dependency output files, they are included files with
# $(DEPDIR) in their names. We invoke sed twice because it is the
# simplest approach to changing $(DEPDIR) to its actual value in
the
# expansion.
for file in `sed -n "
s/^$am__include $am__quote\(.*(DEPDIR).*\) $am__quote" '$/\1/p'
<"$mf" | \
sed -e 's/\$(DEPDIR)/"$DEPDIR"/g'; do
# Make sure the directory exists.
test -f "$dirpart/$file" && continue
fdir=`$as_dirname -- "$file" ||
$as_expr X"$file" : 'X\([^/]\)\/*[^/][^/]*/*$' \| \
X"$file" : 'X\(/\)\[^/]' \| \
X"$file" : 'X\(/\)\$' \| \
X"$file" : 'X\(/\) ' \| . 2>/dev/null ||
$as_echo X"$file" |
sed '/^X\([^/]\)\/*[^/][^/]*\/*$/{
s//\1/
q
}
/^X\(\//\)\[^/].*${
s//\1/
q
}
/^X\(\//\)\$/{
s//\1/
q
}
/^X\(\//\).*${
s//\1/
q
}
s/././; q'`
as_dir=$dirpart/$fdir; as_fn_mkdir_p
# echo "creating $dirpart/$file"
echo '# dummy' > "$dirpart/$file"
done
done
}
;;
"libtool":C)

# See if we are running on zsh, and set the options which allow
our
# commands through without removal of \ escapes.
if test -n "${ZSH_VERSION+set}" ; then
setopt NO_GLOB_SUBST

```

```

fi

cfgfile="${ofile}T"
trap "$RM \"\$cfgfile\"; exit 1" 1 2 15
$RM "$cfgfile"

cat <<_LT_EOF >> "$cfgfile"
#! $SHELL

# `ECHO "$ofile" | sed 's%^.*/%%'` - Provide generalized library-
building support services.
# Generated automatically by $as_me ($PACKAGE$TIMESTAMP) $VERSION
# Libtool was configured on host `(hostname || uname -n) 2>/dev/null |
sed lq`:
# NOTE: Changes made to this file will be lost: look at ltmain.sh.
#
# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005,
#           2006, 2007, 2008, 2009, 2010, 2011 Free Software
#           Foundation, Inc.
# Written by Gordon Matzigkeit, 1996
#
# This file is part of GNU Libtool.
#
# GNU Libtool is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of
# the License, or (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html, or
# obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

# The names of the tagged configurations supported by this script.
available_tags=""

# ### BEGIN LIBTOOL CONFIG

# Which release of libtool.m4 was used?

```



```
macro_version=$macro_version
macro_revision=$macro_revision

# Whether or not to build shared libraries.
build_libtool_libs=$enable_shared

# Whether or not to build static libraries.
build_old_libs=$enable_static

# What type of objects to build.
pic_mode=$pic_mode

# Whether or not to optimize for fast installation.
fast_install=$enable_fast_install

# Shell to use when invoking shell scripts.
SHELL=$lt_SHELL

# An echo program that protects backslashes.
ECHO=$lt_ECHO

# The PATH separator for the build system.
PATH_SEPARATOR=$lt_PATH_SEPARATOR

# The host system.
host_alias=$host_alias
host=$host
host_os=$host_os

# The build system.
build_alias=$build_alias
build=$build
build_os=$build_os

# A sed program that does not truncate output.
SED=$lt_SED

# Sed that helps us avoid accidentally triggering echo(1) options like
-n.
Xsed="\$SED -e 1s/^X/"

# A grep program that handles long lines.
GREP=$lt_GREP

# An ERE matcher.
EGREP=$lt_EGREP

# A literal string matcher.
FGREP=$lt_FGREP

# A BSD- or MS-compatible name lister.
NM=$lt_NM
```

```
# Whether we need soft or hard links.
LN_S=$lt_LN_S

# What is the maximum length of a command?
max_cmd_len=$max_cmd_len

# Object file suffix (normally "o").
objext=$ac_objext

# Executable file suffix (normally "").
exeext=$exeext

# whether the shell understands "unset".
lt_unset=$lt_unset

# turn spaces into newlines.
SP2NL=$lt_lt_SP2NL

# turn newlines into spaces.
NL2SP=$lt_lt_NL2SP

# convert \${build} file names to \${host} format.
to_host_file_cmd=$lt_cv_to_host_file_cmd

# convert \${build} files to toolchain format.
to_tool_file_cmd=$lt_cv_to_tool_file_cmd

# An object symbol dumper.
OBJDUMP=$lt_OBJDUMP

# Method to check whether dependent libraries are shared objects.
deplibs_check_method=$lt_deplibs_check_method

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd=$lt_file_magic_cmd

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=$lt_file_magic_glob

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob=$lt_want_nocaseglob

# DLL creation program.
DLLTOOL=$lt_DLLTOOL

# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd=$lt_sharedlib_from_linklib_cmd

# The archiver.
```

```
AR=$lt_AR

# Flags to create an archive.
AR_FLAGS=$lt_AR_FLAGS

# How to feed a file listing to the archiver.
archiver_list_spec=$lt_archiver_list_spec

# A symbol stripping program.
STRIP=$lt_STRIP

# Commands used to install an old-style archive.
RANLIB=$lt_RANLIB
old_postinstall_cmds=$lt_old_postinstall_cmds
old_postuninstall_cmds=$lt_old_postuninstall_cmds

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=$lock_old_archive_extraction

# A C compiler.
LTCC=$lt_CC

# LTCC compiler flags.
LTCFLAGS=$lt_CFLAGS

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe=$lt_lt_cv_sys_global_symbol_pipe

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl=$lt_lt_cv_sys_global_symbol_to_cdecl

# Transform the output of nm in a C name address pair.
global_symbol_to_c_name_address=$lt_lt_cv_sys_global_symbol_to_c_name_
address

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix=$lt_lt_cv_sys_global_symbol
_to_c_name_address_lib_prefix

# Specify filename containing input files for \ $NM.
nm_file_list_spec=$lt_nm_file_list_spec

# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=$lt_sysroot

# The name of the directory that contains temporary libtool files.
objdir=$objdir

# Used to examine libraries when file_magic_cmd begins with "file".
```

```
MAGIC_CMD=$MAGIC_CMD

# Must we lock files when doing compilation?
need_locks=$lt_need_locks

# Manifest tool.
MANIFEST_TOOL=$lt_MANIFEST_TOOL

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
DSYMUTIL=$lt_DSYMUTIL

# Tool to change global to local symbols on Mac OS X.
NMEDIT=$lt_NMEDIT

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=$lt_LIPO

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=$lt_OTOOL

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=$lt_OTOOL64

# Old archive suffix (normally "a").
libext=$libext

# Shared library suffix (normally ".so").
shrext_cmds=$lt_shrext_cmds

# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=$lt_extract_expsyms_cmds

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink=$lt_variables_saved_for_relink

# Do we need the "lib" prefix for modules?
need_lib_prefix=$need_lib_prefix

# Do we need a version for libraries?
need_version=$need_version

# Library versioning type.
version_type=$version_type

# Shared library runtime path variable.
runpath_var=$runpath_var

# Shared library path variable.
shlibpath_var=$shlibpath_var
```

```
# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=$shlibpath_overrides_runpath

# Format of library name prefix.
libname_spec=$lt_libname_spec

# List of archive names.  First name is the real one, the rest are
links.
# The last name is the one that the linker finds with -lNAME
library_names_spec=$lt_library_names_spec

# The coded name of the library, if different from the real name.
soname_spec=$lt_soname_spec

# Permission mode override for installation of shared libraries.
install_override_mode=$lt_install_override_mode

# Command to use after installation of a shared archive.
postinstall_cmds=$lt_postinstall_cmds

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=$lt_postuninstall_cmds

# Commands used to finish a libtool library installation in a
directory.
finish_cmds=$lt_finish_cmds

# As "finish_cmds", except a single script fragment to be evaled but
# not shown.
finish_eval=$lt_finish_eval

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=$hardcode_into_libs

# Compile-time system search path for libraries.
sys_lib_search_path_spec=$lt_sys_lib_search_path_spec

# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec=$lt_sys_lib_dlsearch_path_spec

# Whether dlopen is supported.
dlopen_support=$enable_dlopen

# Whether dlopen of programs is supported.
dlopen_self=$enable_dlopen_self

# Whether dlopen of statically linked programs is supported.
dlopen_self_static=$enable_dlopen_self_static

# Commands to strip libraries.
old_strip_lib=$lt_old_strip_lib
```

```
striplib=$lt_striplib

# The linker used to build libraries.
LD=$lt_LD

# How to create reloadable object files.
reload_flag=$lt_reload_flag
reload_cmds=$lt_reload_cmds

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds

# A language specific compiler.
CC=$lt_compiler

# Is the compiler the GNU compiler?
with_gcc=$GCC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds
```

```
# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds
archive_expsym_cmds=$lt_archive_expsym_cmds

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds
module_expsym_cmds=$lt_module_expsym_cmds

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var
```

```

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action

# ### END LIBTOOL CONFIG

_LT_EOF

case $host_os in
aix3*)
  cat <<\_LT_EOF >> "$cfgfile"
# AIX sometimes has problems with the GCC collect2 program.  For some
# reason, if we set the COLLECT_NAMES environment variable, the
problems
# vanish in a puff of smoke.
if test "X${COLLECT_NAMES+set}" != Xset; then

```



```

COLLECT_NAMES=
export COLLECT_NAMES
fi
_LT_EOF
;;
esac

```

```
ltmain="$ac_aux_dir/ltmain.sh"
```

```

# We use sed instead of cat because bash on DJGPP gets confused if
# if finds mixed CR/LF and LF-only lines.  Since sed operates in
# text mode, it properly converts lines to CR/LF.  This bash problem
# is reportedly fixed, but why not run on old versions too?
sed '$q' "$ltmain" >> "$cfgfile" \
  || (rm -f "$cfgfile"; exit 1)

if test x"$xsi_shell" = xyes; then
  sed -e '/^func_dirname ()$/,/^{ } # func_dirname /c\
func_dirname ()\
{\
 \   case ${1} in\
 \     */*) func_dirname_result="${1%/*}${2}" ;;\
 \     * ) func_dirname_result="${3}" ;;\
 \   esac\
} # Extended-shell func_dirname implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_basename ()$/,/^{ } # func_basename /c\
func_basename ()\
{\
 \   func_basename_result="${1##*/}"\
} # Extended-shell func_basename implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_dirname_and_basename ()$/,/^{ } #
func_dirname_and_basename /c\
func_dirname_and_basename ()\
{\
 \   case ${1} in\
 \     */*) func_dirname_result="${1%/*}${2}" ;;\

```

```

\      * ) func_dirname_result="${3}" ;;\
\      esac\
\      func_basename_result="${1##*/}"\
} # Extended-shell func_dirname_and_basename implementation'
"$cfgfile" > $cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_stripname ()$/,/^\} # func_stripname /c\
func_stripname ()\
{\
\      # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are\
\      # positional parameters, so assign one to ordinary parameter
first.\
\      func_stripname_result=${3}\
\      func_stripname_result=${func_stripname_result#"${1}"}\
\      func_stripname_result=${func_stripname_result%"${2}"}\
} # Extended-shell func_stripname implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_split_long_opt ()$/,/^\} # func_split_long_opt /c\
func_split_long_opt ()\
{\
\      func_split_long_opt_name=${1%%=*}\
\      func_split_long_opt_arg=${1##*}\
} # Extended-shell func_split_long_opt implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_split_short_opt ()$/,/^\} # func_split_short_opt /c\
func_split_short_opt ()\
{\
\      func_split_short_opt_arg=${1#??}\
\      func_split_short_opt_name=${1%"$func_split_short_opt_arg"}\
} # Extended-shell func_split_short_opt implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_lo2o ()$/,/^{ } # func_lo2o /c\
func_lo2o ()\
{\
\
    case ${1} in\
\
        *.lo) func_lo2o_result=${1%.lo}.${objext} ;;\
\
        *)    func_lo2o_result=${1} ;;\
\
    esac\
} # Extended-shell func_lo2o implementation' "$cfgfile" > $cfgfile.tmp
\
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_xform ()$/,/^{ } # func_xform /c\
func_xform ()\
{\
\
    func_xform_result=${1%.*}.lo\
} # Extended-shell func_xform implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_arith ()$/,/^{ } # func_arith /c\
func_arith ()\
{\
\
    func_arith_result=$(( $* ))\
} # Extended-shell func_arith implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_len ()$/,/^{ } # func_len /c\
func_len ()\
{\
\
    func_len_result=${#1}\
} # Extended-shell func_len implementation' "$cfgfile" > $cfgfile.tmp
\
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

fi

if test x"$lt_shell_append" = xyes; then
  sed -e '/^func_append ()$/,/^{ # func_append /c\
func_append ()\
{\
  eval "${1}+=\\${2}"\
} # Extended-shell func_append implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_append_quoted ()$/,/^{ # func_append_quoted /c\
func_append_quoted ()\
{\
  func_quote_for_eval "${2}"\
  eval "${1}+=\\\ \\\$func_quote_for_eval_result"\
} # Extended-shell func_append_quoted implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  # Save a `func_append' function call where possible by direct use of
  '+='
  sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1+= "%g' $cfgfile >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
  test 0 -eq $? || _lt_function_replace_fail=:
else
  # Save a `func_append' function call even when '+' is not available
  sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1=" $\1%g' $cfgfile >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
  test 0 -eq $? || _lt_function_replace_fail=:
fi

if test x"$lt_function_replace_fail" = x:"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: Unable to
substitute extended shell functions in $ofile" >&5
$as_echo "$as_me: WARNING: Unable to substitute extended shell
functions in $ofile" >&2;}
fi

```

```

    mv -f "$cfgfile" "$ofile" ||
      (rm -f "$ofile" && cp "$cfgfile" "$ofile" && rm -f "$cfgfile")
    chmod +x "$ofile"

;;

    esac
done # for ac_tag

as_fn_exit 0

File = config.status.~1~

#!/bin/sh
# Generated by configure.
# Run this file to recreate the current configuration.
# Compiler output produced by configure, useful for debugging
# configure, is in config.log if it exists.

debug=false
ac_cs_recheck=false
ac_cs_silent=false

SHELL=${CONFIG_SHELL-/bin/sh}
export SHELL
## ----- ##
## M4sh Initialization. ##
## ----- ##

# Be more Bourne compatible
DUALCASE=1; export DUALCASE # for MKS sh
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
:
  emulate sh
  NULLCMD=:
  # Pre-4.2 versions of Zsh do word splitting on ${1+"$@"}, which
  # is contrary to our usage.  Disable this feature.
  alias -g '${1+"$@"}'='"$@"'
  setopt NO_GLOB_SUBST
else
  case `(set -o) 2>/dev/null` in #(
    *posix*) :
      set -o posix ;; #(
    *) :
      ;;
  esac
fi

```



```

# IFS
# We need space, tab and new line, in precisely that order. Quoting
is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " "$as_nl"

# Find who we are. Look in the path if we contain no directory
separator.
as_myself=
case $0 in #((
  *[\ \/]* ) as_myself=$0 ;;
  *) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done
IFS=$as_save_IFS

;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
  as_myself=$0
fi
if test ! -f "$as_myself"; then
  $as_echo "$as_myself: error: cannot find myself; rerun with an
absolute file name" >&2
  exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
1"
# suppresses any "Segmentation fault" message there. '((' could
# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH
do eval test x\${$as_var+set} = xset \
  && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

# NLS nuisances.

```

```

LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
# are
# provided, also output the error to LOG_FD, referencing LINENO. Then
# exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
    as_status=$1; test $as_status -eq 0 && as_status=1
    if test "$4"; then
        as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
        $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
    fi
    $as_echo "$as_me: error: $2" >&2
    as_fn_exit $as_status
} # as_fn_error

# as_fn_set_status STATUS
# -----
# Set $? to STATUS, without forking.
as_fn_set_status ()
{
    return $1
} # as_fn_set_status

# as_fn_exit STATUS
# -----
# Exit the shell with STATUS, even in a "trap 0" or "set -e" context.
as_fn_exit ()
{
    set +e
    as_fn_set_status $1
    exit $1
} # as_fn_exit

# as_fn_unset VAR
# -----
# Portably unset VAR.
as_fn_unset ()
{

```



```

    { eval $1=; unset $1;}
}
as_unset=as_fn_unset
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
# VAR. Take
# advantage of any shell optimizations that allow amortized linear
# growth over
# repeated appends, instead of the typical quadratic growth present in
# naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\${as_var} = x12") 2>/dev/null;
then :
    eval 'as_fn_append ()
    {
        eval $1+=\${2}
    }'
else
    as_fn_append ()
    {
        eval $1=\${$1}\${2}
    }
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
# the
# global $as_val. Take advantage of shells that can avoid forks. The
# arguments
# must be portable across $(( )) and expr.
if (eval "test \${(( 1 + 1 ))} = 2") 2>/dev/null; then :
    eval 'as_fn_arith ()
    {
        as_val=${( $* )}
    }'
else
    as_fn_arith ()
    {
        as_val=`expr "$@" || test $? -eq 1`
    }
fi # as_fn_arith

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)'`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

```

```

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
"X/"; then
    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\/\([^\/]*\)/*$' \| \| \
X"$0" : 'X\(/\/\)$$' \| \| \
X"$0" : 'X\(/\/\)\' \| \| . 2>/dev/null ||
$as_echo X/"$0" |
    sed '/^.*\/\([^\/]*\)\/*$/ {
        s//\1/
        q
    }
/^X\/\(\(\/\/\)$$)/ {
        s//\1/
        q
    }
/^X\/\(\(\/\/\)\.*)/ {
        s//\1/
        q
    }
s/.*\/./; q'`

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyz'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

ECHO_C= ECHO_N= ECHO_T=
case `echo -n x` in #((((
-n*))
    case `echo 'xy\c'` in
    *c*) ECHO_T=' ';; # ECHO_T is single tab character.
    xy) ECHO_C='\c';;
    *) echo `echo ksh88 bug on AIX 6.1` > /dev/null
        ECHO_T=' ';;
    esac;;
*)
    ECHO_N='-n';;
esac

```

```

rm -f conf$$ conf$$$.exe conf$$$.file
if test -d conf$$$.dir; then
  rm -f conf$$$.dir/conf$$$.file
else
  rm -f conf$$$.dir
  mkdir conf$$$.dir 2>/dev/null
fi
if (echo >conf$$$.file) 2>/dev/null; then
  if ln -s conf$$$.file conf$$ 2>/dev/null; then
    as_ln_s='ln -s'
    # ... but there are two gotchas:
    # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
    # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
    executable.
    # In both cases, we have to default to `cp -pR'.
    ln -s conf$$$.file conf$$$.dir 2>/dev/null && test ! -f conf$$$.exe
  ||
    as_ln_s='cp -pR'
  elif ln conf$$$.file conf$$ 2>/dev/null; then
    as_ln_s=ln
  else
    as_ln_s='cp -pR'
  fi
else
  as_ln_s='cp -pR'
fi
rm -f conf$$ conf$$$.exe conf$$$.dir/conf$$$.file conf$$$.file
rmdir conf$$$.dir 2>/dev/null

# as_fn_mkdir_p
# -----
# Create "$as_dir" as a directory, including parents if necessary.
as_fn_mkdir_p ()
{
  case $as_dir in #(
  -*) as_dir=./$as_dir;;
  esac
  test -d "$as_dir" || eval $as_mkdir_p || {
    as_dirs=
    while ;; do
      case $as_dir in #(
      *'*) as_qdir=`$as_echo "$as_dir" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'/g"`;;
      #'(
      *) as_qdir=$as_dir;;
      esac
      as_dirs="'$as_qdir' $as_dirs"
      as_dir=`$as_dirname -- "$as_dir" ||
$as_expr X"$as_dir" : 'X\([^/]\)\/*\([^/]\[\^/]\)*/*$' \| \
X"$as_dir" : 'X\(/\/)\[\^/]' \| \

```

```

        X"$sas_dir" : 'X\(//\) '$' \ | \
        X"$sas_dir" : 'X\(/\)' \ | . 2>/dev/null ||
$sas_echo X"$sas_dir" |
    sed '/^X\(. *[^/]\)\ \ \ /* [^/][^/]* \ /* $/{
        s//\1/
        q
    }
/^X\(\ \ \ \) [^/].* $/{
    s//\1/
    q
}
/^X\(\ \ \ \) $/{
    s//\1/
    q
}
/^X\(\ \ \) .* $/{
    s//\1/
    q
}
s/.* /./; q'`
    test -d "$sas_dir" && break
done
    test -z "$sas_dirs" || eval "mkdir $sas_dirs"
} || test -d "$sas_dir" || as_fn_error $? "cannot create directory
$sas_dir"

```

```

} # as_fn_mkdir_p
if mkdir -p . 2>/dev/null; then
    as_mkdir_p='mkdir -p "$sas_dir"'
else
    test -d ./-p && rmdir ./-p
    as_mkdir_p=false
fi

```

```

# as_fn_executable_p FILE
# -----
# Test if FILE is an executable regular file.
as_fn_executable_p ()
{
    test -f "$1" && test -x "$1"
} # as_fn_executable_p
as_test_x='test -x'
as_executable_p=as_fn_executable_p

```

```

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed 'y%*$sas_cr_letters%P$sas_cr_LETTERS%;s%[^_ $sas_cr_alnum]%%_g'"

```

```

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_ $sas_cr_alnum]%%_g'"

```

```

exec 6>&1
## ----- ##
## Main body of $CONFIG_STATUS script. ##
## ----- ##
# Save the log message, to keep $0 and so on meaningful, and to
# report actual input values of CONFIG_FILES etc. instead of their
# values after options handling.
ac_log="
This file was extended by dbus $as_me 1.6.8, which was
generated by GNU Autoconf 2.69.  Invocation command line was

    CONFIG_FILES    = $CONFIG_FILES
    CONFIG_HEADERS  = $CONFIG_HEADERS
    CONFIG_LINKS    = $CONFIG_LINKS
    CONFIG_COMMANDS = $CONFIG_COMMANDS
$ $0 $@

on `(hostname || uname -n) 2>/dev/null | sed 1q`
"

# Files that config.status was made for.
config_files=" Doxyfile dbus/versioninfo.rc dbus/dbus-arch-deps.h
bus/system.conf bus/session.conf bus/messagebus bus/messagebus-config
bus/org.freedesktop.dbus-session.plist bus/rc.messagebus
bus/dbus.service bus/dbus.socket Makefile dbus/Makefile bus/Makefile
tools/Makefile test/Makefile test/name-test/Makefile doc/Makefile
doc/dbus-daemon.1 dbus-1.pc dbus-1-uninstalled.pc test/data/valid-
config-files/debug-allow-all.conf test/data/valid-config-files/debug-
allow-all-sha1.conf test/data/valid-config-files-system/debug-allow-
all-pass.conf test/data/valid-config-files-system/debug-allow-all-
fail.conf test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service

```

```
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service test/data/invalid-
service-files-system/org.freedesktop.DBus.TestSuiteNoUser.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service"
config_headers=" config.h"
config_commands=" depfiles libtool"
```

```
ac_cs_usage="\
\`$as_me' instantiates files and other configuration actions
from templates according to the current configuration. Unless the
files
and actions are specified as TAGs, all are instantiated by default.
```

```
Usage: $0 [OPTION]... [TAG]...
```

```
-h, --help          print this help, then exit
-V, --version       print version number and configuration settings,
then exit
--config           print configuration, then exit
-q, --quiet, --silent
do not print progress messages
-d, --debug        don't remove temporary files
--recheck         update $as_me by reconfiguring in the same
conditions
--file=FILE[:TEMPLATE]
instantiate the configuration file FILE
--header=FILE[:TEMPLATE]
instantiate the configuration header FILE
```

```
Configuration files:
$config_files
```

```
Configuration headers:
$config_headers
```

```
Configuration commands:
$config_commands
```

```
Report bugs to
<https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus>."
```

```
ac_cs_config="--build=i686-linux' '--host=arm-oe-linux-gnueabi' '--
target=arm-oe-linux-gnueabi' '--prefix=/usr' '--exec_prefix=/usr' '--
bindir=/usr/bin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib/dbus' '--
-datadir=/usr/share' '--sysconfdir=/etc' '--sharedstatedir=/com' '--
localstatedir=/var' '--libdir=/usr/lib' '--includedir=/usr/include' '--
-oldincludedir=/usr/include' '--infodir=/usr/share/info' '--
mandir=/usr/share/man' '--disable-silent-rules' '--disable-dependency-
tracking' '--with-libtool-
sysroot=/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1' '--disable-tests' '--disable-
```

```
checks' '--disable-xml-docs' '--disable-doxygen-docs' '--disable-
libaudit' '--with-xml=expat' '--disable-systemd' '--without-
systemdsystemunitdir' '--without-x' '--enable-nls' 'build_alias=i686-
linux' 'host_alias=arm-oe-linux-gnueabi' 'target_alias=arm-oe-linux-
gnueabi' 'CC=arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1' 'CFLAGS=-
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types' 'LDFLAGS=-
L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/lib -Wl,-O1 -Wl,--hash-style=gnu'
'CPPFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/include -fstack-protector'
'CXX=arm-linux-gnueabi-g++ -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1' 'CXXFLAGS=-
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-gcc-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-gcc-4.7-2013.03-20130313_linux/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -fpermissive' 'CPP=arm-linux-
gnueabi-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'"
ac_cs_version="\
dbus config.status 1.6.8
configured by /home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-
external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/configure, generated by GNU
Autoconf 2.69,
  with options \"\$ac_cs_config\"
```

Copyright (C) 2012 Free Software Foundation, Inc.
This config.status script is free software; the Free Software
Foundation
gives unlimited permission to copy, distribute and modify it."

```
ac_pwd='/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-
gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8'
srcdir='.'
INSTALL='/usr/bin/install -c'
MKDIR_P='/bin/mkdir -p'
AWK='gawk'
test -n "$AWK" || AWK=awk
# The default lists apply if the user does not specify any file.
ac_need_defaults=:
while test $# != 0
```

```

do
case $1 in
--*=?*)
    ac_option=`expr "X$1" : 'X\[([^\=]*)\]='`
    ac_optarg=`expr "X$1" : 'X\[^\=]*=\(.*\)`
    ac_shift=:
    ;;
--*=)
    ac_option=`expr "X$1" : 'X\[([^\=]*)\]='`
    ac_optarg=
    ac_shift=:
    ;;
*)
    ac_option=$1
    ac_optarg=$2
    ac_shift=shift
    ;;
esac

case $ac_option in
# Handling of the options.
-recheck | --recheck | --rehec | --reche | --rech | --rec | --re |
--r)
    ac_cs_recheck=: ;;
--version | --versio | --versi | --vers | --ver | --ve | --v | -V )
    $as_echo "$ac_cs_version"; exit ;;
--config | --confi | --conf | --con | --co | --c )
    $as_echo "$ac_cs_config"; exit ;;
--debug | --debu | --deb | --de | --d | -d )
    debug=: ;;
--file | --fil | --fi | --f )
    $ac_shift
    case $ac_optarg in
*'') ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'/g"`
;;
'') as_fn_error $? "missing file argument" ;;
esac
as_fn_append CONFIG_FILES " '$ac_optarg'"
ac_need_defaults=false;;
--header | --heade | --head | --hea )
    $ac_shift
    case $ac_optarg in
*'') ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'/g"`
;;
esac
as_fn_append CONFIG_HEADERS " '$ac_optarg'"
ac_need_defaults=false;;
--he | --h)
    # Conflict between --help and --header
    as_fn_error $? "ambiguous option: \`$1'
Try \`$0 --help' for more information.";;
--help | --hel | -h )

```



```

    $as_echo "$ac_cs_usage"; exit ;;
-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil | --si | --s)
    ac_cs_silent=: ;;

# This is an error.
-*) as_fn_error $? "unrecognized option: \`$1'
Try \`$0 --help' for more information." ;;

*) as_fn_append ac_config_targets " $1"
    ac_need_defaults=false ;;

esac
shift
done

ac_configure_extra_args=

if $ac_cs_silent; then
    exec 6>/dev/null
    ac_configure_extra_args="$ac_configure_extra_args --silent"
fi

if $ac_cs_recheck; then
    set X /bin/sh '/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/configure' '--build=i686-linux' '--host=arm-oe-linux-gnueabi' '--target=arm-oe-linux-gnueabi' '--prefix=/usr' '--exec_prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib/dbus' '--datadir=/usr/share' '--sysconfdir=/etc' '--sharedstatedir=/com' '--localstatedir=/var' '--libdir=/usr/lib' '--includedir=/usr/include' '--oldincludedir=/usr/include' '--infodir=/usr/share/info' '--mandir=/usr/share/man' '--disable-silent-rules' '--disable-dependency-tracking' '--with-libtool-sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' '--disable-tests' '--disable-checks' '--disable-xml-docs' '--disable-doxygen-docs' '--disable-libaudit' '--with-xml=expat' '--disable-systemd' '--without-systemdsystemunitdir' '--without-x' '--enable-nls' 'build_alias=i686-linux' 'host_alias=arm-oe-linux-gnueabi' 'target_alias=arm-oe-linux-gnueabi' 'CC=arm-linux-gnueabihf-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1' 'CFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/include -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types' 'LDFLAGS=-L/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-rpath-link,/home/gangadhar/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/arm-linux-gnueabihf/lib -Wl,-O1 -Wl,--hash-style=gnu'

```

```

'CPPFLAGS=-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector'
'CXX=arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1' 'CXXFLAGS=-
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/include -fstack-protector -O2 -
pipe -g -feliminate-unused-debug-types -fpermissive' 'CPP=arm-linux-
gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8'
$ac_configure_extra_args --no-create --no-recursion
shift
$as_echo "running CONFIG_SHELL=/bin/sh $" >&6
CONFIG_SHELL='/bin/sh'
export CONFIG_SHELL
exec "$@"
fi

exec 5>>config.log
{
echo
sed 'h;s/./-/g;s/^\.../## /;s/...$/ ##/;p;x;p;x' <<_ASBOX
## Running $as_me. ##
_ASBOX
$as_echo "$ac_log"
} >&5

#
# INIT-COMMANDS
#
AMDEP_TRUE="#" ac_aux_dir="."

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

sed_quote_subst='s/\(["`$\\]\)/\\1/g'
double_quote_subst='s/\(["`\\]\)/\\1/g'
delay_variable_subst='s/\\\\\\\\\\\\\\\\$\\\\\\\\\\\\\\\\$/g'
macro_version='2.4.2'
macro_revision='1.3337'
enable_shared='yes'
enable_static='yes'
pic_mode='default'
enable_fast_install='yes'
SHELL='/bin/sh'
ECHO='printf %s\n'
PATH_SEPARATOR=':'

```

```
host_alias='arm-oe-linux-gnueabi'
host='arm-oe-linux-gnueabi'
host_os='linux-gnueabi'
build_alias='i686-linux'
build='i686-pc-linux-gnu'
build_os='linux-gnu'
SED='/bin/sed'
Xsed='/bin/sed -e 1s/^X//'
GREP='/bin/grep'
EGREP='/bin/grep -E'
FGREP='/bin/grep -F'
LD='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
NM='arm-linux-gnueabi-hf-nm'
LN_S='ln -s'
max_cmd_len='1572864'
ac_objext='o'
exeext=''
lt_unset='unset'
lt_SP2NL='tr \040 \012'
lt_NL2SP='tr \015\012 \040\040'
lt_cv_to_host_file_cmd='func_convert_file_noop'
lt_cv_to_tool_file_cmd='func_convert_file_noop'
reload_flag=' -r'
reload_cmds='$LD$reload_flag -o $output$reload_objs'
OBJDUMP='arm-linux-gnueabi-hf-objdump'
deplibs_check_method='pass_all'
file_magic_cmd='$MAGIC_CMD'
file_magic_glob=''
want_nocaseglob='no'
DLLTOOL='false'
sharedlib_from_linklib_cmd='printf %s\n'
AR='arm-linux-gnueabi-hf-ar'
AR_FLAGS='cru'
archiver_list_spec='@'
STRIP='arm-linux-gnueabi-hf-strip'
RANLIB='arm-linux-gnueabi-hf-ranlib'
old_postinstall_cmds='chmod 644 $oldlib~$RANLIB $tool_oldlib'
old_postuninstall_cmds=''
old_archive_cmds='$AR $AR_FLAGS $oldlib$oldobjs~$RANLIB $tool_oldlib'
lock_old_archive_extraction='no'
CC='arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-interwork -
mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
CFLAGS=' -Wall -Wextra -Wchar-subscripts -Wmissing-declarations -
Wmissing-prototypes -Wnested-externs -Wpointer-arith -Wcast-align -
Wno-address -Wfloat-equal -Wdeclaration-after-statement -Wno-unused-
label -Wno-missing-field-initializers -Wno-unused-parameter -Wno-sign-
compare -Wno-pointer-sign -Wno-type-limits -fno-common -fno-strict-
aliasing -isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
```

```

2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types'
compiler=''
GCC='yes'
lt_cv_sys_global_symbol_pipe='sed -n -e '\''s/^.*[
]\([ABCDGIRSTW][ABCDGIRSTW]*\)[ ]*\([_A-Za-z][_A-Za-z0-
9]*\)\$/\1 \2 \2/p'\'' | sed '\''/_gnu_lto/d'\''
lt_cv_sys_global_symbol_to_cdecl='sed -n -e '\''s/^T .* \(.*)\$/extern
int \1();/p'\'' -e '\''s/^[ABCDGIRSTW]* .* \(.*)\$/extern char
\1;/p'\''
lt_cv_sys_global_symbol_to_c_name_address='sed -n -e '\''s/^: \([
]*\)\[ ]*\$/ {"\1", (void *) 0},/p'\'' -e '\''s/^[ABCDGIRSTW]* \([
]*\)\ \([
]*\)\$/ {"\2", (void *) \&\2},/p'\''
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix='sed -n -e
'\''s/^: \([
]*\)\[ ]*\$/ {"\1", (void *) 0},/p'\'' -e
'\''s/^[ABCDGIRSTW]* \([
]*\)\ \([lib[
]*\)\$/ {"\2", (void *)
\&\2},/p'\'' -e '\''s/^[ABCDGIRSTW]* \([
]*\)\ \([
]*\)\$/ {"lib\2",
(void *) \&\2},/p'\''
nm_file_list_spec='@'
lt_sysroot='/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-
external-linaro-toolchain/sysroots/delphi-vp1'
objdir='.libs'
MAGIC_CMD='file'
lt_prog_compiler_no_builtin_flag=' -fno-builtin'
lt_prog_compiler_pic=' -fPIC -DPIC'
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_static='-static'
lt_cv_prog_compiler_c_o='yes'
need_locks='no'
MANIFEST_TOOL=':'
DSYMUTIL=''
NMEDIT=''
LIPO=''
OTOOL=''
OTOOL64=''
libext='a'
shrext_cmds='.so'
extract_expsyms_cmds=''
archive_cmds_need_lc='no'
enable_shared_with_static_runtimes='no'
export_dynamic_flag_spec='${wl}--export-dynamic'
whole_archive_flag_spec='${wl}--whole-archive$convenience ${wl}--no-
whole-archive'
compiler_needs_object='no'
old_archive_from_new_cmds=''
old_archive_from_expsyms_cmds=''
archive_cmds='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
archive_expsym_cmds='echo "{ global:" > $output_objdir/$libname.ver~
cat $export_symbols | sed -e "s/\(.*\)/\1/" >>
$output_objdir/$libname.ver~
echo "local: *; };" >> $output_objdir/$libname.ver~

```

```

        $CC -shared $pic_flag $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname ${wl}-version-script
${wl}$output_objdir/$libname.ver -o $lib'
module_cmds=''
module_expsym_cmds=''
with_gnu_ld='yes'
allow_undefined_flag=''
no_undefined_flag=''
hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
hardcode_libdir_separator=''
hardcode_direct='no'
hardcode_direct_absolute='no'
hardcode_minus_L='no'
hardcode_shlibpath_var='unsupported'
hardcode_automatic='no'
inherit_rpath='no'
link_all_deplibs='unknown'
always_export_symbols='no'
export_symbols_cmds='$NM $libobjs $convenience | $global_symbol_pipe |
$SED '\''s/.* //'\' | sort | uniq > $export_symbols'
exclude_expsyms='_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*'
include_expsyms=''
prelink_cmds=''
postlink_cmds=''
file_list_spec=''
variables_saved_for_relink='PATH LD_LIBRARY_PATH LD_RUN_PATH
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH'
need_lib_prefix='no'
need_version='no'
version_type='linux'
runpath_var='LD_RUN_PATH'
shlibpath_var='LD_LIBRARY_PATH'
shlibpath_overrides_runpath='no'
libname_spec='lib$name'
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
install_override_mode=''
postinstall_cmds=''
postuninstall_cmds=''
finish_cmds='PATH="\$PATH:/sbin" ldconfig -n $libdir'
finish_eval=''
hardcode_into_libs='yes'
sys_lib_search_path_spec='/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabi-4.7.3
/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/lib/gcc /home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/lib/gcc/arm-linux-gnueabi-
/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/arm-linux-gnueabi/lib
/home/gangadhar/newyoctobuild/tisd/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib

```

```
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib '
sys_lib_dlsearch_path_spec='/usr/lib /lib'
hardcode_action='immediate'
enable_dlopen='unknown'
enable_dlopen_self='unknown'
enable_dlopen_self_static='unknown'
old_striplib='arm-linux-gnueabi-hf-strip --strip-debug'
striplib='arm-linux-gnueabi-hf-strip --strip-unneeded'
compiler_lib_search_dirs=''
predep_objects=''
postdep_objects=''
predeps=''
postdeps=''
compiler_lib_search_path=''
LD_CXX='arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
LD_RC=''
reload_flag_CXX=' -r'
reload_flag_RC=''
reload_cmds_CXX='$LD$reload_flag -o $output$reload_objs'
reload_cmds_RC=''
old_archive_cmds_CXX='$AR $AR_FLAGS $oldlib$oldobjs~$RANLIB
$tool_oldlib'
old_archive_cmds_RC=''
compiler_CXX='arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
compiler_RC=''
GCC_CXX='yes'
GCC_RC=''
lt_prog_compiler_no_builtin_flag_CXX=' -fno-builtin'
lt_prog_compiler_no_builtin_flag_RC=''
lt_prog_compiler_pic_CXX=' -fPIC -DPIC'
lt_prog_compiler_pic_RC=''
lt_prog_compiler_wl_CXX=' -Wl,'
lt_prog_compiler_wl_RC=''
lt_prog_compiler_static_CXX=' -static'
lt_prog_compiler_static_RC=''
lt_cv_prog_compiler_c_o_CXX='yes'
lt_cv_prog_compiler_c_o_RC='yes'
archive_cmds_need_lc_CXX='no'
archive_cmds_need_lc_RC=''
enable_shared_with_static_runtimes_CXX='no'
enable_shared_with_static_runtimes_RC=''
export_dynamic_flag_spec_CXX='${wl}--export-dynamic'
export_dynamic_flag_spec_RC=''
whole_archive_flag_spec_CXX='${wl}--whole-archive$convenience ${wl}--
no-whole-archive'
whole_archive_flag_spec_RC=''
```

```

compiler_needs_object_CXX='no'
compiler_needs_object_RC=''
old_archive_from_new_cmds_CXX=''
old_archive_from_new_cmds_RC=''
old_archive_from_expsyms_cmds_CXX=''
old_archive_from_expsyms_cmds_RC=''
archive_cmds_CXX='$CC $pic_flag -shared -nostdlib $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags ${wl}-soname
$wl$soname -o $lib'
archive_cmds_RC=''
archive_expsym_cmds_CXX='$CC $pic_flag -shared -nostdlib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname $wl$soname ${wl}-retain-symbols-file $wl$export_symbols -
o $lib'
archive_expsym_cmds_RC=''
module_cmds_CXX=''
module_cmds_RC=''
module_expsym_cmds_CXX=''
module_expsym_cmds_RC=''
with_gnu_ld_CXX='yes'
with_gnu_ld_RC=''
allow_undefined_flag_CXX=''
allow_undefined_flag_RC=''
no_undefined_flag_CXX=''
no_undefined_flag_RC=''
hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
hardcode_libdir_flag_spec_RC=''
hardcode_libdir_separator_CXX=''
hardcode_libdir_separator_RC=''
hardcode_direct_CXX='no'
hardcode_direct_RC=''
hardcode_direct_absolute_CXX='no'
hardcode_direct_absolute_RC=''
hardcode_minus_L_CXX='no'
hardcode_minus_L_RC=''
hardcode_shlibpath_var_CXX='unsupported'
hardcode_shlibpath_var_RC=''
hardcode_automatic_CXX='no'
hardcode_automatic_RC=''
inherit_rpath_CXX='no'
inherit_rpath_RC=''
link_all_deplibs_CXX='unknown'
link_all_deplibs_RC=''
always_export_symbols_CXX='no'
always_export_symbols_RC=''
export_symbols_cmds_CXX='$NM $libobjs $convenience |
$global_symbol_pipe | $SED '\''s/.* //'\' | sort | uniq >
$export_symbols'
export_symbols_cmds_RC=''
exclude_expsyms_CXX='_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*'
exclude_expsyms_RC=''
include_expsyms_CXX=''

```

```
include_expsyms_RC=''
prelink_cmds_CXX=''
prelink_cmds_RC=''
postlink_cmds_CXX=''
postlink_cmds_RC=''
file_list_spec_CXX=''
file_list_spec_RC=''
hardcode_action_CXX='immediate'
hardcode_action_RC=''
compiler_lib_search_dirs_CXX='/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7.3 /home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/../lib/gcc /home/gangadhar/gcc-linaro-arm-
linux-gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7.3/../../../../arm-linux-gnueabi/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib'
compiler_lib_search_dirs_RC=''
predep_objects_CXX='/home/gangadhar/newyoctobuild/tisdk/build/arago-
tmp-external-linaro-toolchain/sysroots/delphi-vp1/usr/lib/crti.o
/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-
20130313_linux/bin/../lib/gcc/arm-linux-gnueabi-4.7.3/crtbeginS.o'
predep_objects_RC=''
postdep_objects_CXX='/home/gangadhar/gcc-linaro-arm-linux-gnueabi-
4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7.3/crtendS.o
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib/crtn.o'
postdep_objects_RC=''
predeps_CXX=''
predeps_RC=''
postdeps_CXX='-lstdc++ -lm -lgcc_s -lc -lgcc_s'
postdeps_RC=''
compiler_lib_search_path_CXX='-L/home/gangadhar/gcc-linaro-arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7.3 -L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/bin/../lib/gcc -L/home/gangadhar/gcc-linaro-
arm-linux-gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-
linux-gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7-2013.03-20130313_linux/bin/../lib/gcc/arm-linux-
gnueabi-4.7.3/../../../../arm-linux-gnueabi/lib -
L/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/lib -
L/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/sysroots/delphi-vp1/usr/lib'
compiler_lib_search_path_RC=''
```



```

LTCC='arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork
-mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'
LTCFLAGS='-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-4.7-
2013.03-20130313_linux/arm-linux-gnueabi/include -fstack-protector
-O2 -pipe -g -feliminate-unused-debug-types'
compiler='arm-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1'

```

```

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
    eval 'cat <<_LTECHO_EOF'
$1
_LTECHO_EOF'
}

```

```

# Quote evaled strings.
for var in SHELL ECHO PATH_SEPARATOR SED GREP EGREP FGREP LD NM LN_S
lt_SP2NL lt_NL2SP reload_flag OBJDUMP deplibs_check_method
file_magic_cmd file_magic_glob want_nocaseglob DLLTOOL
sharedlib_from_linklib_cmd AR AR_FLAGS archiver_list_spec STRIP RANLIB
CC CFLAGS compiler lt_cv_sys_global_symbol_pipe
lt_cv_sys_global_symbol_to_cdecl
lt_cv_sys_global_symbol_to_c_name_address
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix nm_file_list_spec
lt_prog_compiler_no_builtin_flag lt_prog_compiler_pic
lt_prog_compiler_wl lt_prog_compiler_static lt_cv_prog_compiler_c_o
need_locks MANIFEST_TOOL DSYMUTIL NMEDIT LIPO OTOOL OTOOL64
shrext_cmds export_dynamic_flag_spec whole_archive_flag_spec
compiler_needs_object with_gnu_ld allow_undefined_flag
no_undefined_flag hardcode_libdir_flag_spec hardcode_libdir_separator
exclude_expsyms include_expsyms file_list_spec
variables_saved_for_relink libname_spec library_names_spec soname_spec
install_override_mode finish_eval old_strip lib striplib
compiler_lib_search_dirs predep_objects postdep_objects predeps
postdeps compiler_lib_search_path LD_CXX LD_RC reload_flag_CXX
reload_flag_RC compiler_CXX compiler_RC
lt_prog_compiler_no_builtin_flag_CXX
lt_prog_compiler_no_builtin_flag_RC lt_prog_compiler_pic_CXX
lt_prog_compiler_pic_RC lt_prog_compiler_wl_CXX lt_prog_compiler_wl_RC
lt_prog_compiler_static_CXX lt_prog_compiler_static_RC
lt_cv_prog_compiler_c_o_CXX lt_cv_prog_compiler_c_o_RC
export_dynamic_flag_spec_CXX export_dynamic_flag_spec_RC
whole_archive_flag_spec_CXX whole_archive_flag_spec_RC
compiler_needs_object_CXX compiler_needs_object_RC with_gnu_ld_CXX
with_gnu_ld_RC allow_undefined_flag_CXX allow_undefined_flag_RC
no_undefined_flag_CXX no_undefined_flag_RC
hardcode_libdir_flag_spec_CXX hardcode_libdir_flag_spec_RC

```

```

hardcode_libdir_separator_CXX hardcode_libdir_separator_RC
exclude_expsyms_CXX exclude_expsyms_RC include_expsyms_CXX
include_expsyms_RC file_list_spec_CXX file_list_spec_RC
compiler_lib_search_dirs_CXX compiler_lib_search_dirs_RC
predep_objects_CXX predep_objects_RC postdep_objects_CXX
postdep_objects_RC predeps_CXX predeps_RC postdeps_CXX postdeps_RC
compiler_lib_search_path_CXX compiler_lib_search_path_RC; do
    case `eval \\$ECHO \\`"\\$var"\\` in
        *[\`"\\$]*)
            eval "lt_$var=\\`"\\$ECHO \\`"\\$var\\" | \\$SED
\\`"\\$sed_quote_subst"\\`"\\`"\\`"
            ;;
        *)
            eval "lt_$var=\\`"\\$var\\`"\\`"
            ;;
    esac
done

# Double-quote double-eval'd strings.
for var in reload_cmds old_postinstall_cmds old_postuninstall_cmds
old_archive_cmds extract_expsyms_cmds old_archive_from_new_cmds
old_archive_from_expsyms_cmds archive_cmds archive_expsym_cmds
module_cmds module_expsym_cmds export_symbols_cmds prelink_cmds
postlink_cmds postinstall_cmds postuninstall_cmds finish_cmds
sys_lib_search_path_spec sys_lib_dlsearch_path_spec reload_cmds_CXX
reload_cmds_RC old_archive_cmds_CXX old_archive_cmds_RC
old_archive_from_new_cmds_CXX old_archive_from_new_cmds_RC
old_archive_from_expsyms_cmds_CXX old_archive_from_expsyms_cmds_RC
archive_cmds_CXX archive_cmds_RC archive_expsym_cmds_CXX
archive_expsym_cmds_RC module_cmds_CXX module_cmds_RC
module_expsym_cmds_CXX module_expsym_cmds_RC export_symbols_cmds_CXX
export_symbols_cmds_RC prelink_cmds_CXX prelink_cmds_RC
postlink_cmds_CXX postlink_cmds_RC; do
    case `eval \\$ECHO \\`"\\$var"\\` in
        *[\`"\\$]*)
            eval "lt_$var=\\`"\\$ECHO \\`"\\$var\\" | \\$SED -e
\\`"\\$double_quote_subst" -e \\`"\\$sed_quote_subst" -e
\\`"\\$delay_variable_subst"\\`"\\`"\\`"
            ;;
        *)
            eval "lt_$var=\\`"\\$var\\`"\\`"
            ;;
    esac
done

ac_aux_dir='.'
xsi_shell='yes'
lt_shell_append='yes'

# See if we are running on zsh, and set the options which allow our
# commands through without removal of \ escapes INIT.
if test -n "${ZSH_VERSION+set}" ; then

```

```

setopt NO_GLOB_SUBST
fi

PACKAGE='dbus'
VERSION='1.6.8'
TIMESTAMP=''
RM='rm -f'
ofile='arm-oe-linux-gnueabi-libtool'

# Handling of arguments.
for ac_config_target in $ac_config_targets
do
  case $ac_config_target in
    "config.h") CONFIG_HEADERS="$CONFIG_HEADERS config.h" ;;
    "depfiles") CONFIG_COMMANDS="$CONFIG_COMMANDS depfiles" ;;
    "libtool") CONFIG_COMMANDS="$CONFIG_COMMANDS libtool" ;;
    "Doxyfile") CONFIG_FILES="$CONFIG_FILES Doxyfile" ;;
    "dbus/versioninfo.rc") CONFIG_FILES="$CONFIG_FILES
dbus/versioninfo.rc" ;;
    "dbus/dbus-arch-deps.h") CONFIG_FILES="$CONFIG_FILES dbus/dbus-
arch-deps.h" ;;
    "bus/system.conf") CONFIG_FILES="$CONFIG_FILES bus/system.conf" ;;
    "bus/session.conf") CONFIG_FILES="$CONFIG_FILES bus/session.conf"
;;
    "bus/messagebus") CONFIG_FILES="$CONFIG_FILES bus/messagebus" ;;
    "bus/messagebus-config") CONFIG_FILES="$CONFIG_FILES
bus/messagebus-config" ;;
    "bus/org.freedesktop.dbus-session.plist")
CONFIG_FILES="$CONFIG_FILES bus/org.freedesktop.dbus-session.plist" ;;
    "bus/rc.messagebus") CONFIG_FILES="$CONFIG_FILES
bus/rc.messagebus" ;;
    "bus/dbus.service") CONFIG_FILES="$CONFIG_FILES bus/dbus.service"
;;
    "bus/dbus.socket") CONFIG_FILES="$CONFIG_FILES bus/dbus.socket" ;;
    "Makefile") CONFIG_FILES="$CONFIG_FILES Makefile" ;;
    "dbus/Makefile") CONFIG_FILES="$CONFIG_FILES dbus/Makefile" ;;
    "bus/Makefile") CONFIG_FILES="$CONFIG_FILES bus/Makefile" ;;
    "tools/Makefile") CONFIG_FILES="$CONFIG_FILES tools/Makefile" ;;
    "test/Makefile") CONFIG_FILES="$CONFIG_FILES test/Makefile" ;;
    "test/name-test/Makefile") CONFIG_FILES="$CONFIG_FILES test/name-
test/Makefile" ;;
    "doc/Makefile") CONFIG_FILES="$CONFIG_FILES doc/Makefile" ;;

```

```
"doc/dbus-daemon.1") CONFIG_FILES="$CONFIG_FILES doc/dbus-
daemon.1" ;;
"dbus-1.pc") CONFIG_FILES="$CONFIG_FILES dbus-1.pc" ;;
"dbus-1-uninstalled.pc") CONFIG_FILES="$CONFIG_FILES dbus-1-
uninstalled.pc" ;;
"test/data/valid-config-files/debug-allow-all.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files/debug-allow-
all.conf" ;;
"test/data/valid-config-files/debug-allow-all-sha1.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files/debug-allow-
all-sha1.conf" ;;
"test/data/valid-config-files-system/debug-allow-all-pass.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files-system/debug-
allow-all-pass.conf" ;;
"test/data/valid-config-files-system/debug-allow-all-fail.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files-system/debug-
allow-all-fail.conf" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service"
;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service")
```

```

CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service"
;;
    "test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service" ;;
    "test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service")
CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service" ;;
    "test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service")
CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service" ;;
    "test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service" ;;

    *) as_fn_error $? "invalid argument: \`${ac_config_target}'" "$LINENO"
5;;
    esac
done

# If the user did not use the arguments to specify the items to
# instantiate,
# then the envvar interface is used.  Set only those that are not.
# We use the long form for the default assignment because of an
# extremely
# bizarre bug on SunOS 4.1.3.
if $ac_need_defaults; then
    test "${CONFIG_FILES+set}" = set || CONFIG_FILES=$config_files
    test "${CONFIG_HEADERS+set}" = set || CONFIG_HEADERS=$config_headers
    test "${CONFIG_COMMANDS+set}" = set ||
CONFIG_COMMANDS=$config_commands
fi

# Have a temporary directory for convenience.  Make it in the build
# tree
# simply because there is no reason against having it here, and in
# addition,
# creating and moving files from /tmp can sometimes cause problems.
# Hook for its removal unless debugging.
# Note that there is a small window in which the directory will not be
# cleaned:
# after its creation but before its name has been assigned to `$tmp'.
$debug ||
{
    tmp= ac_tmp=
    trap 'exit_status=$?'

```

```

: "${ac_tmp:=${tmp}}"
{ test ! -d "$ac_tmp" || rm -fr "$ac_tmp"; } && exit $exit_status
' 0
trap 'as_fn_exit 1' 1 2 13 15
}
# Create a (secure) tmp directory for tmp files.

{
tmp=`(umask 077 && mktemp -d "./confXXXXXX") 2>/dev/null` &&
test -d "$tmp"
} ||
{
tmp=./conf$$-$RANDOM
(umask 077 && mkdir "$tmp")
} || as_fn_error $? "cannot create a temporary directory in ."
"$LINENO" 5
ac_tmp=$tmp

# Set up the scripts for CONFIG_FILES section.
# No need to generate them if there are no CONFIG_FILES.
# This happens for instance with `./config.status config.h'.
if test -n "$CONFIG_FILES"; then

ac_cr=`echo X | tr X '\015'`
# On cygwin, bash can eat \r inside `` if the user requested igncr.
# But we know of no other shell where ac_cr would be empty at this
# point, so we can use a bashism as a fallback.
if test "x$ac_cr" = x; then
eval ac_cr=\$\'\r\'
fi
ac_cs_awk_cr=`$AWK 'BEGIN { print "a\rb" }' </dev/null 2>/dev/null`
if test "$ac_cs_awk_cr" = "a${ac_cr}b"; then
ac_cs_awk_cr='\r'
else
ac_cs_awk_cr=$ac_cr
fi

echo 'BEGIN {' >"$ac_tmp/subs1.awk" &&
cat >>"$ac_tmp/subs1.awk" <<\_ACAWK &&
S["am__EXEEXT_FALSE"]=""
S["am__EXEEXT_TRUE"]="#"
S["LTLIBOBS"]=""
S["LIBOBS"]=""
S["DBUS_SESSION_BUS_DEFAULT_ADDRESS"]="unix:tmpdir=/tmp"
S["DBUS_SESSION_SOCKET_DIR"]="/tmp"
S["TEST_LISTEN"]="unix:tmpdir=/tmp"
S["TEST_SOCKET_DIR"]="/tmp"
S["TEST_LAUNCH_HELPER_BINARY"]="/home/gangadhar/newyoctobuild/tisd/bu
ild/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-
oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/d"\
"bus-1.6.8/bus/dbus-daemon-launch-helper-test"

```

```
S["TEST_BUS_BINARY"]=""  
S["DBUS_TEST_EXEC"]="/home/gangadhar/newyoctobuild/tisdk/build/arago-  
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-  
gnueabi/dbus/1.6.8-r6.0-arago1/d\  
"bus-1.6.8/test"  
S["DBUS_TEST_DATA"]="/home/gangadhar/newyoctobuild/tisdk/build/arago-  
tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-  
gnueabi/dbus/1.6.8-r6.0-arago1/d\  
"bus-1.6.8/test/data"  
S["DBUS_LIBEXECDIR"]="/usr/lib/dbus"  
S["DBUS_BINDIR"]="/usr/bin"  
S["DBUS_DAEMONDIR"]="/usr/bin"  
S["DBUS_DATADIR"]="/usr/share"  
S["DBUS_PREFIX"]="/usr"  
S["DBUS_USER"]="messagebus"  
S["DBUS_CONSOLE_OWNER_FILE"]=""  
S["DBUS_CONSOLE_AUTH_DIR"]="/var/run/console/"  
S["DBUS_SYSTEM_PID_FILE"]="/var/run/dbus/pid"  
S["DBUS_SYSTEM_BUS_DEFAULT_ADDRESS"]="unix:path=/var/run/dbus/system_b  
us_socket"  
S["DBUS_SYSTEM_SOCKET"]="/var/run/dbus/system_bus_socket"  
S["HAVE_SYSTEMD_FALSE"]=""  
S["HAVE_SYSTEMD_TRUE"]="#"  
S["systemdsystemunitdir"]=""  
S["DBUS_INIT_SCRIPTS_CYGWIN_FALSE"]=""  
S["DBUS_INIT_SCRIPTS_CYGWIN_TRUE"]="#"  
S["DBUS_INIT_SCRIPTS_SLACKWARE_FALSE"]=""  
S["DBUS_INIT_SCRIPTS_SLACKWARE_TRUE"]="#"  
S["DBUS_INIT_SCRIPTS_RED_HAT_FALSE"]=""  
S["DBUS_INIT_SCRIPTS_RED_HAT_TRUE"]="#"  
S["EXPANDED_DATADIR"]="/usr/share"  
S["EXPANDED_LIBEXECDIR"]="/usr/lib/dbus"  
S["EXPANDED_LIBDIR"]="/usr/lib"  
S["EXPANDED_BINDIR"]="/usr/bin"  
S["EXPANDED_SYSCONFDIR"]="/etc"  
S["EXPANDED_LOCALSTATEDIR"]="/var"  
S["EXPANDED_PREFIX"]="/usr"  
S["DBUS_CAN_UPLOAD_DOCS_FALSE"]=""  
S["DBUS_CAN_UPLOAD_DOCS_TRUE"]="#"  
S["DBUS_HAVE_MAN2HTML_FALSE"]=""  
S["DBUS_HAVE_MAN2HTML_TRUE"]="#"  
S["MAN2HTML"]=""  
S["DBUS_XML_DOCS_ENABLED_FALSE"]=""  
S["DBUS_XML_DOCS_ENABLED_TRUE"]="#"  
S["XMLTO"]="no"  
S["DBUS_HAVE_XSLTPROC_FALSE"]="#"  
S["DBUS_HAVE_XSLTPROC_TRUE"]=""  
S["XSLTPROC"]="xsltproc"  
S["DBUS_DOXYGEN_DOCS_ENABLED_FALSE"]=""  
S["DBUS_DOXYGEN_DOCS_ENABLED_TRUE"]="#"  
S["DOXYGEN"]="/usr/bin/doxygen"  
S["DBUS_X_LIBS"]=""
```

```
S["DBUS_X_CFLAGS"]=""  
S["X_EXTRA_LIBS"]=""  
S["X_LIBS"]=""  
S["X_PRE_LIBS"]=""  
S["X_CFLAGS"]=""  
S["LIBDBUS_LIBS"]="-lpthread -lrt "  
S["VALGRIND_LIBS"]=""  
S["VALGRIND_CFLAGS"]=""  
S["NETWORK_libs"]=""  
S["ADT_LIBS"]=""  
S["SELINUX_LIBS"]=""  
S["HAVE_LIBAUDIT_FALSE"]=""  
S["HAVE_LIBAUDIT_TRUE"]="#" "  
S["SYSTEMD_LIBS"]=""  
S["SYSTEMD_CFLAGS"]=""  
S["HAVE_CONSOLE_OWNER_FILE_FALSE"]=""  
S["HAVE_CONSOLE_OWNER_FILE_TRUE"]="#" "  
S["LAUNCHD_AGENT_DIR"]="/Library/LaunchAgents"  
S["DBUS_ENABLE_LAUNCHD_FALSE"]=""  
S["DBUS_ENABLE_LAUNCHD_TRUE"]="#" "  
S["LAUNCHCTL"]=""  
S["DBUS_BUS_ENABLE_KQUEUE_FALSE"]=""  
S["DBUS_BUS_ENABLE_KQUEUE_TRUE"]="#" "  
S["HAVE_LINUX_EPOLL_FALSE"]="#" "  
S["HAVE_LINUX_EPOLL_TRUE"]=""  
S["DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE"]=""  
S["DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE"]="#" "  
S["DBUS_BUS_ENABLE_INOTIFY_FALSE"]="#" "  
S["DBUS_BUS_ENABLE_INOTIFY_TRUE"]=""  
S["HAVE_SELINUX_FALSE"]=""  
S["HAVE_SELINUX_TRUE"]="#" "  
S["THREAD_LIBS"]="-lpthread -lrt"  
S["XML_LIBS"]="-lexpat"  
S["XML_CFLAGS"]=""  
S["DBUS_USE_LIBXML_FALSE"]=""  
S["DBUS_USE_LIBXML_TRUE"]="#" "  
S["DBUS_USE_EXPAT_FALSE"]="#" "  
S["DBUS_USE_EXPAT_TRUE"]=""  
S["LIBXML_LIBS"]=""  
S["LIBXML_CFLAGS"]=""  
S["DBUS_PATH_OR_ABSTRACT"]="abstract"  
S["DBUS_INT16_TYPE"]="short"  
S["DBUS_INT32_TYPE"]="int"  
S["DBUS_HAVE_INT64"]="1"  
S["DBUS_UINT64_CONSTANT"]="(val##ULL) "  
S["DBUS_INT64_CONSTANT"]="(val##LL) "  
S["DBUS_INT64_TYPE"]="long long"  
S["R_DYNAMIC_LDFLAG"]=""  
S["pkgpyexecdir"]="$ {pyexecdir}/dbus"  
S["pyexecdir"]="$ {exec_prefix}/lib/python2.7/dist-packages"  
S["pkgpythondir"]="$ {pythondir}/dbus"  
S["pythondir"]="$ {prefix}/lib/python2.7/dist-packages"
```



```
S["PYTHON_PLATFORM"]="linux2"
S["PYTHON_EXEC_PREFIX"]="${exec_prefix}"
S["PYTHON_LIB_PREFIX"]="${libdir}"
S["PYTHON_PREFIX"]="${prefix}"
S["PYTHON_VERSION"]="2.7"
S["PYTHON"]="/usr/bin/python"
S["DBUS_ENABLE_INSTALLED_TESTS_FALSE"]=""
S["DBUS_ENABLE_INSTALLED_TESTS_TRUE"]="#"
S["DBUS_WITH_GLIB_FALSE"]="#"
S["DBUS_WITH_GLIB_TRUE"]=""
S["DBUS_ENABLE_MODULAR_TESTS_FALSE"]=""
S["DBUS_ENABLE_MODULAR_TESTS_TRUE"]="#"
S["DBUS_GLIB_LIBS"]=""
S["DBUS_GLIB_CFLAGS"]=""
S["GLIB_LIBS"]=""
S["GLIB_CFLAGS"]=""
S["DBUS_ENABLE_EMBEDDED_TESTS_FALSE"]=""
S["DBUS_ENABLE_EMBEDDED_TESTS_TRUE"]="#"
S["DBUS_BUILD_TESTS_FALSE"]=""
S["DBUS_BUILD_TESTS_TRUE"]="#"
S["DBUS_STATIC_BUILD_CPPFLAGS"]=""
S["DBUS_CYGWIN_FALSE"]=""
S["DBUS_CYGWIN_TRUE"]="#"
S["DBUS_UNIX_FALSE"]="#"
S["DBUS_UNIX_TRUE"]=""
S["DBUS_WINCĒ_FALSE"]=""
S["DBUS_WINCĒ_TRUE"]="#"
S["DBUS_WIN_FALSE"]=""
S["DBUS_WIN_TRUE"]="#"
S["WINDRES"]=""
S["BUILD_FILEVERSION"]=""
S["BUILD_TIMESTAMP"]=""
S["RC"]=""
S["PKG_CONFIG"]="/home/gangadhar/newyoctobuild/tisdsk/build/arago-tmp-external-linaro-toolchain/sysroots/i686-linux/usr/bin/pkg-config"
S["CXXCPP"]="arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --sysroot=/home/gangadhar/newyoctobuild\"/tisdsk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1-E"
S["OTOOL64"]=""
S["OTOOL"]=""
S["LIPO"]=""
S["NMEDIT"]=""
S["DSYMUTIL"]=""
S["MANIFEST_TOOL"]=":"
S["RANLIB"]="arm-linux-gnueabi-hf-ranlib"
S["ac_ct_AR"]=""
S["AR"]="arm-linux-gnueabi-hf-ar"
S["DLLTOOL"]="false"
S["OBJDUMP"]="arm-linux-gnueabi-hf-objdump"
S["LN_S"]="ln -s"
```

```
S["NM"]="arm-linux-gnueabi-hf-nm"
S["ac_ct_DUMPBIN"]=" "
S["DUMPBIN"]=" "
S["LD"]="arm-linux-gnueabi-hf-ld --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1"
S["FGREP"]="/bin/grep -F"
S["SED"]="/bin/sed"
S["LIBTOOL"]="$ (top_builddir)/arm-oe-linux-gnueabi-libtool"
S["EGREP"]="/bin/grep -E"
S["GREP"]="/bin/grep"
S["CPP"]="arm-linux-gnueabi-hf-gcc -E --
sysroot=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-
linaro-toolchain/sysroots/delphi-vp1 -march=armv7" \
"-a -marm -mthumb-interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-
a8"
S["am__fastdepCXX_FALSE"]=" "
S["am__fastdepCXX_TRUE"]=" #"
S["CXXDEPMODE"]="depmode=none"
S["ac_ct_CXX"]=" "
S["CXXFLAGS"]="-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector -O2 -pipe -g -felim" \
"inate-unused-debug-types -fpermissive"
S["CXX"]="arm-linux-gnueabi-hf-g++ -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mcpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild" \
"/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1"
S["am__fastdepCC_FALSE"]=" "
S["am__fastdepCC_TRUE"]=" #"
S["CCDEPMODE"]="depmode=none"
S["am__nodep"]=" "
S["AMDEPBACKSLASH"]=" "
S["AMDEP_FALSE"]=" "
S["AMDEP_TRUE"]=" #"
S["am__quote"]=" "
S["am__include"]="include"
S["DEPDIR"]=" .deps"
S["OBJEXT"]="o"
S["EXEEXT"]=" "
S["ac_ct_CC"]=" "
S["CPPFLAGS"]="-isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-
4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/include -fstack-
protector"
S["LDFLAGS"]="-L/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-
2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -Wl,-rpath-
link,/home/gangadhar/gcc-linaro-arm-l" \
"inux-gnueabi-hf-4.7-2013.03-20130313_linux/arm-linux-gnueabi-hf/lib -
Wl,-O1 -Wl,--hash-style=gnu"
S["CFLAGS"]="-Wall -Wextra -Wchar-subscripts -Wmissing-declarations -
Wmissing-prototypes -Wnested-externs -Wpointer-arith -Wcast-align -
Wno-address -Wfloat-equa" \
```

```

"l -Wdeclaration-after-statement -Wno-unused-label -Wno-missing-field-
initializers -Wno-unused-parameter -Wno-sign-compare -Wno-pointer-sign
-Wno-typp"\
"e-limits -fno-common -fno-strict-aliasing -
isystem/home/gangadhar/gcc-linaro-arm-linux-gnueabi-hf-4.7-2013.03-
20130313_linux/arm-linux-gnueabi-hf/incl"\
"ude -fstack-protector -O2 -pipe -g -feliminate-unused-debug-types"
S["CC"]="arm-linux-gnueabi-hf-gcc -march=armv7-a -marm -mthumb-
interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a8 --
sysroot=/home/gangadhar/newyoctobuild"\
"/tisdk/build/arago-tmp-external-linaro-toolchain/sysroots/delphi-vp1"
S["DBUS_VERSION"]="1.6.8"
S["DBUS_MICRO_VERSION"]="8"
S["DBUS_MINOR_VERSION"]="6"
S["DBUS_MAJOR_VERSION"]="1"
S["LT_AGE"]="7"
S["LT_REVISION"]="2"
S["LT_CURRENT"]="10"
S["AM_BACKSLASH"]="\"\"
S["AM_DEFAULT_VERBOSITY"]="1"
S["AM_DEFAULT_V"]="$ (AM_DEFAULT_VERBOSITY) "
S["AM_V"]="$ (V) "
S["MAINT"]=""
S["MAINTAINER_MODE_FALSE"]="#"
S["MAINTAINER_MODE_TRUE"]=""
S["GETTEXT_PACKAGE"]="dbus-1"
S["am__untar"]="tar -xf -"
S["am__tar"]="tar --format=ustar -chf - \"$$stardir\""
S["AMTAR"]="$${TAR-tar}"
S["am__leading_dot"]="."
S["SET_MAKE"]=""
S["AWK"]="gawk"
S["mkdir_p"]="$ (MKDIR_P) "
S["MKDIR_P"]="/bin/mkdir -p"
S["INSTALL_STRIP_PROGRAM"]="$ (install_sh) -c -s"
S["STRIP"]="arm-linux-gnueabi-hf-strip"
S["install_sh"]="$${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"\
"-arago1/dbus-1.6.8/install-sh"
S["MAKEINFO"]="$${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"\
"-arago1/dbus-1.6.8/missing --run makeinfo"
S["AUTOHEADER"]="$${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"\
"-arago1/dbus-1.6.8/missing --run autoheader"

```

```
S["AUTOMAKE"]="${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"
"-arago1/dbus-1.6.8/missing --run automake-1.12"
S["AUTOCONF"]="${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"
"-arago1/dbus-1.6.8/missing --run autoconf"
S["ACLOCAL"]="${SHELL}
/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-
r6.0"
"-arago1/dbus-1.6.8/missing --run aclocal-1.12"
S["VERSION"]="1.6.8"
S["PACKAGE"]="dbus"
S["CYGPATH_W"]="echo"
S["am__isrc"]=""
S["INSTALL_DATA"]="${INSTALL} -m 644"
S["INSTALL_SCRIPT"]="${INSTALL}"
S["INSTALL_PROGRAM"]="${INSTALL}"
S["host_os"]="linux-gnueabi"
S["host_vendor"]="oe"
S["host_cpu"]="arm"
S["host"]="arm-oe-linux-gnueabi"
S["build_os"]="linux-gnu"
S["build_vendor"]="pc"
S["build_cpu"]="i686"
S["build"]="i686-pc-linux-gnu"
S["target_alias"]="arm-oe-linux-gnueabi"
S["host_alias"]="arm-oe-linux-gnueabi"
S["build_alias"]="i686-linux"
S["LIBS"]=""
S["ECHO_T"]=""
S["ECHO_N"]="-n"
S["ECHO_C"]=""
S["DEFS"]="-DHAVE_CONFIG_H"
S["mandir"]="/usr/share/man"
S["localedir"]="${datarootdir}/locale"
S["libdir"]="/usr/lib"
S["psdir"]="${docdir}"
S["pdfdir"]="${docdir}"
S["dvidir"]="${docdir}"
S["htmldir"]="${docdir}"
S["infodir"]="/usr/share/info"
S["docdir"]="${datarootdir}/doc/${PACKAGE_TARNAME}"
S["oldincludedir"]="/usr/include"
S["includedir"]="/usr/include"
S["localstatedir"]="/var"
S["sharedstatedir"]="/com"
S["sysconfdir"]="/etc"
```

```

S["datadir"]="/usr/share"
S["datarootdir"]="${prefix}/share"
S["libexecdir"]="/usr/lib/dbus"
S["sbindir"]="/usr/sbin"
S["bindir"]="/usr/bin"
S["program_transform_name"]="s,x,x,"
S["prefix"]="/usr"
S["exec_prefix"]="/usr"
S["PACKAGE_URL"]=""
S["PACKAGE_BUGREPORT"]="https://bugs.freedesktop.org/enter_bug.cgi?product=dbus"
S["PACKAGE_STRING"]="dbus 1.6.8"
S["PACKAGE_VERSION"]="1.6.8"
S["PACKAGE_TARNAME"]="dbus"
S["PACKAGE_NAME"]="dbus"
S["PATH_SEPARATOR"]=":"
S["SHELL"]="/bin/sh"
_ACAWK
cat >>"$ac_tmp/subs1.awk" <<_ACAWK &&
  for (key in S) S_is_set[key] = 1
  FS = " "

}
{
  line = $ 0
  nfields = split(line, field, "@")
  substed = 0
  len = length(field[1])
  for (i = 2; i < nfields; i++) {
    key = field[i]
    keylen = length(key)
    if (S_is_set[key]) {
      value = S[key]
      line = substr(line, 1, len) "" value "" substr(line, len +
keylen + 3)
      len += length(value) + length(field[++i])
      substed = 1
    } else
      len += 1 + keylen
  }

  print line
}

_ACAWK
if sed "s/$ac_cr//" < /dev/null > /dev/null 2>&1; then
  sed "s/$ac_cr\\$//; s/$ac_cr/$ac_cs_awk_cr/g"
else
  cat
fi < "$ac_tmp/subs1.awk" > "$ac_tmp/subs.awk" \
|| as_fn_error $? "could not setup config files machinery" "$LINENO"
5

```

```

fi # test -n "$CONFIG_FILES"

# Set up the scripts for CONFIG_HEADERS section.
# No need to generate them if there are no CONFIG_HEADERS.
# This happens for instance with './config.status Makefile'.
if test -n "$CONFIG_HEADERS"; then
cat >"$ac_tmp/defines.awk" <<\_ACAWK ||
BEGIN {
D["PACKAGE_NAME"]=" \\"dbus\\"\"
D["PACKAGE_TARNAME"]=" \\"dbus\\"\"
D["PACKAGE_VERSION"]=" \\"1.6.8\\"\"
D["PACKAGE_STRING"]=" \\"dbus 1.6.8\\"\"
D["PACKAGE_BUGREPORT"]="
\\"https://bugs.freedesktop.org/enter_bug.cgi?product=dbus\\"\"
D["PACKAGE_URL"]=" \\"\\\"\"
D["PACKAGE"]=" \\"dbus\\"\"
D["VERSION"]=" \\"1.6.8\\"\"
D["GETTEXT_PACKAGE"]=" \\"dbus-1\\"\"
D["DBUS_DAEMON_NAME"]=" \\"dbus-daemon\\"\"
D["STDC_HEADERS"]=" 1"
D["HAVE_SYS_TYPES_H"]=" 1"
D["HAVE_SYS_STAT_H"]=" 1"
D["HAVE_STDLIB_H"]=" 1"
D["HAVE_STRING_H"]=" 1"
D["HAVE_MEMORY_H"]=" 1"
D["HAVE_STRINGS_H"]=" 1"
D["HAVE_INTTYPES_H"]=" 1"
D["HAVE_STDINT_H"]=" 1"
D["HAVE_UNISTD_H"]=" 1"
D["__EXTENSIONS__"]=" 1"
D["__ALL_SOURCE"]=" 1"
D["__GNU_SOURCE"]=" 1"
D["__POSIX_PTHREAD_SEMANTICS"]=" 1"
D["__TANDEM_SOURCE"]=" 1"
D["STDC_HEADERS"]=" 1"
D["HAVE_DLFCN_H"]=" 1"
D["LT_OBJDIR"]=" \\".libs/\\"\"
D["DBUS_UNIX"]=" 1"
D["DBUS_WITH_GLIB"]=" 1"
D["DBUS_DISABLE_ASSERT"]=" 1"
D["DBUS_DISABLE_CHECKS"]=" 1"
D["G_DISABLE_CHECKS"]=" 1"
D["DBUS_ENABLE_USERDB_CACHE"]=" 1"
D["SIZEOF_CHAR"]=" 1"
D["SIZEOF_SHORT"]=" 2"
D["SIZEOF_LONG"]=" 4"
D["SIZEOF_INT"]=" 4"
D["SIZEOF_VOID_P"]=" 4"
D["SIZEOF_LONG_LONG"]=" 8"
D["SIZEOF__INT64"]=" 0"
D["DBUS_INT64_PRINTF_MODIFIER"]=" \\"l1\\"\"
D["DBUS_VA_COPY"]=" va_copy"

```

```
D["DBUS_USE_SYNC"]=" 1"  
D["HAVE_VSNPRINTF"]=" 1"  
D["HAVE_VASPRINTF"]=" 1"  
D["HAVE_NANOSLEEP"]=" 1"  
D["HAVE_USLEEP"]=" 1"  
D["HAVE_SETENV"]=" 1"  
D["HAVE_CLEARENV"]=" 1"  
D["HAVE_UNSETENV"]=" 1"  
D["HAVE_SOCKETPAIR"]=" 1"  
D["HAVE_GETGROUPLIST"]=" 1"  
D["HAVE_FPATHCONF"]=" 1"  
D["HAVE_SETRLIMIT"]=" 1"  
D["HAVE_POLL"]=" 1"  
D["HAVE_SETLOCALE"]=" 1"  
D["HAVE_LOCALECONV"]=" 1"  
D["HAVE_STRTOLL"]=" 1"  
D["HAVE_STRTOULL"]=" 1"  
D["HAVE_GETRESUID"]=" 1"  
D["HAVE_SYSLOG_H"]=" 1"  
D["HAVE_DECL_LOG_PERROR"]=" 1"  
D["HAVE_DIRFD"]=" 1"  
D["HAVE_SYS_RESOURCE_H"]=" 1"  
D["HAVE_DIRENT_H"]=" 1"  
D["HAVE_EXECINFO_H"]=" 1"  
D["HAVE_BACKTRACE"]=" 1"  
D["HAVE_ERRNO_H"]=" 1"  
D["HAVE_SIGNAL_H"]=" 1"  
D["HAVE_LOCALE_H"]=" 1"  
D["HAVE_BYTESWAP_H"]=" 1"  
D["HAVE_UNISTD_H"]=" 1"  
D["HAVE_POSIX_GETPWNAM_R"]=" 1"  
D["HAVE_SOCKLEN_T"]=" 1"  
D["HAVE_SYS_UIO_H"]=" 1"  
D["HAVE_WRITEV"]=" 1"  
D["HAVE_DECL_MSG_NOSIGNAL"]=" 1"  
D["HAVE_ISO_VARARGS"]=" 1"  
D["HAVE_GNUC_VARARGS"]=" 1"  
D["HAVE_PIPE2"]=" 1"  
D["HAVE_ACCEPT4"]=" 1"  
D["HAVE_ABSTRACT_SOCKETS"]=" 1"  
D["HAVE_EXPAT_H"]=" 1"  
D["HAVE_MONOTONIC_CLOCK"]=" 1"  
D["HAVE_SYS_INOTIFY_H"]=" 1"  
D["DBUS_BUS_ENABLE_INOTIFY"]=" 1"  
D["HAVE_INOTIFY_INIT1"]=" 1"  
D["DBUS_HAVE_LINUX_EPOLL"]=" 1"  
D["HAVE_UNIX_FD_PASSING"]=" 1"  
D["X_DISPLAY_MISSING"]=" 1"  
D["DBUS_SYSTEM_SOCKET"]=" \"/var/run/dbus/system_bus_socket\""  
D["DBUS_SYSTEM_BUS_DEFAULT_ADDRESS"]="  
\"unix:path=/var/run/dbus/system_bus_socket\""  
D["DBUS_CONSOLE_AUTH_DIR"]=" \"/var/run/console/\""
```

```

D["DBUS_CONSOLE_OWNER_FILE"]=" \"\"
D["DBUS_USER"]=" \"messagebus\"
D["DBUS_PREFIX"]=" \"/usr\"
D["DBUS_DATADIR"]=" \"/usr/share\"
D["DBUS_DAEMONDIR"]=" \"/usr/bin\"
D["DBUS_BINDIR"]=" \"/usr/bin\"
D["DBUS_LIBEXECDIR"]=" \"/usr/lib/dbus\"
D["DBUS_TEST_EXEC"]="
\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/\"
dbus/1.6.8-r6.0-arago1/dbus-1.6.8/test\"
D["DBUS_EXEEXT"]=" \"\"
D["TEST_BUS_BINARY"]="
\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi\"
\"/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/bus/dbus-daemon\"
D["DBUS_TEST_LAUNCH_HELPER_BINARY"]="
\"/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-
toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8/bus/dbus-daemon-
launch-helper-test\"
D["DBUS_TEST_SOCKET_DIR"]=" \"/tmp\"
D["TEST_LISTEN"]=" \"unix:tmpdir=/tmp\"
D["DBUS_SESSION_SOCKET_DIR"]=" \"/tmp\"
    for (key in D) D_is_set[key] = 1
    FS = " "
}
/^[\t ]*#[\t ]*(define|undef)[\t
]+[_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ][_abcdefghijk
lmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]*([\t (]|$)/ {
    line = $ 0
    split(line, arg, " ")
    if (arg[1] == "#") {
        defundef = arg[2]
        mac1 = arg[3]
    } else {
        defundef = substr(arg[1], 2)
        mac1 = arg[2]
    }
    split(mac1, mac2, "(") #)
    macro = mac2[1]
    prefix = substr(line, 1, index(line, defundef) - 1)
    if (D_is_set[macro]) {
        # Preserve the white space surrounding the "#".
        print prefix "define", macro P[macro] D[macro]
        next
    } else {
        # Replace #undef with comments. This is necessary, for example,
        # in the case of _POSIX_SOURCE, which is predefined and required
        # on some systems where configure will not decide to define it.
        if (defundef == "undef") {
            print "/*", prefix defundef, macro, "*/"
        }
    }
}

```



```

        next
    }
}
{ print }
_ACAWK
as_fn_error $? "could not setup config headers machinery" "$LINENO"
5
fi # test -n "$CONFIG_HEADERS"

eval set X " :F $CONFIG_FILES :H $CONFIG_HEADERS :C
$CONFIG_COMMANDS"
shift
for ac_tag
do
    case $ac_tag in
        :[FHLC]) ac_mode=$ac_tag; continue;;
    esac
    case $ac_mode$ac_tag in
        :[FHL]*:*) ;;
        :L* | :C*:*) as_fn_error $? "invalid tag \`$ac_tag'" "$LINENO" 5;;
        :[FH]-) ac_tag=-:-;;
        :[FH]*) ac_tag=$ac_tag:$ac_tag.in;;
    esac
    ac_save_IFS=$IFS
    IFS=:
    set x $ac_tag
    IFS=$ac_save_IFS
    shift
    ac_file=$1
    shift

    case $ac_mode in
        :L) ac_source=$1;;
        :[FH])
            ac_file_inputs=
            for ac_f
            do
                case $ac_f in
                    -) ac_f="$ac_tmp/stdin";;
                    *) # Look for the file first in the build tree, then in the
source tree
                    # (if the path is not absolute). The absolute path cannot be
DOS-style,
                    # because $ac_f cannot contain `:'.
                    test -f "$ac_f" ||
                    case $ac_f in
                        [\\/$]*) false;;
                    *) test -f "$srcdir/$ac_f" && ac_f="$srcdir/$ac_f";;
                    esac ||
                    as_fn_error 1 "cannot find input file: \`$ac_f'" "$LINENO" 5;;
                esac
            done
        esac
    done
done

```



```

        /^X\(\\\/\).*\/{
            s//\1/
            q
        }
        s/.*/./; q'`
as_dir="$ac_dir"; as_fn_mkdir_p
ac_builddir=.

case "$ac_dir" in
.) ac_dir_suffix= ac_top_builddir_sub=. ac_top_build_prefix= ;;
*)
    ac_dir_suffix=`$as_echo "$ac_dir" | sed 's|^\.([\\/]|||'`
    # A "." for each directory in $ac_dir_suffix.
    ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/|/[\^\/]*|/..|g;s|/|||'`
    case $ac_top_builddir_sub in
    "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
    *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
    esac ;;
esac
ac_abs_top_builddir=$ac_pwd
ac_abs_builddir=$ac_pwd$ac_dir_suffix
# for backward compatibility:
ac_top_builddir=$ac_top_build_prefix

case $srcdir in
.) # We are building in place.
    ac_srcdir=.
    ac_top_srcdir=$ac_top_builddir_sub
    ac_abs_top_srcdir=$ac_pwd ;;
[\/]* | ?:[\/]* ) # Absolute name.
    ac_srcdir=$srcdir$ac_dir_suffix;
    ac_top_srcdir=$srcdir
    ac_abs_top_srcdir=$srcdir ;;
*) # Relative name.
    ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
    ac_top_srcdir=$ac_top_build_prefix$srcdir
    ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

case $ac_mode in
:F)
#
# CONFIG_FILE
#

case $INSTALL in
[\/$]* | ?:[\/]* ) ac_INSTALL=$INSTALL ;;
*) ac_INSTALL=$ac_top_build_prefix$INSTALL ;;
esac

```

```

ac_MKDIR_P=$MKDIR_P
case $MKDIR_P in
  [\\/$]* | ?:[\\/]*) ;;
  /*) ac_MKDIR_P=$ac_top_build_prefix$MKDIR_P ;;
esac
# If the template does not know about datarootdir, expand it.
# FIXME: This hack should be removed a few years after 2.60.
ac_datarootdir_hack=; ac_datarootdir_seen=
ac_sed_dataroot='
/datarootdir/ {
  p
  q
}
/@datadir@/p
/@docdir@/p
/@infodir@/p
/@localedir@/p
/@mandir@/p'
case `eval "sed -n \"\$ac_sed_dataroot\" \$ac_file_inputs" ` in
*datarootdir*) ac_datarootdir_seen=yes;;
*@datadir@*|*@docdir@*|*@infodir@*|*@localedir@*|*@mandir@*)
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file_inputs
seems to ignore the --datarootdir setting" >&5
$as_echo "$as_me: WARNING: $ac_file_inputs seems to ignore the --
datarootdir setting" >&2;}
  ac_datarootdir_hack='
s&@datadir@&/usr/share&g
s&@docdir@&${datarootdir}/doc/${PACKAGE_TARNAME}&g
s&@infodir@&/usr/share/info&g
s&@localedir@&${datarootdir}/locale&g
s&@mandir@&/usr/share/man&g
s&\${datarootdir}&${prefix}/share&g' ;;
esac
ac_sed_extra="/^[          ]*VPATH[          ]*=[          ]*/{
h
s///
s/^\://
s/[          ]*$$/:/
s/:\$(srcdir)::/g
s/:\${srcdir}::/g
s/:@srcdir@::/g
s/^\:*//
s/:\:*//
x
s/\(=[          ]*\)\.*/\1/
G
s/\n//
s/^[^=]*=[          ]*$//
}

:t
/@[a-zA-Z_][a-zA-Z_0-9]*@/!b

```

```

s|@configure_input@|${ac_sed_conf_input}|;t t
s@top_builddir@&${ac_top_builddir_sub}&;t t
s@top_build_prefix@&${ac_top_build_prefix}&;t t
s@srcdir@&${ac_srcdir}&;t t
s@abs_srcdir@&${ac_abs_srcdir}&;t t
s@top_srcdir@&${ac_top_srcdir}&;t t
s@abs_top_srcdir@&${ac_abs_top_srcdir}&;t t
s@builddir@&${ac_builddir}&;t t
s@abs_builddir@&${ac_abs_builddir}&;t t
s@abs_top_builddir@&${ac_abs_top_builddir}&;t t
s@INSTALL@&${ac_INSTALL}&;t t
s@MKDIR_P@&${ac_MKDIR_P}&;t t
${ac_datarootdir_hack}
"
eval sed \"\${ac_sed_extra}\" "${ac_file_inputs}" | $AWK -f
"${ac_tmp}/subs.awk" \
>${ac_tmp}/out || as_fn_error $? "could not create $ac_file" "$LINENO"
5

test -z "${ac_datarootdir_hack}${ac_datarootdir_seen}" &&
{ ac_out=`sed -n '/\${datarootdir}/p' "${ac_tmp}/out"`; test -n
"${ac_out}"; } &&
{ ac_out=`sed -n '/^[ ]*datarootdir[ ]*:*=/p' \
"${ac_tmp}/out"`; test -z "${ac_out}"; } &&
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file contains
a reference to the variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&5
$as_echo "$as_me: WARNING: $ac_file contains a reference to the
variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&2;}

rm -f "${ac_tmp}/stdin"
case $ac_file in
-) cat "${ac_tmp}/out" && rm -f "${ac_tmp}/out";;
*) rm -f "$ac_file" && mv "${ac_tmp}/out" "$ac_file";;
esac \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
;;
:H)
#
# CONFIG_HEADER
#
if test x"$ac_file" != x-; then
{
$as_echo "/* $configure_input */" \
&& eval '$AWK -f "${ac_tmp}/defines.awk" "${ac_file_inputs}"
} >"${ac_tmp}/config.h" \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
if diff "$ac_file" "${ac_tmp}/config.h" >/dev/null 2>&1; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: $ac_file is unchanged"
>&5
$as_echo "$as_me: $ac_file is unchanged" >&6;}

```

```

else
  rm -f "$ac_file"
  mv "$ac_tmp/config.h" "$ac_file" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5
fi
else
  $as_echo "/* $configure_input */" \
  && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs" \
  || as_fn_error $? "could not create -" "$LINENO" 5
fi
# Compute "$ac_file"'s index in $config_headers.
_am_arg="$ac_file"
_am_stamp_count=1
for _am_header in $config_headers ;; do
  case $_am_header in
    $_am_arg | $_am_arg:* )
      break ;;
    * )
      _am_stamp_count=`expr $_am_stamp_count + 1` ;;
  esac
done
echo "timestamp for $_am_arg" >`$as_dirname -- "$_am_arg" ||
$as_expr X"$_am_arg" : 'X\(.*[^/]\)\([^/]*\)[^/]*/*$' \|| \
  X"$_am_arg" : 'X\(//\)[^/]' \|| \
  X"$_am_arg" : 'X\(//\)$' \|| \
  X"$_am_arg" : 'X\(/\)' \|| . 2>/dev/null ||
$as_echo X"$_am_arg" |
  sed '/^X\(.*[^/]\)\([^/]*\)[^/]*/*$/{
    s//\1/
    q
  }
/^X\(\\\/\)\([^\].*/{
  s//\1/
  q
}
/^X\(\\\/\)$/{
  s//\1/
  q
}
/^X\(\\\/\).*/{
  s//\1/
  q
}
s/.*\/./; q'`/stamp-h$_am_stamp_count

;;

:C) { $as_echo "$as_me:${as_lineno-$LINENO}: executing $ac_file
commands" >&5
$as_echo "$as_me: executing $ac_file commands" >&6;}
;;
esac

```



```

fi
# Extract the definition of DEPDIR, am__include, and am__quote
# from the Makefile without running 'make'.
DEPDIR=`sed -n 's/^DEPDIR = //p' < "$mf"`
test -z "$DEPDIR" && continue
am__include=`sed -n 's/^am__include = //p' < "$mf"`
test -z "am__include" && continue
am__quote=`sed -n 's/^am__quote = //p' < "$mf"`
# Find all dependency output files, they are included files with
# $(DEPDIR) in their names. We invoke sed twice because it is the
# simplest approach to changing $(DEPDIR) to its actual value in
the
# expansion.
for file in `sed -n "
  s/^$am__include $am__quote\(.*(DEPDIR).*\) $am__quote" '$/\1/p'
<"$mf" | \
  sed -e 's/\$(DEPDIR)/"$DEPDIR"/g`; do
# Make sure the directory exists.
test -f "$dirpart/$file" && continue
fdir=`$as_dirname -- "$file" ||
$as_expr X"$file" : 'X\([^\/]\)\/*\([^\/]\)\/*$' \| \
  X"$file" : 'X\(\//\)\[\^\/]' \| \
  X"$file" : 'X\(\//\)$' \| \
  X"$file" : 'X\(\//\)' \| . 2>/dev/null ||
$as_echo X"$file" |
  sed '/^\X\(\.[^\/]\)\[\^\/]\*\[\^\/]\*\[\^\/]\*\$/{
    s//\1/
    q
  }
/^X\(\//\)\[\^\/]\.*/{
  s//\1/
  q
}
/^X\(\//\)\$/{
  s//\1/
  q
}
/^X\(\//\)\.*/{
  s//\1/
  q
}
s/.*\/./; q'`
as_dir=$dirpart/$fdir; as_fn_mkdir_p
# echo "creating $dirpart/$file"
echo '# dummy' > "$dirpart/$file"
done
done
}
;;
"libtool":C)

```



```

    # See if we are running on zsh, and set the options which allow
our
    # commands through without removal of \ escapes.
    if test -n "${ZSH_VERSION+set}" ; then
        setopt NO_GLOB_SUBST
    fi

    cfgfile="${ofile}T"
    trap "$RM \"\$cfgfile\"; exit 1" 1 2 15
    $RM "$cfgfile"

    cat <<_LT_EOF >> "$cfgfile"
#! $SHELL

# `ECHO "$ofile" | sed 's%^.*/%%'` - Provide generalized library-
building support services.
# Generated automatically by $as_me ($PACKAGE$TIMESTAMP) $VERSION
# Libtool was configured on host `(hostname || uname -n) 2>/dev/null |
sed lq`:
# NOTE: Changes made to this file will be lost: look at ltmain.sh.
#
# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005,
#           2006, 2007, 2008, 2009, 2010, 2011 Free Software
#           Foundation, Inc.
# Written by Gordon Matzigkeit, 1996
#
# This file is part of GNU Libtool.
#
# GNU Libtool is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of
# the License, or (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html, or
# obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

# The names of the tagged configurations supported by this script.

```

```
available_tags="CXX RC "  
  
# ### BEGIN LIBTOOL CONFIG  
  
# Which release of libtool.m4 was used?  
macro_version=$macro_version  
macro_revision=$macro_revision  
  
# Whether or not to build shared libraries.  
build_libtool_libs=$enable_shared  
  
# Whether or not to build static libraries.  
build_old_libs=$enable_static  
  
# What type of objects to build.  
pic_mode=$pic_mode  
  
# Whether or not to optimize for fast installation.  
fast_install=$enable_fast_install  
  
# Shell to use when invoking shell scripts.  
SHELL=$lt_SHELL  
  
# An echo program that protects backslashes.  
ECHO=$lt_ECHO  
  
# The PATH separator for the build system.  
PATH_SEPARATOR=$lt_PATH_SEPARATOR  
  
# The host system.  
host_alias=$host_alias  
host=$host  
host_os=$host_os  
  
# The build system.  
build_alias=$build_alias  
build=$build  
build_os=$build_os  
  
# A sed program that does not truncate output.  
SED=$lt_SED  
  
# Sed that helps us avoid accidentally triggering echo(1) options like  
-n.  
Xsed="\$SED -e 1s/^X/"  
  
# A grep program that handles long lines.  
GREP=$lt_GREP  
  
# An ERE matcher.  
EGREP=$lt_EGREP
```

```
# A literal string matcher.
FGREP=$lt_FGREP

# A BSD- or MS-compatible name lister.
NM=$lt_NM

# Whether we need soft or hard links.
LN_S=$lt_LN_S

# What is the maximum length of a command?
max_cmd_len=$max_cmd_len

# Object file suffix (normally "o").
objext=$ac_objext

# Executable file suffix (normally "").
exeext=$exeext

# whether the shell understands "unset".
lt_unset=$lt_unset

# turn spaces into newlines.
SP2NL=$lt_lt_SP2NL

# turn newlines into spaces.
NL2SP=$lt_lt_NL2SP

# convert \${build} file names to \${host} format.
to_host_file_cmd=$lt_cv_to_host_file_cmd

# convert \${build} files to toolchain format.
to_tool_file_cmd=$lt_cv_to_tool_file_cmd

# An object symbol dumper.
OBJDUMP=$lt_OBJDUMP

# Method to check whether dependent libraries are shared objects.
deplibs_check_method=$lt_deplibs_check_method

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd=$lt_file_magic_cmd

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=$lt_file_magic_glob

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob=$lt_want_nocaseglob

# DLL creation program.
DLLTOOL=$lt_DLLTOOL
```

```
# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd=$lt_sharedlib_from_linklib_cmd

# The archiver.
AR=$lt_AR

# Flags to create an archive.
AR_FLAGS=$lt_AR_FLAGS

# How to feed a file listing to the archiver.
archiver_list_spec=$lt_archiver_list_spec

# A symbol stripping program.
STRIP=$lt_STRIP

# Commands used to install an old-style archive.
RANLIB=$lt_RANLIB
old_postinstall_cmds=$lt_old_postinstall_cmds
old_postuninstall_cmds=$lt_old_postuninstall_cmds

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=$lock_old_archive_extraction

# A C compiler.
LTCC=$lt_CC

# LTCC compiler flags.
LTCFLAGS=$lt_CFLAGS

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe=$lt_lt_cv_sys_global_symbol_pipe

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl=$lt_lt_cv_sys_global_symbol_to_cdecl

# Transform the output of nm in a C name address pair.
global_symbol_to_c_name_address=$lt_lt_cv_sys_global_symbol_to_c_name_
address

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix=$lt_lt_cv_sys_global_symbol
_to_c_name_address_lib_prefix

# Specify filename containing input files for \ $NM.
nm_file_list_spec=$lt_nm_file_list_spec

# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=$lt_sysroot
```

```
# The name of the directory that contains temporary libtool files.
objdir=$objdir

# Used to examine libraries when file_magic_cmd begins with "file".
MAGIC_CMD=$MAGIC_CMD

# Must we lock files when doing compilation?
need_locks=$lt_need_locks

# Manifest tool.
MANIFEST_TOOL=$lt_MANIFEST_TOOL

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
DSYMUTIL=$lt_DSYMUTIL

# Tool to change global to local symbols on Mac OS X.
NMEDIT=$lt_NMEDIT

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=$lt_LIPO

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=$lt_OTOOL

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=$lt_OTOOL64

# Old archive suffix (normally "a").
libext=$libext

# Shared library suffix (normally ".so").
shrext_cmds=$lt_shrext_cmds

# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=$lt_extract_expsyms_cmds

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink=$lt_variables_saved_for_relink

# Do we need the "lib" prefix for modules?
need_lib_prefix=$need_lib_prefix

# Do we need a version for libraries?
need_version=$need_version

# Library versioning type.
version_type=$version_type
```

```
# Shared library runtime path variable.
runpath_var=$runpath_var

# Shared library path variable.
shlibpath_var=$shlibpath_var

# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=$shlibpath_overrides_runpath

# Format of library name prefix.
libname_spec=$lt_libname_spec

# List of archive names.  First name is the real one, the rest are
links.
# The last name is the one that the linker finds with -lNAME
library_names_spec=$lt_library_names_spec

# The coded name of the library, if different from the real name.
soname_spec=$lt_soname_spec

# Permission mode override for installation of shared libraries.
install_override_mode=$lt_install_override_mode

# Command to use after installation of a shared archive.
postinstall_cmds=$lt_postinstall_cmds

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=$lt_postuninstall_cmds

# Commands used to finish a libtool library installation in a
directory.
finish_cmds=$lt_finish_cmds

# As "finish_cmds", except a single script fragment to be evaled but
# not shown.
finish_eval=$lt_finish_eval

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=$hardcode_into_libs

# Compile-time system search path for libraries.
sys_lib_search_path_spec=$lt_sys_lib_search_path_spec

# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec=$lt_sys_lib_dlsearch_path_spec

# Whether dlopen is supported.
dlopen_support=$enable_dlopen

# Whether dlopen of programs is supported.
dlopen_self=$enable_dlopen_self
```

```
# Whether dlopen of statically linked programs is supported.
dlopen_self_static=$enable_dlopen_self_static

# Commands to strip libraries.
old_striplib=$lt_old_striplib
striplib=$lt_striplib

# The linker used to build libraries.
LD=$lt_LD

# How to create reloadable object files.
reload_flag=$lt_reload_flag
reload_cmds=$lt_reload_cmds

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds

# A language specific compiler.
CC=$lt_compiler

# Is the compiler the GNU compiler?
with_gcc=$GCC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec
```

```
# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds
archive_expsym_cmds=$lt_archive_expsym_cmds

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds
module_expsym_cmds=$lt_module_expsym_cmds

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
```



```
hardcode_minus_L=$hardcode_minus_L

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects
```

```

postdep_objects=$lt_postdep_objects
predeps=$lt_predeps
postdeps=$lt_postdeps

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path

# ### END LIBTOOL CONFIG

_LT_EOF

case $host_os in
aix3*)
  cat <<\_LT_EOF >> "$cfgfile"
# AIX sometimes has problems with the GCC collect2 program.  For some
# reason, if we set the COLLECT_NAMES environment variable, the
problems
# vanish in a puff of smoke.
if test "X${COLLECT_NAMES+set}" != Xset; then
  COLLECT_NAMES=
  export COLLECT_NAMES
fi
_LT_EOF
;;
esac

ltmain="$ac_aux_dir/ltmain.sh"

# We use sed instead of cat because bash on DJGPP gets confused if
# if finds mixed CR/LF and LF-only lines.  Since sed operates in
# text mode, it properly converts lines to CR/LF.  This bash problem
# is reportedly fixed, but why not run on old versions too?
sed '$q' "$ltmain" >> "$cfgfile" \
  || (rm -f "$cfgfile"; exit 1)

if test x"$xsi_shell" = xyes; then
  sed -e '/^func_dirname ()$/,/^{}/ # func_dirname /c\
func_dirname ()\
{\
\   case ${1} in\
\     */*) func_dirname_result="${1%/*}${2}" ;;\
\     * ) func_dirname_result="${3}" ;;\
\   esac\
} # Extended-shell func_dirname implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_basename ()$/,/^{ # func_basename /c\
func_basename ()\
{\
\    func_basename_result="${1##*/}"\
} # Extended-shell func_basename implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_dirname_and_basename ()$/,/^{ #
func_dirname_and_basename /c\
func_dirname_and_basename ()\
{\
\    case ${1} in\
\        */*) func_dirname_result="${1%/*}${2}" ;;\
\        * ) func_dirname_result="${3}" ;;\
\    esac\
\    func_basename_result="${1##*/}"\
} # Extended-shell func_dirname_and_basename implementation'
"$cfgfile" > $cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_stripname ()$/,/^{ # func_stripname /c\
func_stripname ()\
{\
\    # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are\
\    # positional parameters, so assign one to ordinary parameter
first.\
\    func_stripname_result=${3}\
\    func_stripname_result=${func_stripname_result#"${1}"}\
\    func_stripname_result=${func_stripname_result%"${2}"}\
} # Extended-shell func_stripname implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_split_long_opt ()$/,/^{ # func_split_long_opt /c\
func_split_long_opt ()\
{\
\    func_split_long_opt_name=${1%%=*}\

```

```

\   func_split_long_opt_arg=${1#*=}\
} # Extended-shell func_split_long_opt implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_split_short_opt ()$/,/^\} # func_split_short_opt /c\
func_split_short_opt ()\
{\
\   func_split_short_opt_arg=${1#??}\
\   func_split_short_opt_name=${1%"$func_split_short_opt_arg"}\
} # Extended-shell func_split_short_opt implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_lo2o ()$/,/^\} # func_lo2o /c\
func_lo2o ()\
{\
\   case ${1} in\
\     *.lo) func_lo2o_result=${1%.lo}.${objext} ;;\
\     *)    func_lo2o_result=${1} ;;\
\   esac\
} # Extended-shell func_lo2o implementation' "$cfgfile" > $cfgfile.tmp
\
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_xform ()$/,/^\} # func_xform /c\
func_xform ()\
{\
\   func_xform_result=${1%.*}.lo\
} # Extended-shell func_xform implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_arith ()$/,/^\} # func_arith /c\
func_arith ()\
{\

```

```

    func_arith_result=$(( $* ))\
} # Extended-shell func_arith implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_len ()$/,/^{ } # func_len /c\
func_len ()\
{\
    func_len_result=${#1}\
} # Extended-shell func_len implementation' "$cfgfile" > $cfgfile.tmp
\
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

fi

if test x"$lt_shell_append" = xyes; then
    sed -e '/^func_append ()$/,/^{ } # func_append /c\
func_append ()\
{\
    eval "${1}+=\\${2}"\
} # Extended-shell func_append implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_append_quoted ()$/,/^{ } # func_append_quoted /c\
func_append_quoted ()\
{\
    func_quote_for_eval "${2}"\
    eval "${1}+=\\\ \\$func_quote_for_eval_result"\
} # Extended-shell func_append_quoted implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    # Save a `func_append' function call where possible by direct use of
    '+='
    sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1+= "%g' $cfgfile >
$cfgfile.tmp \

```

```

    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
    test 0 -eq $? || _lt_function_replace_fail=:
else
    # Save a `func_append' function call even when '+=' is not available
    sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1="$\1%g' $cfgfile >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
    test 0 -eq $? || _lt_function_replace_fail=:
fi

if test x"$_lt_function_replace_fail" = x:""; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: Unable to
substitute extended shell functions in $ofile" >&5
$as_echo "$as_me: WARNING: Unable to substitute extended shell
functions in $ofile" >&2;}
fi

    mv -f "$cfgfile" "$ofile" ||
    (rm -f "$ofile" && cp "$cfgfile" "$ofile" && rm -f "$cfgfile")
    chmod +x "$ofile"

    cat <<_LT_EOF >> "$ofile"

# ### BEGIN LIBTOOL TAG CONFIG: CXX

# The linker used to build libraries.
LD=$lt_LD_CXX

# How to create reloadable object files.
reload_flag=$lt_reload_flag_CXX
reload_cmds=$lt_reload_cmds_CXX

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds_CXX

# A language specific compiler.
CC=$lt_compiler_CXX

# Is the compiler the GNU compiler?
with_gcc=$GCC_CXX

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag_CXX

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic_CXX

```

```
# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl_CXX

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static_CXX

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o_CXX

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc_CXX

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes_CXX

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec_CXX

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec_CXX

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object_CXX

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds_CXX

# Create a temporary old-style archive to link instead of a shared archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds_CXX

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds_CXX
archive_expsym_cmds=$lt_archive_expsym_cmds_CXX

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds_CXX
module_expsym_cmds=$lt_module_expsym_cmds_CXX

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld_CXX

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag_CXX

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag_CXX
```

```
# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=${lt_hardcode_libdir_flag_spec_CXX}

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=${lt_hardcode_libdir_separator_CXX}

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=${hardcode_direct_CXX}

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=${hardcode_direct_absolute_CXX}

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=${hardcode_minus_L_CXX}

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=${hardcode_shlibpath_var_CXX}

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=${hardcode_automatic_CXX}

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=${inherit_rpath_CXX}

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=${link_all_deplibs_CXX}

# Set to "yes" if exported symbols are required.
always_export_symbols=${always_export_symbols_CXX}

# The commands to list exported symbols.
export_symbols_cmds=${lt_export_symbols_cmds_CXX}

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=${lt_exclude_expsyms_CXX}
```



```
# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms_CXX

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds_CXX

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds_CXX

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec_CXX

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action_CXX

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs_CXX

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects_CXX
postdep_objects=$lt_postdep_objects_CXX
predeps=$lt_predeps_CXX
postdeps=$lt_postdeps_CXX

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path_CXX

# ### END LIBTOOL TAG CONFIG: CXX
_LT_EOF

    cat <<_LT_EOF >> "$ofile"

# ### BEGIN LIBTOOL TAG CONFIG: RC

# The linker used to build libraries.
LD=$lt_LD_RC

# How to create reloadable object files.
reload_flag=$lt_reload_flag_RC
reload_cmds=$lt_reload_cmds_RC

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds_RC

# A language specific compiler.
CC=$lt_compiler_RC
```

```
# Is the compiler the GNU compiler?
with_gcc=$GCC_RC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag_RC

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic_RC

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl_RC

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static_RC

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o_RC

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc_RC

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes_RC

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec_RC

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec_RC

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object_RC

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds_RC

# Create a temporary old-style archive to link instead of a shared archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds_RC

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds_RC
archive_expsym_cmds=$lt_archive_expsym_cmds_RC

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds_RC
module_expsym_cmds=$lt_module_expsym_cmds_RC

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld_RC
```

```
# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag_RC

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag_RC

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec_RC

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator_RC

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct_RC

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute_RC

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L_RC

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var_RC

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic_RC

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath_RC

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs_RC

# Set to "yes" if exported symbols are required.
```

```
always_export_symbols=$always_export_symbols_RC

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds_RC

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms_RC

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms_RC

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds_RC

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds_RC

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec_RC

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action_RC

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs_RC

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects_RC
postdep_objects=$lt_postdep_objects_RC
predeps=$lt_predeps_RC
postdeps=$lt_postdeps_RC

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path_RC

# ### END LIBTOOL TAG CONFIG: RC
_LT_EOF

;;

esac
done # for ac_tag

as_fn_exit 0
```

File = config.sub

```
#!/bin/sh
# Configuration validation subroutine script.
# Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
# 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
# 2011, 2012 Free Software Foundation, Inc.

timestamp='2012-07-31'

# This file is (in principle) common to ALL GNU software.
# The presence of a machine in this file suggests that SOME GNU
software
# can handle that machine. It does not imply ALL GNU software can.
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, see <http://www.gnu.org/licenses/>.
#
# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program that contains a
# configuration script generated by Autoconf, you may include it under
# the same distribution terms that you use for the rest of that
program.

# Please send patches to <config-patches@gnu.org>. Submit a context
# diff and a properly formatted GNU ChangeLog entry.
#
# Configuration subroutine to validate and canonicalize a
configuration type.
# Supply the specified configuration type as an argument.
# If it is invalid, we print an error message on stderr and exit with
code 1.
# Otherwise, we print the canonical config type on stdout and succeed.

# You can get the latest version of this script from:
#
http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config
.sub;hb=HEAD

# This file is supposed to be the same for all GNU packages
# and recognize all the CPU types, system types and aliases
```

```
# that are meaningful with *any* GNU software.
# Each package is responsible for reporting which valid configurations
# it does not support.  The user should be able to distinguish
# a failure to support a valid configuration from a meaningless
# configuration.
```

```
# The goal of this file is to map all the various variations of a
given
# machine specification into a single specification in the form:
#   CPU_TYPE-MANUFACTURER-OPERATING_SYSTEM
# or in some cases, the newer four-part form:
#   CPU_TYPE-MANUFACTURER-KERNEL-OPERATING_SYSTEM
# It is wrong to echo any other type of specification.
```

```
me=`echo "$0" | sed -e 's,./,,'`
```

```
usage="\
Usage: $0 [OPTION] CPU-MFR-OPSYS
       $0 [OPTION] ALIAS
```

```
Canonicalize a configuration name.
```

```
Operation modes:
```

```
  -h, --help           print this help, then exit
  -t, --time-stamp     print date of last modification, then exit
  -v, --version        print version number, then exit
```

```
Report bugs and patches to <config-patches@gnu.org>."
```

```
version="\
GNU config.sub ($timestamp)
```

```
Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012
Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions.  There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE."
```

```
help="
Try `"$me" --help' for more information."
```

```
# Parse command line
while test $# -gt 0 ; do
  case $1 in
    --time-stamp | --time* | -t )
      echo "$timestamp" ; exit ;;
    --version | -v )
      echo "$version" ; exit ;;
    --help | --h* | -h )
```

```

    echo "$usage"; exit ;;
-- )      # Stop option processing
    shift; break ;;
- )      # Use stdin as input.
    break ;;
-* )
    echo "$me: invalid option $1$help"
    exit 1 ;;

*local*)
    # First pass through any local machine types.
    echo $1
    exit ;;

* )
    break ;;
esac
done

case $# in
0) echo "$me: missing argument$help" >&2
    exit 1;;
1) ;;
*) echo "$me: too many arguments$help" >&2
    exit 1;;
esac

# Separate what the user gave into CPU-COMPANY and OS or KERNEL-OS (if
any).
# Here we must recognize all the valid KERNEL-OS combinations.
maybe_os=`echo $1 | sed 's/^\(.*\) - \([^-]*-[-]*\)$/\2/'`
case $maybe_os in
nto-qnx* | linux-gnu* | linux-android* | linux-dietlibc | linux-
newlib* | \
linux-uclibc* | uclinux-uclibc* | uclinux-gnu* | kfreebsd*-gnu* | \
knetbsd*-gnu* | netbsd*-gnu* | \
kopensolaris*-gnu* | \
storm-chaos* | os2-emx* | rtmk-nova*)
    os=-$maybe_os
    basic_machine=`echo $1 | sed 's/^\(.*\) - \([^-]*-[-]*\)$/\1/'`
    ;;
android-linux)
    os=-linux-android
    basic_machine=`echo $1 | sed 's/^\(.*\) - \([^-]*-[-]*\)$/\1/'`-
unknown
    ;;
*)
    basic_machine=`echo $1 | sed 's/-[-]*$//'`
    if [ $basic_machine != $1 ]
    then os=`echo $1 | sed 's/.*-/-/'`
    else os=; fi
    ;;

```

```

esac

### Let's recognize common machines as not being operating systems so
### that things like config.sub decstation-3100 work. We also
### recognize some manufacturers as not being operating systems, so we
### can provide default operating systems below.
case $os in
    -sun*os*)
        # Prevent following clause from handling this invalid
input.
        ;;
    -dec* | -mips* | -sequent* | -encore* | -pc532* | -sgi* | -sony*
| \
    -att* | -7300* | -3300* | -delta* | -motorola* | -sun[234]* | \
    -unicom* | -ibm* | -next | -hp | -isi* | -apollo | -altos* | \
    -convergent* | -ncr* | -news | -32* | -3600* | -3100* | -hitachi*
| \
tti* | \
    -c[123]* | -convex* | -sun | -crds | -omron* | -dg | -ultra | -
| \
    -harris | -dolphin | -highlevel | -gould | -cbm | -ns | -masscomp
| \
    -apple | -axis | -knuth | -cray | -microblaze)
        os=
        basic_machine=$1
        ;;
    -bluegene*)
        os=-cnk
        ;;
    -sim | -cisco | -oki | -wec | -winbond)
        os=
        basic_machine=$1
        ;;
    -scout)
        ;;
    -wrs)
        os=-vxworks
        basic_machine=$1
        ;;
    -chorusos*)
        os=-chorusos
        basic_machine=$1
        ;;
    -chorusrdb)
        os=-chorusrdb
        basic_machine=$1
        ;;
    -hiux*)
        os=-hiuxwe2
        ;;
    -sco6)
        os=-sco5v6
        basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`

```



```

;;
-sco5)
  os=-sco3.2v5
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco4)
  os=-sco3.2v4
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco3.2.[4-9]*)
  os=`echo $os | sed -e 's/sco3.2./sco3.2v/'`
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco3.2v[4-9]*)
  # Don't forget version if it is 3.2v4 or newer.
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco5v6*)
  # Don't forget version if it is 3.2v4 or newer.
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco*)
  os=-sco3.2v2
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-udk*)
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-isc)
  os=-isc2.2
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-clix*)
  basic_machine=clipper-intergraph
;;
-isc*)
  basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-lynx*178)
  os=-lynxos178
;;
-lynx*5)
  os=-lynxos5
;;
-lynx*)
  os=-lynxos
;;
-ptx*)
  basic_machine=`echo $1 | sed -e 's/86-./86-sequent/'`
;;
-windowsnt*)
  os=`echo $os | sed -e 's/windowsnt/winnt/'`

```

```

        ;;
-psos*)
    os=-psos
        ;;
-mint | -mint[0-9]*)
    basic_machine=m68k-atari
    os=-mint
        ;;
esac

# Decode aliases for certain CPU-COMPANY combinations.
case $basic_machine in
    # Recognize the basic CPU types without company name.
    # Some are omitted here because they have special meanings below.
    1750a | 580 \
    | a29k \
    | aarch64 | aarch64_be \
    | alpha | alphaev[4-8] | alphaev56 | alphaev6[78] | alphapca5[67]
\
    | alpha64 | alpha64ev[4-8] | alpha64ev56 | alpha64ev6[78] |
alpha64pca5[67] \
    | am33_2.0 \
    | arc | arm | arm[b]le | arme[lb] | armv[2345] | armv[345][lb] |
avr | avr32 \
    | be32 | be64 \
    | bfin \
    | c4x | clipper \
    | d10v | d30v | dlx | dsp16xx \
    | epiphany \
    | fido | fr30 | frv \
    | h8300 | h8500 | hppa | hppa1.[01] | hppa2.0 | hppa2.0[nw] |
hppa64 \
    | hexagon \
    | i370 | i860 | i960 | ia64 \
    | ip2k | iq2000 \
    | le32 | le64 \
    | lm32 \
    | m32c | m32r | m32rle | m68000 | m68k | m88k \
    | maxq | mb | microblaze | mcore | mep | metag \
    | mips | mipsbe | mipseb | mipsel | mipsle \
    | mips16 \
    | mips64 | mips64el \
    | mips64octeon | mips64octeonel \
    | mips64orion | mips64orionel \
    | mips64r5900 | mips64r5900el \
    | mips64vr | mips64vrel \
    | mips64vr4100 | mips64vr4100el \
    | mips64vr4300 | mips64vr4300el \
    | mips64vr5000 | mips64vr5000el \
    | mips64vr5900 | mips64vr5900el \
    | mipsisa32 | mipsisa32el \
    | mipsisa32r2 | mipsisa32r2el \

```

```

| mipsisa64 | mipsisa64el \
| mipsisa64r2 | mipsisa64r2el \
| mipsisa64sb1 | mipsisa64sb1el \
| mipsisa64sr71k | mipsisa64sr71kel \
| mipstx39 | mipstx39el \
| mn10200 | mn10300 \
| moxie \
| mt \
| msp430 \
| nds32 | nds32le | nds32be \
| nios | nios2 \
| ns16k | ns32k \
| open8 \
| or32 \
| pdp10 | pdp11 | pj | pj1 \
| powerpc | powerpc64 | powerpc64le | powerpcle \
| pyramid \
| rl78 | rx \
| score \
| sh | sh[1234] | sh[24]a | sh[24]aeb | sh[23]e | sh[34]eb | sheb
| shbe | shle | sh[1234]le | sh3ele \
| sh64 | sh64le \
| sparc | sparc64 | sparc64b | sparc64v | sparc86x | sparclite |
sparclite \
| sparcv8 | sparcv9 | sparcv9b | sparcv9v \
| spu \
| tahoe | tic4x | tic54x | tic55x | tic6x | tic80 | tron \
| ubicom32 \
| v850 | v850e | v850e1 | v850e2 | v850es | v850e2v3 \
| we32k \
| x86 | xc16x | xstormy16 | xtensa \
| z8k | z80)
    basic_machine=$basic_machine-unknown
    ;;
c54x)
    basic_machine=tic54x-unknown
    ;;
c55x)
    basic_machine=tic55x-unknown
    ;;
c6x)
    basic_machine=tic6x-unknown
    ;;
m6811 | m68hc11 | m6812 | m68hc12 | m68hcs12x | picochip)
    basic_machine=$basic_machine-unknown
    os=-none
    ;;
m88110 | m680[12346]0 | m683?2 | m68360 | m5200 | v70 | w65 |
z8k)
    ;;
ms1)
    basic_machine=mt-unknown

```

```

        ;;

strongarm | thumb | xscale)
    basic_machine=arm-unknown
    ;;
xgate)
    basic_machine=$basic_machine-unknown
    os=-none
    ;;
xscaleeel)
    basic_machine=armel-unknown
    ;;

xscaleeel)
    basic_machine=armel-unknown
    ;;

# We use `pc' rather than `unknown'
# because (1) that's what they normally are, and
# (2) the word "unknown" tends to confuse beginning users.
i*86 | x86_64)
    basic_machine=$basic_machine-pc
    ;;
# Object if more than one company name word.
*-*-* )
    echo Invalid configuration \`${1}\': machine
\`${basic_machine}\` not recognized 1>&2
    exit 1
    ;;

# Recognize the basic CPU types with company name.
580-* \
| a29k-* \
| aarch64-* | aarch64_be-* \
| alpha-* | alphaev[4-8]-* | alphaev56-* | alphaev6[78]-* \
| alpha64-* | alpha64ev[4-8]-* | alpha64ev56-* | alpha64ev6[78]-* \
| alphapca5[67]-* | alpha64pca5[67]-* | arc-* \
| arm-* | armbe-* | armle-* | armb-* | armv*-* \
| avr-* | avr32-* \
| be32-* | be64-* \
| bfin-* | bs2000-* \
| c[123]* | c30-* | [cjt]90-* | c4x-* \
| clipper-* | craynv-* | cydra-* \
| d10v-* | d30v-* | dlx-* \
| elxsi-* \
| f30[01]-* | f700-* | fido-* | fr30-* | frv-* | fx80-* \
| h8300-* | h8500-* \
| hppa-* | hppa1.[01]-* | hppa2.0-* | hppa2.0[nw]-* | hppa64-* \
| hexagon-* \
| i*86-* | i860-* | i960-* | ia64-* \
| ip2k-* | iq2000-* \
| le32-* | le64-* \

```

```

| lm32-* \
| m32c-* | m32r-* | m32rle-* \
| m68000-* | m680[012346]0-* | m68360-* | m683?2-* | m68k-* \
| m88110-* | m88k-* | maxq-* | mcore-* | metag-* | microblaze-* \
| mips-* | mipsbe-* | mipseb-* | mipsel-* | mipsle-* \
| mips16-* \
| mips64-* | mips64el-* \
| mips64octeon-* | mips64octeonel-* \
| mips64orion-* | mips64orionel-* \
| mips64r5900-* | mips64r5900el-* \
| mips64vr-* | mips64vrel-* \
| mips64vr4100-* | mips64vr4100el-* \
| mips64vr4300-* | mips64vr4300el-* \
| mips64vr5000-* | mips64vr5000el-* \
| mips64vr5900-* | mips64vr5900el-* \
| mipsisa32-* | mipsisa32el-* \
| mipsisa32r2-* | mipsisa32r2el-* \
| mipsisa64-* | mipsisa64el-* \
| mipsisa64r2-* | mipsisa64r2el-* \
| mipsisa64sb1-* | mipsisa64sb1el-* \
| mipsisa64sr71k-* | mipsisa64sr71kel-* \
| mipstx39-* | mipstx39el-* \
| mmix-* \
| mt-* \
| msp430-* \
| nds32-* | nds32le-* | nds32be-* \
| nios-* | nios2-* \
| none-* | np1-* | ns16k-* | ns32k-* \
| open8-* \
| orion-* \
| pdp10-* | pdp11-* | pj-* | pj1-* | pn-* | power-* \
| powerpc-* | powerpc64-* | powerpc64le-* | powerpcle-* \
| pyramid-* \
| rl78-* | romp-* | rs6000-* | rx-* \
| sh-* | sh[1234]-* | sh[24]a-* | sh[24]aeb-* | sh[23]e-* |
sh[34]eb-* | sheb-* | shbe-* \
| shle-* | sh[1234]le-* | sh3ele-* | sh64-* | sh64le-* \
| sparc-* | sparc64-* | sparc64b-* | sparc64v-* | sparc86x-* |
sparclet-* \
| sparclite-* \
| sparcv8-* | sparcv9-* | sparcv9b-* | sparcv9v-* | sv1-* | sx?-*
\
| tahoe-* \
| tic30-* | tic4x-* | tic54x-* | tic55x-* | tic6x-* | tic80-* \
| tile*-* \
| tron-* \
| ubicom32-* \
| v850-* | v850e-* | v850e1-* | v850es-* | v850e2-* | v850e2v3-*
\
| vax-* \
| we32k-* \
| x86-* | x86_64-* | xc16x-* | xps100-* \

```

```

| xstormy16-* | xtensa*-* \
| ymp-* \
| z8k-* | z80-*)
    ;;
# Recognize the basic CPU types without company name, with glob
match.
xtensa*)
    basic_machine=$basic_machine-unknown
    ;;
# Recognize the various machine names and aliases which stand
# for a CPU type and a company and sometimes even an OS.
386bsd)
    basic_machine=i386-unknown
    os=-bsd
    ;;
3b1 | 7300 | 7300-att | att-7300 | pc7300 | safari | unixpc)
    basic_machine=m68000-att
    ;;
3b*)
    basic_machine=we32k-att
    ;;
a29khif)
    basic_machine=a29k-amd
    os=-udi
    ;;
abacus)
    basic_machine=abacus-unknown
    ;;
adobe68k)
    basic_machine=m68010-adobe
    os=-scout
    ;;
alliant | fx80)
    basic_machine=fx80-alliant
    ;;
altos | altos3068)
    basic_machine=m68k-altos
    ;;
am29k)
    basic_machine=a29k-none
    os=-bsd
    ;;
amd64)
    basic_machine=x86_64-pc
    ;;
amd64-*)
    basic_machine=x86_64-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
amdahl)
    basic_machine=580-amdahl
    os=-sysv

```

```

        ;;
amiga | amiga-*)
    basic_machine=m68k-unknown
    ;;
amigaos | amigados)
    basic_machine=m68k-unknown
    os=-amigaos
    ;;
amigaunix | amix)
    basic_machine=m68k-unknown
    os=-sysv4
    ;;
apollo68)
    basic_machine=m68k-apollo
    os=-sysv
    ;;
apollo68bsd)
    basic_machine=m68k-apollo
    os=-bsd
    ;;
aros)
    basic_machine=i386-pc
    os=-aros
    ;;
aux)
    basic_machine=m68k-apple
    os=-aux
    ;;
balance)
    basic_machine=ns32k-sequent
    os=-dynix
    ;;
blackfin)
    basic_machine=bfin-unknown
    os=-linux
    ;;
blackfin-*)
    basic_machine=bfin-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    os=-linux
    ;;
bluegene*)
    basic_machine=powerpc-ibm
    os=-cnk
    ;;
c54x-*)
    basic_machine=tic54x-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
c55x-*)
    basic_machine=tic55x-`echo $basic_machine | sed 's/^[^-]*-
//'\`

```

```

        ;;
c6x-*)
    basic_machine=tic6x-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
c90)
    basic_machine=c90-cray
    os=-unicos
        ;;
cegcc)
    basic_machine=arm-unknown
    os=-cegcc
        ;;
convex-c1)
    basic_machine=c1-convex
    os=-bsd
        ;;
convex-c2)
    basic_machine=c2-convex
    os=-bsd
        ;;
convex-c32)
    basic_machine=c32-convex
    os=-bsd
        ;;
convex-c34)
    basic_machine=c34-convex
    os=-bsd
        ;;
convex-c38)
    basic_machine=c38-convex
    os=-bsd
        ;;
cray | j90)
    basic_machine=j90-cray
    os=-unicos
        ;;
craynv)
    basic_machine=craynv-cray
    os=-unicosmp
        ;;
cr16 | cr16-*)
    basic_machine=cr16-unknown
    os=-elf
        ;;
crds | unos)
    basic_machine=m68k-crds
        ;;
crisv32 | crisv32-* | etraxfs*)
    basic_machine=crisv32-axis
        ;;
cris | cris-* | etrax*)

```



```

        basic_machine=cris-axis
        ;;
crx)
        basic_machine=crx-unknown
        os=-elf
        ;;
da30 | da30-*)
        basic_machine=m68k-da30
        ;;
decstation | decstation-3100 | pmax | pmax-* | pmin | dec3100 |
decstatn)
        basic_machine=mips-dec
        ;;
decsystem10* | dec10*)
        basic_machine=pdp10-dec
        os=-tops10
        ;;
decsystem20* | dec20*)
        basic_machine=pdp10-dec
        os=-tops20
        ;;
delta | 3300 | motorola-3300 | motorola-delta \
| 3300-motorola | delta-motorola)
        basic_machine=m68k-motorola
        ;;
delta88)
        basic_machine=m88k-motorola
        os=-sysv3
        ;;
dicos)
        basic_machine=i686-pc
        os=-dicos
        ;;
djgpp)
        basic_machine=i586-pc
        os=-msdosdjgpp
        ;;
dpx20 | dpx20-*)
        basic_machine=rs6000-bull
        os=-bosx
        ;;
dpx2* | dpx2*-bull)
        basic_machine=m68k-bull
        os=-sysv3
        ;;
ebmon29k)
        basic_machine=a29k-amd
        os=-ebmon
        ;;
elxsi)
        basic_machine=elxsi-elxsi
        os=-bsd

```

```
;;
encore | umax | mmax)
    basic_machine=ns32k-encore
;;
es1800 | OSE68k | ose68k | ose | OSE)
    basic_machine=m68k-ericsson
    os=-ose
;;
fx2800)
    basic_machine=i860-alliant
;;
genix)
    basic_machine=ns32k-ns
;;
gmicro)
    basic_machine=tron-gmicro
    os=-sysv
;;
go32)
    basic_machine=i386-pc
    os=-go32
;;
h3050r* | hiux*)
    basic_machine=hppa1.1-hitachi
    os=-hiuxwe2
;;
h8300hms)
    basic_machine=h8300-hitachi
    os=-hms
;;
h8300xray)
    basic_machine=h8300-hitachi
    os=-xray
;;
h8500hms)
    basic_machine=h8500-hitachi
    os=-hms
;;
harris)
    basic_machine=m88k-harris
    os=-sysv3
;;
hp300-*)
    basic_machine=m68k-hp
;;
hp300bsd)
    basic_machine=m68k-hp
    os=-bsd
;;
hp300hpux)
    basic_machine=m68k-hp
    os=-hpux
```

```

;;
hp3k9[0-9][0-9] | hp9[0-9][0-9])
    basic_machine=hppa1.0-hp
;;
hp9k2[0-9][0-9] | hp9k31[0-9])
    basic_machine=m68000-hp
;;
hp9k3[2-9][0-9])
    basic_machine=m68k-hp
;;
hp9k6[0-9][0-9] | hp6[0-9][0-9])
    basic_machine=hppa1.0-hp
;;
hp9k7[0-79][0-9] | hp7[0-79][0-9])
    basic_machine=hppa1.1-hp
;;
hp9k78[0-9] | hp78[0-9])
    # FIXME: really hppa2.0-hp
    basic_machine=hppa1.1-hp
;;
hp9k8[67]1 | hp8[67]1 | hp9k80[24] | hp80[24] | hp9k8[78]9 |
hp8[78]9 | hp9k893 | hp893)
    # FIXME: really hppa2.0-hp
    basic_machine=hppa1.1-hp
;;
hp9k8[0-9][13679] | hp8[0-9][13679])
    basic_machine=hppa1.1-hp
;;
hp9k8[0-9][0-9] | hp8[0-9][0-9])
    basic_machine=hppa1.0-hp
;;
hppa-next)
    os=-nextstep3
;;
hppaosf)
    basic_machine=hppa1.1-hp
    os=-osf
;;
hppro)
    basic_machine=hppa1.1-hp
    os=-proelf
;;
i370-ibm* | ibm*)
    basic_machine=i370-ibm
;;
i*86v32)
    basic_machine=`echo $1 | sed -e 's/86./86-pc/'`
    os=-sysv32
;;
i*86v4*)
    basic_machine=`echo $1 | sed -e 's/86./86-pc/'`
    os=-sysv4

```

```

        ;;
i*86v)
    basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
    os=-sysv
        ;;
i*86sol2)
    basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
    os=-solaris2
        ;;
i386mach)
    basic_machine=i386-mach
    os=-mach
        ;;
i386-vsta | vsta)
    basic_machine=i386-unknown
    os=-vsta
        ;;
iris | iris4d)
    basic_machine=mips-sgi
    case $os in
        -irix*)
            ;;
        *)
            os=-irix4
            ;;
    esac
        ;;
isi68 | isi)
    basic_machine=m68k-isi
    os=-sysv
        ;;
m68knommu)
    basic_machine=m68k-unknown
    os=-linux
        ;;
m68knommu-*)
    basic_machine=m68k-`echo $basic_machine | sed 's/^[^-]*-
//'`
    os=-linux
        ;;
m88k-omron*)
    basic_machine=m88k-omron
        ;;
magnum | m3230)
    basic_machine=mips-mips
    os=-sysv
        ;;
merlin)
    basic_machine=ns32k-utek
    os=-sysv
        ;;
microblaze)

```

```

        basic_machine=microblaze-xilinx
        ;;
mingw64)
        basic_machine=x86_64-pc
        os=-mingw64
        ;;
mingw32)
        basic_machine=i386-pc
        os=-mingw32
        ;;
mingw32ce)
        basic_machine=arm-unknown
        os=-mingw32ce
        ;;
miniframe)
        basic_machine=m68000-convergent
        ;;
*mint | -mint[0-9]* | *MiNT | *MiNT[0-9]*)
        basic_machine=m68k-atari
        os=-mint
        ;;
mips3*~*)
        basic_machine=`echo $basic_machine | sed -e
's/mips3/mips64/'`
        ;;
mips3*)
        basic_machine=`echo $basic_machine | sed -e
's/mips3/mips64/'`-unknown
        ;;
monitor)
        basic_machine=m68k-rom68k
        os=-coff
        ;;
morphos)
        basic_machine=powerpc-unknown
        os=-morphos
        ;;
msdos)
        basic_machine=i386-pc
        os=-msdos
        ;;
ms1~*)
        basic_machine=`echo $basic_machine | sed -e 's/ms1-/mt-/'`
        ;;
msys)
        basic_machine=i386-pc
        os=-msys
        ;;
mvs)
        basic_machine=i370-ibm
        os=-mvs
        ;;

```

```

nacl)
    basic_machine=le32-unknown
    os=-nacl
    ;;
ncr3000)
    basic_machine=i486-ncr
    os=-sysv4
    ;;
netbsd386)
    basic_machine=i386-unknown
    os=-netbsd
    ;;
netwinder)
    basic_machine=armv4l-rebel
    os=-linux
    ;;
news | news700 | news800 | news900)
    basic_machine=m68k-sony
    os=-newsos
    ;;
news1000)
    basic_machine=m68030-sony
    os=-newsos
    ;;
news-3600 | risc-news)
    basic_machine=mips-sony
    os=-newsos
    ;;
necv70)
    basic_machine=v70-nec
    os=-sysv
    ;;
next | m*-next )
    basic_machine=m68k-next
    case $os in
        -nextstep* )
            ;;
        -ns2*)
            os=-nextstep2
            ;;
        *)
            os=-nextstep3
            ;;
    esac
    ;;
nh3000)
    basic_machine=m68k-harris
    os=-cxux
    ;;
nh[45]000)
    basic_machine=m88k-harris
    os=-cxux

```

```
;;
nindy960)
    basic_machine=i960-intel
    os=-nindy
;;
mon960)
    basic_machine=i960-intel
    os=-mon960
;;
nonstopux)
    basic_machine=mips-compaq
    os=-nonstopux
;;
np1)
    basic_machine=np1-gould
;;
neo-tandem)
    basic_machine=neo-tandem
;;
nse-tandem)
    basic_machine=nse-tandem
;;
nsr-tandem)
    basic_machine=nsr-tandem
;;
op50n-* | op60c-*)
    basic_machine=hppa1.1-oki
    os=-proelf
;;
openrisc | openrisc-*)
    basic_machine=or32-unknown
;;
os400)
    basic_machine=powerpc-ibm
    os=-os400
;;
OSE68000 | ose68000)
    basic_machine=m68000-ericsson
    os=-ose
;;
os68k)
    basic_machine=m68k-none
    os=-os68k
;;
pa-hitachi)
    basic_machine=hppa1.1-hitachi
    os=-hiuxwe2
;;
paragon)
    basic_machine=i860-intel
    os=-osf
;;
```

```

parisc)
    basic_machine=hppa-unknown
    os=-linux
    ;;
parisc-*)
    basic_machine=hppa-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    os=-linux
    ;;
pbd)
    basic_machine=sparc-tti
    ;;
pbb)
    basic_machine=m68k-tti
    ;;
pc532 | pc532-*)
    basic_machine=ns32k-pc532
    ;;
pc98)
    basic_machine=i386-pc
    ;;
pc98-*)
    basic_machine=i386-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
pentium | p5 | k5 | k6 | nexgen | viac3)
    basic_machine=i586-pc
    ;;
pentiumpro | p6 | 6x86 | athlon | athlon_*)
    basic_machine=i686-pc
    ;;
pentiumii | pentium2 | pentiumiii | pentium3)
    basic_machine=i686-pc
    ;;
pentium4)
    basic_machine=i786-pc
    ;;
pentium-* | p5-* | k5-* | k6-* | nexgen-* | viac3-*)
    basic_machine=i586-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
pentiumpro-* | p6-* | 6x86-* | athlon-*)
    basic_machine=i686-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
pentiumii-* | pentium2-* | pentiumiii-* | pentium3-*)
    basic_machine=i686-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
pentium4-*)
    basic_machine=i786-`echo $basic_machine | sed 's/^[^-]*-
//'\`

```



```

        ;;
    pn)
        basic_machine=pn-gould
        ;;
    power)
        basic_machine=power-ibm
        ;;
    ppc | ppcbe)
        basic_machine=powerpc-unknown
        ;;
    ppc-* | ppcbe-*)
        basic_machine=powerpc-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
    ppcle | powerpclittle | ppc-le | powerpc-little)
        basic_machine=powerpcle-unknown
        ;;
    ppcle-* | powerpclittle-*)
        basic_machine=powerpcle-`echo $basic_machine | sed 's/^[^-
]*-//'\`
        ;;
    ppc64)
        basic_machine=powerpc64-unknown
        ;;
    ppc64-*)
        basic_machine=powerpc64-`echo $basic_machine | sed
's/^[^-]*-//'\`
        ;;
    ppc64le | powerpc64little | ppc64-le | powerpc64-little)
        basic_machine=powerpc64le-unknown
        ;;
    ppc64le-* | powerpc64little-*)
        basic_machine=powerpc64le-`echo $basic_machine | sed
's/^[^-]*-//'\`
        ;;
    ps2)
        basic_machine=i386-ibm
        ;;
    pw32)
        basic_machine=i586-unknown
        os=-pw32
        ;;
    rdos)
        basic_machine=i386-pc
        os=-rdos
        ;;
    rom68k)
        basic_machine=m68k-rom68k
        os=-coff
        ;;
    rm[46]00)
        basic_machine=mips-siemens
        ;;
    rtpc | rtpc-*)
        basic_machine=romp-ibm
        ;;

```

```
s390 | s390-*)
    basic_machine=s390-ibm
    ;;
s390x | s390x-*)
    basic_machine=s390x-ibm
    ;;
sa29200)
    basic_machine=a29k-amd
    os=-udi
    ;;
sb1)
    basic_machine=mipsisa64sb1-unknown
    ;;
sb1el)
    basic_machine=mipsisa64sb1el-unknown
    ;;
sde)
    basic_machine=mipsisa32-sde
    os=-elf
    ;;
sei)
    basic_machine=mips-sei
    os=-seix
    ;;
sequent)
    basic_machine=i386-sequent
    ;;
sh)
    basic_machine=sh-hitachi
    os=-hms
    ;;
sh5el)
    basic_machine=sh5le-unknown
    ;;
sh64)
    basic_machine=sh64-unknown
    ;;
sparclite-wrs | simso-wrs)
    basic_machine=sparclite-wrs
    os=-vxworks
    ;;
sps7)
    basic_machine=m68k-bull
    os=-sysv2
    ;;
spur)
    basic_machine=spur-unknown
    ;;
st2000)
    basic_machine=m68k-tandem
    ;;
stratus)
```

```
        basic_machine=i860-stratus
        os=-sysv4
        ;;
strongarm-* | thumb-*)
        basic_machine=arm-`echo $basic_machine | sed 's/^[^-]*-//'`
        ;;
sun2)
        basic_machine=m68000-sun
        ;;
sun2os3)
        basic_machine=m68000-sun
        os=-sunos3
        ;;
sun2os4)
        basic_machine=m68000-sun
        os=-sunos4
        ;;
sun3os3)
        basic_machine=m68k-sun
        os=-sunos3
        ;;
sun3os4)
        basic_machine=m68k-sun
        os=-sunos4
        ;;
sun4os3)
        basic_machine=sparc-sun
        os=-sunos3
        ;;
sun4os4)
        basic_machine=sparc-sun
        os=-sunos4
        ;;
sun4sol2)
        basic_machine=sparc-sun
        os=-solaris2
        ;;
sun3 | sun3-*)
        basic_machine=m68k-sun
        ;;
sun4)
        basic_machine=sparc-sun
        ;;
sun386 | sun386i | roadrunner)
        basic_machine=i386-sun
        ;;
sv1)
        basic_machine=sv1-cray
        os=-unicos
        ;;
symmetry)
        basic_machine=i386-sequent
```

```
        os=-dynix
        ;;
t3e)
    basic_machine=alphaev5-cray
    os=-unicos
    ;;
t90)
    basic_machine=t90-cray
    os=-unicos
    ;;
tile*)
    basic_machine=$basic_machine-unknown
    os=-linux-gnu
    ;;
tx39)
    basic_machine=mipstx39-unknown
    ;;
tx39el)
    basic_machine=mipstx39el-unknown
    ;;
toad1)
    basic_machine=pdp10-xkl
    os=-tops20
    ;;
tower | tower-32)
    basic_machine=m68k-ncr
    ;;
tpf)
    basic_machine=s390x-ibm
    os=-tpf
    ;;
udi29k)
    basic_machine=a29k-amd
    os=-udi
    ;;
ultra3)
    basic_machine=a29k-nyu
    os=-sym1
    ;;
v810 | necv810)
    basic_machine=v810-nec
    os=-none
    ;;
vaxv)
    basic_machine=vax-dec
    os=-sysv
    ;;
vms)
    basic_machine=vax-dec
    os=-vms
    ;;
vpp* | vx | vx-*)
```

```

        basic_machine=f301-fujitsu
        ;;
vxworks960)
        basic_machine=i960-wrs
        os=-vxworks
        ;;
vxworks68)
        basic_machine=m68k-wrs
        os=-vxworks
        ;;
vxworks29k)
        basic_machine=a29k-wrs
        os=-vxworks
        ;;
w65*)
        basic_machine=w65-wdc
        os=-none
        ;;
w89k-*)
        basic_machine=hppa1.1-winbond
        os=-proelf
        ;;
xbox)
        basic_machine=i686-pc
        os=-mingw32
        ;;
xps | xps100)
        basic_machine=xps100-honeywell
        ;;
xscale-* | xscalee[bl]-*)
        basic_machine=`echo $basic_machine | sed 's/^xscale/arm/'`
        ;;
ymp)
        basic_machine=ymp-cray
        os=-unicos
        ;;
z8k-*--coff)
        basic_machine=z8k-unknown
        os=-sim
        ;;
z80-*--coff)
        basic_machine=z80-unknown
        os=-sim
        ;;
none)
        basic_machine=none-none
        os=-none
        ;;

```

```

# Here we handle the default manufacturer of certain CPU types. It is
in
# some cases the only manufacturer, in others, it is the most popular.

```

```
w89k)
    basic_machine=hppa1.1-winbond
    ;;
op50n)
    basic_machine=hppa1.1-okni
    ;;
op60c)
    basic_machine=hppa1.1-okni
    ;;
romp)
    basic_machine=romp-ibm
    ;;
mmix)
    basic_machine=mmix-knuth
    ;;
rs6000)
    basic_machine=rs6000-ibm
    ;;
vax)
    basic_machine=vax-dec
    ;;
pdp10)
    # there are many clones, so DEC is not a safe bet
    basic_machine=pdp10-unknown
    ;;
pdp11)
    basic_machine=pdp11-dec
    ;;
we32k)
    basic_machine=we32k-att
    ;;
sh[1234] | sh[24]a | sh[24]aeb | sh[34]eb | sh[1234]le |
sh[23]ele)
    basic_machine=sh-unknown
    ;;
sparc | sparcv8 | sparcv9 | sparcv9b | sparcv9v)
    basic_machine=sparc-sun
    ;;
cydra)
    basic_machine=cydra-cydrome
    ;;
orion)
    basic_machine=orion-highlevel
    ;;
orion105)
    basic_machine=clipper-highlevel
    ;;
mac | mpw | mac-mpw)
    basic_machine=m68k-apple
    ;;
pmac | pmac-mpw)
    basic_machine=powerpc-apple
```

```

        ;;
*-unknown)
    # Make sure to match an already-canonicalized machine name.
    ;;
*)
    echo Invalid configuration ``$1``: machine
``$basic_machine`` not recognized 1>&2
    exit 1
    ;;
esac

# Here we canonicalize certain aliases for manufacturers.
case $basic_machine in
    *-digital*)
        basic_machine=`echo $basic_machine | sed
's/digital.*/dec/'`
        ;;
    *-commodore*)
        basic_machine=`echo $basic_machine | sed
's/commodore.*/cbm/'`
        ;;
*)
    ;;
esac

# Decode manufacturer-specific aliases for certain operating systems.

if [ x"$os" != x"" ]
then
case $os in
    # First match some system type aliases
    # that might get confused with valid system types.
    # -solaris* is a basic system type, with this one exception.
    -auroraux)
        os=-auroraux
        ;;
    -solaris1 | -solaris1.*)
        os=`echo $os | sed -e 's|solaris1|sunos4|'`
        ;;
    -solaris)
        os=-solaris2
        ;;
    -svr4*)
        os=-sysv4
        ;;
    -unixware*)
        os=-sysv4.2uw
        ;;
    -gnu/linux*)
        os=`echo $os | sed -e 's|gnu/linux|linux-gnu|'`
        ;;
    # First accept the basic system types.

```

```

# The portable systems comes first.
# Each alternative MUST END IN A *, to match a version number.
# -sysv* is not here because it comes later, after sysvr4.
-gnu* | -bsd* | -mach* | -minix* | -genix* | -ultrix* | -irix* \
| -*vms* | -sco* | -esix* | -isc* | -aix* | -cnk* | -sunos
| -sunos[34]*\
-solaris* \
| -hpux* | -unos* | -osf* | -luna* | -dgux* | -auroraux* |
| -sym* | -kopensolaris* \
-aof* \
| -amigaos* | -amigados* | -msdos* | -newsos* | -unicos* |
| -aos* | -aros* \
| -nindy* | -vxsim* | -vxworks* | -ebmon* | -hms* | -mvs* \
| -clix* | -riscos* | -uniplus* | -iris* | -rtu* | -xenix*
\
| -hiux* | -386bsd* | -knetbsd* | -mirbsd* | -netbsd* \
| -bitrig* | -openbsd* | -solidbsd* \
| -ekkobsd* | -kfreebsd* | -freebsd* | -riscix* | -lynxos*
\
| -bosx* | -nextstep* | -cxux* | -aout* | -elf* | -oabi* \
| -ptx* | -coff* | -ecoff* | -winnt* | -domain* | -vsta* \
| -udi* | -eabi* | -lites* | -ieee* | -go32* | -aux* \
| -chorusos* | -chorusrdb* | -cegcc* \
| -cygwin* | -msys* | -pe* | -psos* | -moss* | -proelf* | -
rtems* \
| -mingw32* | -mingw64* | -linux-gnu* | -linux-android* \
| -linux-newlib* | -linux-uclibc* \
| -uxpv* | -beos* | -mpeix* | -udk* \
| -interix* | -uwin* | -mks* | -rhapsody* | -darwin* | -
opened* \
| -openstep* | -oskit* | -conix* | -pw32* | -nonstopux* \
| -storm-chaos* | -tops10* | -tenex* | -tops20* | -its* \
| -os2* | -vos* | -palmos* | -uclinux* | -nucleus* \
| -morphos* | -superux* | -rtmk* | -rtmk-nova* | -windiss*
\
| -powermax* | -dnix* | -nx6 | -nx7 | -sei* | -dragonfly* \
| -skyos* | -haiku* | -rdos* | -toppers* | -drops* | -es*)
# Remember, each alternative MUST END IN *, to match a version
number.
;;
-qnx*)
    case $basic_machine in
        x86-* | i*86-*)
            ;;
        *)
            os=-nto$os
            ;;
    esac
;;
-nto-qnx*)
    ;;
-nto*)

```



```

        os=`echo $os | sed -e 's|nto|nto-qnx|'`
        ;;
    -sim | -es1800* | -hms* | -xray | -os68k* | -none* | -v88r* \
        | -windows* | -osx | -abug | -netware* | -os9* | -beos* | -
haiku* \
        | -macos* | -mpw* | -magic* | -mmixware* | -mon960* | -
lnews*)
        ;;
    -mac*)
        os=`echo $os | sed -e 's|mac|macos|'`
        ;;
    -linux-dietlibc)
        os=-linux-dietlibc
        ;;
    -linux*)
        os=`echo $os | sed -e 's|linux|linux-gnu|'`
        ;;
    -sunos5*)
        os=`echo $os | sed -e 's|sunos5|solaris2|'`
        ;;
    -sunos6*)
        os=`echo $os | sed -e 's|sunos6|solaris3|'`
        ;;
    -opened*)
        os=-openedition
        ;;
    -os400*)
        os=-os400
        ;;
    -wince*)
        os=-wince
        ;;
    -osfrose*)
        os=-osfrose
        ;;
    -osf*)
        os=-osf
        ;;
    -utek*)
        os=-bsd
        ;;
    -dynix*)
        os=-bsd
        ;;
    -acis*)
        os=-aos
        ;;
    -atheos*)
        os=-atheos
        ;;
    -syllable*)
        os=-syllable

```

```
;;
-386bsd)
    os=-bsd
;;
-ctix* | -uts*)
    os=-sysv
;;
-nova*)
    os=-rtmk-nova
;;
-ns2 )
    os=-nextstep2
;;
-nsk*)
    os=-nsk
;;
# Preserve the version number of sinix5.
-sinix5.*)
    os=`echo $os | sed -e 's|sinix|sysv|`
;;
-sinix*)
    os=-sysv4
;;
-tpf*)
    os=-tpf
;;
-triton*)
    os=-sysv3
;;
-oss*)
    os=-sysv3
;;
-svr4)
    os=-sysv4
;;
-svr3)
    os=-sysv3
;;
-sysvr4)
    os=-sysv4
;;
# This must come after -sysvr4.
-sysv*)
    ;;
-ose*)
    os=-ose
    ;;
-es1800*)
    os=-ose
    ;;
-xenix)
    os=-xenix
```

```

        ;;
        -*mint | -mint[0-9]* | -*MiNT | -MiNT[0-9]*)
            os=-mint
            ;;
        -aros*)
            os=-aros
            ;;
        -kaos*)
            os=-kaos
            ;;
        -zvmoe)
            os=-zvmoe
            ;;
        -dicos*)
            os=-dicos
            ;;
        -nacl*)
            ;;
        -none)
            ;;
        *)
            # Get rid of the '-' at the beginning of $os.
            os=`echo $os | sed 's/[^-]*-//'\`
            echo Invalid configuration ``$1\`: system ``$os\` not
recognized 1>&2
            exit 1
            ;;
esac
else

# Here we handle the default operating systems that come with various
machines.
# The value should be what the vendor currently ships out the door
with their
# machine or put another way, the most popular os provided with the
machine.

# Note that if you're going to try to match "-MANUFACTURER" here (say,
# "-sun"), then you have to tell the case statement up towards the top
# that MANUFACTURER isn't an operating system. Otherwise, code above
# will signal an error saying that MANUFACTURER isn't an operating
# system, and we'll never get to this point.

case $basic_machine in
    score-*)
        os=-elf
        ;;
    spu-*)
        os=-elf
        ;;
    *-acorn)
        os=-riscix1.2

```

```
    ;;
arm*-rebel)
    os=-linux
    ;;
arm*-semi)
    os=-aout
    ;;
c4x-* | tic4x-*)
    os=-coff
    ;;
hexagon-*)
    os=-elf
    ;;
tic54x-*)
    os=-coff
    ;;
tic55x-*)
    os=-coff
    ;;
tic6x-*)
    os=-coff
    ;;
# This must come before the *-dec entry.
pdp10-*)
    os=-tops20
    ;;
pdp11-*)
    os=-none
    ;;
*-dec | vax-*)
    os=-ultrix4.2
    ;;
m68*-apollo)
    os=-domain
    ;;
i386-sun)
    os=-sunos4.0.2
    ;;
m68000-sun)
    os=-sunos3
    ;;
m68*-cisco)
    os=-aout
    ;;
mep-*)
    os=-elf
    ;;
mips*-cisco)
    os=-elf
    ;;
mips*-*)
    os=-elf
```

```
    ;;
or32-*)
    os=-coff
    ;;
*-tti)      # must be before sparc entry or we get the wrong os.
    os=-sysv3
    ;;
sparc-* | *-sun)
    os=-sunos4.1.1
    ;;
*-be)
    os=-beos
    ;;
*-haiku)
    os=-haiku
    ;;
*-ibm)
    os=-aix
    ;;
*-knuth)
    os=-mmixware
    ;;
*-wec)
    os=-proelf
    ;;
*-winbond)
    os=-proelf
    ;;
*-oki)
    os=-proelf
    ;;
*-hp)
    os=-hpux
    ;;
*-hitachi)
    os=-hiux
    ;;
i860-* | *-att | *-ncr | *-altos | *-motorola | *-convergent)
    os=-sysv
    ;;
*-cbm)
    os=-amigaos
    ;;
*-dg)
    os=-dgux
    ;;
*-dolphin)
    os=-sysv3
    ;;
m68k-ccur)
    os=-rtu
    ;;
```

```
m88k-omron*)
    os=-luna
    ;;
*-next )
    os=-nextstep
    ;;
*-sequent)
    os=-ptx
    ;;
*-crds)
    os=-unos
    ;;
*-ns)
    os=-genix
    ;;
i370-*)
    os=-mvs
    ;;
*-next)
    os=-nextstep3
    ;;
*-gould)
    os=-sysv
    ;;
*-highlevel)
    os=-bsd
    ;;
*-encore)
    os=-bsd
    ;;
*-sgi)
    os=-irix
    ;;
*-siemens)
    os=-sysv4
    ;;
*-masscomp)
    os=-rtu
    ;;
f30[01]-fujitsu | f700-fujitsu)
    os=-uxpv
    ;;
*-rom68k)
    os=-coff
    ;;
*-*bug)
    os=-coff
    ;;
*-apple)
    os=-macos
    ;;
*-atari*)
```

```
        os=-mint
        ;;
*)
        os=-none
        ;;
esac
fi
```

Here we handle the case where we know the os, and the CPU type, but not the

manufacturer. We pick the logical manufacturer.

vendor=unknown

```
case $basic_machine in
*-unknown)
        case $os in
                -riscix*)
                        vendor=acorn
                        ;;
                -sunos*)
                        vendor=sun
                        ;;
                -cnk*|-aix*)
                        vendor=ibm
                        ;;
                -beos*)
                        vendor=be
                        ;;
                -hpux*)
                        vendor=hp
                        ;;
                -mpeix*)
                        vendor=hp
                        ;;
                -hiux*)
                        vendor=hitachi
                        ;;
                -unos*)
                        vendor=crds
                        ;;
                -dgux*)
                        vendor=dg
                        ;;
                -luna*)
                        vendor=omron
                        ;;
                -genix*)
                        vendor=ns
                        ;;
                -mvs* | -opened*)
                        vendor=ibm
                        ;;
                -os400*)
```

```

        vendor=ibm
        ;;
-ptx*)
        vendor=sequent
        ;;
-tpf*)
        vendor=ibm
        ;;
-vxsim* | -vxworks* | -windiss*)
        vendor=wrs
        ;;
-aux*)
        vendor=apple
        ;;
-hms*)
        vendor=hitachi
        ;;
-mpw* | -macos*)
        vendor=apple
        ;;
-*mint | -mint[0-9]* | -*MiNT | -MiNT[0-9]*)
        vendor=atari
        ;;
-vos*)
        vendor=stratus
        ;;
    esac
    basic_machine=`echo $basic_machine | sed
"s/unknown/$vendor/"`
    ;;
esac

echo $basic_machine$os
exit

# Local variables:
# eval: (add-hook 'write-file-hooks 'time-stamp)
# time-stamp-start: "timestamp="
# time-stamp-format: "%:y-%02m-%02d"
# time-stamp-end: ""
# End:

File = config.sub.~1~

#!/bin/sh
# Configuration validation subroutine script.
# Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999,
# 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
# 2011, 2012 Free Software Foundation, Inc.

```



```
timestamp='2012-07-31'
```

```
# This file is (in principle) common to ALL GNU software.
# The presence of a machine in this file suggests that SOME GNU
software
# can handle that machine. It does not imply ALL GNU software can.
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, see <http://www.gnu.org/licenses/>.
#
# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program that contains a
# configuration script generated by Autoconf, you may include it under
# the same distribution terms that you use for the rest of that
program.
```

```
# Please send patches to <config-patches@gnu.org>. Submit a context
# diff and a properly formatted GNU ChangeLog entry.
#
# Configuration subroutine to validate and canonicalize a
configuration type.
# Supply the specified configuration type as an argument.
# If it is invalid, we print an error message on stderr and exit with
code 1.
# Otherwise, we print the canonical config type on stdout and succeed.
```

```
# You can get the latest version of this script from:
#
http://git.savannah.gnu.org/gitweb/?p=config.git;a=blob\_plain;f=config
.sub;hb=HEAD
```

```
# This file is supposed to be the same for all GNU packages
# and recognize all the CPU types, system types and aliases
# that are meaningful with *any* GNU software.
# Each package is responsible for reporting which valid configurations
# it does not support. The user should be able to distinguish
# a failure to support a valid configuration from a meaningless
# configuration.
```

```
# The goal of this file is to map all the various variations of a
given
```

```

# machine specification into a single specification in the form:
#   CPU_TYPE-MANUFACTURER-OPERATING_SYSTEM
# or in some cases, the newer four-part form:
#   CPU_TYPE-MANUFACTURER-KERNEL-OPERATING_SYSTEM
# It is wrong to echo any other type of specification.

me=`echo "$0" | sed -e 's,./,,'`

usage="\
Usage: $0 [OPTION] CPU-MFR-OPSYS
      $0 [OPTION] ALIAS

Canonicalize a configuration name.

Operation modes:
  -h, --help           print this help, then exit
  -t, --time-stamp     print date of last modification, then exit
  -v, --version        print version number, then exit

Report bugs and patches to <config-patches@gnu.org>."

version="\
GNU config.sub ($timestamp)

Copyright (C) 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000,
2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012
Free Software Foundation, Inc.

This is free software; see the source for copying conditions.  There
is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE."

help="
Try `"$me" --help' for more information."

# Parse command line
while test $# -gt 0 ; do
  case $1 in
    --time-stamp | --time* | -t )
      echo "$timestamp" ; exit ;;
    --version | -v )
      echo "$version" ; exit ;;
    --help | --h* | -h )
      echo "$usage"; exit ;;
    -- )
      # Stop option processing
      shift; break ;;
    - )
      # Use stdin as input.
      break ;;
    -* )
      echo "$me: invalid option $1$help"
      exit 1 ;;
  esac
done

```

```

    *local*)
        # First pass through any local machine types.
        echo $1
        exit ;;

    * )
        break ;;
esac
done

case $# in
    0) echo "$me: missing argument$help" >&2
        exit 1;;
    1) ;;
    *) echo "$me: too many arguments$help" >&2
        exit 1;;
esac

# Separate what the user gave into CPU-COMPANY and OS or KERNEL-OS (if
any).
# Here we must recognize all the valid KERNEL-OS combinations.
maybe_os=`echo $1 | sed 's/^\(.*\) - \([^-]*-[^-]*\)$/\2/'`
case $maybe_os in
    nto-qnx* | linux-gnu* | linux-android* | linux-dietlibc | linux-
newlib* | \
linux-uclibc* | uclinux-uclibc* | uclinux-gnu* | kfreebsd*-gnu* | \
knetbsd*-gnu* | netbsd*-gnu* | \
kopensolaris*-gnu* | \
storm-chaos* | os2-emx* | rtmk-nova*)
    os=-$maybe_os
    basic_machine=`echo $1 | sed 's/^\(.*\) - \([^-]*-[^-]*\)$/\1/'`
    ;;
    android-linux)
    os=-linux-android
    basic_machine=`echo $1 | sed 's/^\(.*\) - \([^-]*-[^-]*\)$/\1/'`-
unknown
    ;;
    *)
    basic_machine=`echo $1 | sed 's/-[^-]*$//`
    if [ $basic_machine != $1 ]
    then os=`echo $1 | sed 's/.*-/-/'`
    else os=; fi
    ;;
esac

### Let's recognize common machines as not being operating systems so
### that things like config.sub decstation-3100 work. We also
### recognize some manufacturers as not being operating systems, so we
### can provide default operating systems below.
case $os in
    -sun*os*)

```

```

        # Prevent following clause from handling this invalid
input.
        ;;
        -dec* | -mips* | -sequent* | -encore* | -pc532* | -sgi* | -sony*
| \
        -att* | -7300* | -3300* | -delta* | -motorola* | -sun[234]* | \
        -unicom* | -ibm* | -next | -hp | -isi* | -apollo | -altos* | \
        -convergent* | -ncr* | -news | -32* | -3600* | -3100* | -hitachi*
|\
        -c[123]* | -convex* | -sun | -crds | -omron* | -dg | -ultra | -
tti* | \
        -harris | -dolphin | -highlevel | -gould | -cbm | -ns | -masscomp
| \
        -apple | -axis | -knuth | -cray | -microblaze)
            os=
            basic_machine=$1
            ;;
        -bluegene*)
            os=-cnk
            ;;
        -sim | -cisco | -oki | -wec | -winbond)
            os=
            basic_machine=$1
            ;;
        -scout)
            ;;
        -wrs)
            os=-vxworks
            basic_machine=$1
            ;;
        -chorusos*)
            os=-chorusos
            basic_machine=$1
            ;;
        -chorusrdb)
            os=-chorusrdb
            basic_machine=$1
            ;;
        -hiux*)
            os=-hiuxwe2
            ;;
        -sco6)
            os=-sco5v6
            basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
            ;;
        -sco5)
            os=-sco3.2v5
            basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
            ;;
        -sco4)
            os=-sco3.2v4
            basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`

```

```

;;
-sco3.2.[4-9]*)
    os=`echo $os | sed -e 's/sco3.2./sco3.2v/'`
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco3.2v[4-9]*)
    # Don't forget version if it is 3.2v4 or newer.
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco5v6*)
    # Don't forget version if it is 3.2v4 or newer.
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-sco*)
    os=-sco3.2v2
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-udk*)
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-isc)
    os=-isc2.2
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-clix*)
    basic_machine=clipper-intergraph
;;
-isc*)
    basic_machine=`echo $1 | sed -e 's/86-./86-pc/'`
;;
-lynx*178)
    os=-lynxos178
;;
-lynx*5)
    os=-lynxos5
;;
-lynx*)
    os=-lynxos
;;
-ptx*)
    basic_machine=`echo $1 | sed -e 's/86-./86-sequent/'`
;;
-windowsnt*)
    os=`echo $os | sed -e 's/windowsnt/winnt/'`
;;
-psos*)
    os=-psos
;;
-mint | -mint[0-9]*)
    basic_machine=m68k-atari
    os=-mint
;;

```

```

esac

# Decode aliases for certain CPU-COMPANY combinations.
case $basic_machine in
    # Recognize the basic CPU types without company name.
    # Some are omitted here because they have special meanings below.
    1750a | 580 \
    | a29k \
    | aarch64 | aarch64_be \
    | alpha | alphaev[4-8] | alphaev56 | alphaev6[78] | alphapca5[67]
\
    | alpha64 | alpha64ev[4-8] | alpha64ev56 | alpha64ev6[78] |
alpha64pca5[67] \
    | am33_2.0 \
    | arc | arm | arm[bl]e | arme[lb] | armv[2345] | armv[345][lb] |
avr | avr32 \
    | be32 | be64 \
    | bfin \
    | c4x | clipper \
    | dl0v | d30v | dlx | dsp16xx \
    | epiphany \
    | fido | fr30 | frv \
    | h8300 | h8500 | hppa | hppa1.[01] | hppa2.0 | hppa2.0[nw] |
hppa64 \
    | hexagon \
    | i370 | i860 | i960 | ia64 \
    | ip2k | iq2000 \
    | le32 | le64 \
    | lm32 \
    | m32c | m32r | m32rle | m68000 | m68k | m88k \
    | maxq | mb | microblaze | mcore | mep | metag \
    | mips | mipsbe | mipseb | mipsel | mipsle \
    | mips16 \
    | mips64 | mips64el \
    | mips64octeon | mips64octeonel \
    | mips64orion | mips64orionel \
    | mips64r5900 | mips64r5900el \
    | mips64vr | mips64vrel \
    | mips64vr4100 | mips64vr4100el \
    | mips64vr4300 | mips64vr4300el \
    | mips64vr5000 | mips64vr5000el \
    | mips64vr5900 | mips64vr5900el \
    | mipsisa32 | mipsisa32el \
    | mipsisa32r2 | mipsisa32r2el \
    | mipsisa64 | mipsisa64el \
    | mipsisa64r2 | mipsisa64r2el \
    | mipsisa64sb1 | mipsisa64sb1el \
    | mipsisa64sr71k | mipsisa64sr71kel \
    | mipstx39 | mipstx39el \
    | mn10200 | mn10300 \
    | moxie \
    | mt \

```

```

| msp430 \
| nds32 | nds32le | nds32be \
| nios | nios2 \
| ns16k | ns32k \
| open8 \
| or32 \
| pdp10 | pdp11 | pj | pj1 \
| powerpc | powerpc64 | powerpc64le | powerpcle \
| pyramid \
| rl78 | rx \
| score \
| sh | sh[1234] | sh[24]a | sh[24]aeb | sh[23]e | sh[34]eb | sheb
| shbe | shle | sh[1234]le | sh3ele \
| sh64 | sh64le \
| sparc | sparc64 | sparc64b | sparc64v | sparc86x | sparclite |
sparclite \
| sparcv8 | sparcv9 | sparcv9b | sparcv9v \
| spu \
| tahoe | tic4x | tic54x | tic55x | tic6x | tic80 | tron \
| ubicom32 \
| v850 | v850e | v850e1 | v850e2 | v850es | v850e2v3 \
| we32k \
| x86 | xc16x | xstormy16 | xtensa \
| z8k | z80)
    basic_machine=$basic_machine-unknown
    ;;
c54x)
    basic_machine=tic54x-unknown
    ;;
c55x)
    basic_machine=tic55x-unknown
    ;;
c6x)
    basic_machine=tic6x-unknown
    ;;
m6811 | m68hc11 | m6812 | m68hc12 | m68hcs12x | picochip)
    basic_machine=$basic_machine-unknown
    os=-none
    ;;
m88110 | m680[12346]0 | m683?2 | m68360 | m5200 | v70 | w65 |
z8k)
    ;;
ms1)
    basic_machine=mt-unknown
    ;;

strongarm | thumb | xscale)
    basic_machine=arm-unknown
    ;;
xgate)
    basic_machine=$basic_machine-unknown
    os=-none

```

```

;;
xscaleeb)
    basic_machine=armeb-unknown
;;

xscaleel)
    basic_machine=armel-unknown
;;

# We use `pc' rather than `unknown'
# because (1) that's what they normally are, and
# (2) the word "unknown" tends to confuse beginning users.
i*86 | x86_64)
    basic_machine=$basic_machine-pc
;;
# Object if more than one company name word.
*-*-* )
    echo Invalid configuration \`${1}`: machine
\`${basic_machine}` not recognized 1>&2
    exit 1
;;

# Recognize the basic CPU types with company name.
580-* \
| a29k-* \
| aarch64-* | aarch64_be-* \
| alpha-* | alphaev[4-8]-* | alphaev56-* | alphaev6[78]-* \
| alpha64-* | alpha64ev[4-8]-* | alpha64ev56-* | alpha64ev6[78]-*

| alphapca5[67]-* | alpha64pca5[67]-* | arc-* \
| arm-* | armbel-* | armle-* | armbel-* | armv*-* \
| avr-* | avr32-* \
| be32-* | be64-* \
| bfin-* | bs2000-* \
| c[123]* | c30-* | [cjt]90-* | c4x-* \
| clipper-* | craynv-* | cydra-* \
| dl0v-* | d30v-* | dlx-* \
| elxsi-* \
| f30[01]-* | f700-* | fido-* | fr30-* | frv-* | fx80-* \
| h8300-* | h8500-* \
| hppa-* | hppa1.[01]-* | hppa2.0-* | hppa2.0[nw]-* | hppa64-* \
| hexagon-* \
| i*86-* | i860-* | i960-* | ia64-* \
| ip2k-* | iq2000-* \
| le32-* | le64-* \
| lm32-* \
| m32c-* | m32r-* | m32rle-* \
| m68000-* | m680[012346]0-* | m68360-* | m683?2-* | m68k-* \
| m88110-* | m88k-* | maxq-* | mcore-* | metag-* | microblaze-* \
| mips-* | mipsbe-* | mipseb-* | mipsel-* | mipsle-* \
| mips16-* \
| mips64-* | mips64el-* \
| mips64octeon-* | mips64octeonel-* \

```



```

| mips64orion-* | mips64orionel-* \
| mips64r5900-* | mips64r5900el-* \
| mips64vr-* | mips64vrel-* \
| mips64vr4100-* | mips64vr4100el-* \
| mips64vr4300-* | mips64vr4300el-* \
| mips64vr5000-* | mips64vr5000el-* \
| mips64vr5900-* | mips64vr5900el-* \
| mipsisa32-* | mipsisa32el-* \
| mipsisa32r2-* | mipsisa32r2el-* \
| mipsisa64-* | mipsisa64el-* \
| mipsisa64r2-* | mipsisa64r2el-* \
| mipsisa64sb1-* | mipsisa64sb1el-* \
| mipsisa64sr71k-* | mipsisa64sr71kel-* \
| mipstx39-* | mipstx39el-* \
| mmix-* \
| mt-* \
| msp430-* \
| nds32-* | nds32le-* | nds32be-* \
| nios-* | nios2-* \
| none-* | np1-* | ns16k-* | ns32k-* \
| open8-* \
| orion-* \
| pdp10-* | pdp11-* | pj-* | pj1-* | pn-* | power-* \
| powerpc-* | powerpc64-* | powerpc64le-* | powerpcle-* \
| pyramid-* \
| rl78-* | romp-* | rs6000-* | rx-* \
| sh-* | sh[1234]-* | sh[24]a-* | sh[24]aeb-* | sh[23]e-* |
sh[34]eb-* | sheb-* | shbe-* \
| shle-* | sh[1234]le-* | sh3ele-* | sh64-* | sh64le-* \
| sparc-* | sparc64-* | sparc64b-* | sparc64v-* | sparc86x-* |
sparclet-* \
| sparclite-* \
| sparcv8-* | sparcv9-* | sparcv9b-* | sparcv9v-* | sv1-* | sx?-*
\
| tahoe-* \
| tic30-* | tic4x-* | tic54x-* | tic55x-* | tic6x-* | tic80-* \
| tile*-* \
| tron-* \
| ubicom32-* \
| v850-* | v850e-* | v850el-* | v850es-* | v850e2-* | v850e2v3-*
\
| vax-* \
| we32k-* \
| x86-* | x86_64-* | xc16x-* | xps100-* \
| xstormy16-* | xtensa*-* \
| ymp-* \
| z8k-* | z80-*)
;;
# Recognize the basic CPU types without company name, with glob
match.
xtensa*)
    basic_machine=$basic_machine-unknown

```

```

        ;;
# Recognize the various machine names and aliases which stand
# for a CPU type and a company and sometimes even an OS.
386bsd)
    basic_machine=i386-unknown
    os=-bsd
    ;;
3b1 | 7300 | 7300-att | att-7300 | pc7300 | safari | unixpc)
    basic_machine=m68000-att
    ;;
3b*)
    basic_machine=we32k-att
    ;;
a29khif)
    basic_machine=a29k-amd
    os=-udi
    ;;
abacus)
    basic_machine=abacus-unknown
    ;;
adobe68k)
    basic_machine=m68010-adobe
    os=-scout
    ;;
alliant | fx80)
    basic_machine=fx80-alliant
    ;;
altos | altos3068)
    basic_machine=m68k-altos
    ;;
am29k)
    basic_machine=a29k-none
    os=-bsd
    ;;
amd64)
    basic_machine=x86_64-pc
    ;;
amd64-*)
    basic_machine=x86_64-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
amdahl)
    basic_machine=580-amdahl
    os=-sysv
    ;;
amiga | amiga-*)
    basic_machine=m68k-unknown
    ;;
amigaos | amigados)
    basic_machine=m68k-unknown
    os=-amigaos
    ;;

```

```

amigaunix | amix)
    basic_machine=m68k-unknown
    os=-sysv4
    ;;
apollo68)
    basic_machine=m68k-apollo
    os=-sysv
    ;;
apollo68bsd)
    basic_machine=m68k-apollo
    os=-bsd
    ;;
aros)
    basic_machine=i386-pc
    os=-aros
    ;;
aux)
    basic_machine=m68k-apple
    os=-aux
    ;;
balance)
    basic_machine=ns32k-sequent
    os=-dynix
    ;;
blackfin)
    basic_machine=bfin-unknown
    os=-linux
    ;;
blackfin-*)
    basic_machine=bfin-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    os=-linux
    ;;
bluegene*)
    basic_machine=powerpc-ibm
    os=-cnk
    ;;
c54x-*)
    basic_machine=tic54x-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
c55x-*)
    basic_machine=tic55x-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
c6x-*)
    basic_machine=tic6x-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
c90)
    basic_machine=c90-cray
    os=-unicos

```

```
;;
cegcc)
    basic_machine=arm-unknown
    os=-cegcc
;;
convex-c1)
    basic_machine=c1-convex
    os=-bsd
;;
convex-c2)
    basic_machine=c2-convex
    os=-bsd
;;
convex-c32)
    basic_machine=c32-convex
    os=-bsd
;;
convex-c34)
    basic_machine=c34-convex
    os=-bsd
;;
convex-c38)
    basic_machine=c38-convex
    os=-bsd
;;
cray | j90)
    basic_machine=j90-cray
    os=-unicos
;;
craynv)
    basic_machine=craynv-cray
    os=-unicosmp
;;
cr16 | cr16-*)
    basic_machine=cr16-unknown
    os=-elf
;;
crds | unos)
    basic_machine=m68k-crds
;;
crisv32 | crisv32-* | etraxfs*)
    basic_machine=crisv32-axis
;;
cris | cris-* | etrax*)
    basic_machine=cris-axis
;;
crx)
    basic_machine=crx-unknown
    os=-elf
;;
da30 | da30-*)
    basic_machine=m68k-da30
```

```

;;
decstation | decstation-3100 | pmax | pmax-* | pmin | dec3100 |
decstatn)
    basic_machine=mips-dec
;;
decsystem10* | dec10*)
    basic_machine=pdp10-dec
    os=-tops10
;;
decsystem20* | dec20*)
    basic_machine=pdp10-dec
    os=-tops20
;;
delta | 3300 | motorola-3300 | motorola-delta \
| 3300-motorola | delta-motorola)
    basic_machine=m68k-motorola
;;
delta88)
    basic_machine=m88k-motorola
    os=-sysv3
;;
dicos)
    basic_machine=i686-pc
    os=-dicos
;;
djgpp)
    basic_machine=i586-pc
    os=-msdosdjgpp
;;
dpx20 | dpx20-*)
    basic_machine=rs6000-bull
    os=-bosx
;;
dpx2* | dpx2*-bull)
    basic_machine=m68k-bull
    os=-sysv3
;;
ebmon29k)
    basic_machine=a29k-amd
    os=-ebmon
;;
elxsi)
    basic_machine=elxsi-elxsi
    os=-bsd
;;
encore | umax | mmax)
    basic_machine=ns32k-encore
;;
es1800 | OSE68k | ose68k | ose | OSE)
    basic_machine=m68k-ericsson
    os=-ose
;;

```

```
fx2800)
    basic_machine=i860-alliant
    ;;
genix)
    basic_machine=ns32k-ns
    ;;
gmicro)
    basic_machine=tron-gmicro
    os=-sysv
    ;;
go32)
    basic_machine=i386-pc
    os=-go32
    ;;
h3050r* | hiux*)
    basic_machine=hppa1.1-hitachi
    os=-hiuxwe2
    ;;
h8300hms)
    basic_machine=h8300-hitachi
    os=-hms
    ;;
h8300xray)
    basic_machine=h8300-hitachi
    os=-xray
    ;;
h8500hms)
    basic_machine=h8500-hitachi
    os=-hms
    ;;
harris)
    basic_machine=m88k-harris
    os=-sysv3
    ;;
hp300-*)
    basic_machine=m68k-hp
    ;;
hp300bsd)
    basic_machine=m68k-hp
    os=-bsd
    ;;
hp300hpux)
    basic_machine=m68k-hp
    os=-hpux
    ;;
hp3k9[0-9][0-9] | hp9[0-9][0-9])
    basic_machine=hppa1.0-hp
    ;;
hp9k2[0-9][0-9] | hp9k31[0-9])
    basic_machine=m68000-hp
    ;;
hp9k3[2-9][0-9])
```

```

        basic_machine=m68k-hp
        ;;
hp9k6[0-9][0-9] | hp6[0-9][0-9])
        basic_machine=hppa1.0-hp
        ;;
hp9k7[0-79][0-9] | hp7[0-79][0-9])
        basic_machine=hppa1.1-hp
        ;;
hp9k78[0-9] | hp78[0-9])
        # FIXME: really hppa2.0-hp
        basic_machine=hppa1.1-hp
        ;;
hp9k8[67]1 | hp8[67]1 | hp9k80[24] | hp80[24] | hp9k8[78]9 |
hp8[78]9 | hp9k893 | hp893)
        # FIXME: really hppa2.0-hp
        basic_machine=hppa1.1-hp
        ;;
hp9k8[0-9][13679] | hp8[0-9][13679])
        basic_machine=hppa1.1-hp
        ;;
hp9k8[0-9][0-9] | hp8[0-9][0-9])
        basic_machine=hppa1.0-hp
        ;;
hppa-next)
        os=-nextstep3
        ;;
hppaosf)
        basic_machine=hppa1.1-hp
        os=-osf
        ;;
hppro)
        basic_machine=hppa1.1-hp
        os=-proelf
        ;;
i370-ibm* | ibm*)
        basic_machine=i370-ibm
        ;;
i*86v32)
        basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
        os=-sysv32
        ;;
i*86v4*)
        basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
        os=-sysv4
        ;;
i*86v)
        basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
        os=-sysv
        ;;
i*86sol2)
        basic_machine=`echo $1 | sed -e 's/86.*/86-pc/'`
        os=-solaris2

```

```

        ;;
i386mach)
    basic_machine=i386-mach
    os=-mach
    ;;
i386-vsta | vsta)
    basic_machine=i386-unknown
    os=-vsta
    ;;
iris | iris4d)
    basic_machine=mips-sgi
    case $os in
        -irix*)
            ;;
        *)
            os=-irix4
            ;;
    esac
    ;;
isi68 | isi)
    basic_machine=m68k-isi
    os=-sysv
    ;;
m68knommu)
    basic_machine=m68k-unknown
    os=-linux
    ;;
m68knommu-*)
    basic_machine=m68k-`echo $basic_machine | sed 's/^[^-]*-
//'`
    os=-linux
    ;;
m88k-omron*)
    basic_machine=m88k-omron
    ;;
magnum | m3230)
    basic_machine=mips-mips
    os=-sysv
    ;;
merlin)
    basic_machine=ns32k-utek
    os=-sysv
    ;;
microblaze)
    basic_machine=microblaze-xilinx
    ;;
mingw64)
    basic_machine=x86_64-pc
    os=-mingw64
    ;;
mingw32)
    basic_machine=i386-pc

```



```

        os=-mingw32
        ;;
mingw32ce)
    basic_machine=arm-unknown
    os=-mingw32ce
    ;;
miniframe)
    basic_machine=m68000-convergent
    ;;
*mint | -mint[0-9]* | *MiNT | *MiNT[0-9]*)
    basic_machine=m68k-atari
    os=-mint
    ;;
mips3*-*)
    basic_machine=`echo $basic_machine | sed -e
's/mips3/mips64/'`
    ;;
mips3*)
    basic_machine=`echo $basic_machine | sed -e
's/mips3/mips64/'`-unknown
    ;;
monitor)
    basic_machine=m68k-rom68k
    os=-c0ff
    ;;
morphos)
    basic_machine=powerpc-unknown
    os=-morphos
    ;;
msdos)
    basic_machine=i386-pc
    os=-msdos
    ;;
ms1-*)
    basic_machine=`echo $basic_machine | sed -e 's/ms1-/mt-/'`
    ;;
msys)
    basic_machine=i386-pc
    os=-msys
    ;;
mvs)
    basic_machine=i370-ibm
    os=-mvs
    ;;
nacl)
    basic_machine=le32-unknown
    os=-nacl
    ;;
ncr3000)
    basic_machine=i486-ncr
    os=-sysv4
    ;;

```

```

netbsd386)
    basic_machine=i386-unknown
    os=-netbsd
    ;;
netwinder)
    basic_machine=armv4l-rebel
    os=-linux
    ;;
news | news700 | news800 | news900)
    basic_machine=m68k-sony
    os=-newsos
    ;;
news1000)
    basic_machine=m68030-sony
    os=-newsos
    ;;
news-3600 | risc-news)
    basic_machine=mips-sony
    os=-newsos
    ;;
necv70)
    basic_machine=v70-nec
    os=-sysv
    ;;
next | m*-next )
    basic_machine=m68k-next
    case $os in
        -nextstep* )
            ;;
        -ns2*)
            os=-nextstep2
            ;;
        *)
            os=-nextstep3
            ;;
    esac
    ;;
nh3000)
    basic_machine=m68k-harris
    os=-cxux
    ;;
nh[45]000)
    basic_machine=m88k-harris
    os=-cxux
    ;;
nindy960)
    basic_machine=i960-intel
    os=-nindy
    ;;
mon960)
    basic_machine=i960-intel
    os=-mon960

```

```

        ;;
nonstopux)
    basic_machine=mips-compaq
    os=-nonstopux
    ;;
np1)
    basic_machine=np1-gould
    ;;
neo-tandem)
    basic_machine=neo-tandem
    ;;
nse-tandem)
    basic_machine=nse-tandem
    ;;
nsr-tandem)
    basic_machine=nsr-tandem
    ;;
op50n-* | op60c-*)
    basic_machine=hppa1.1-oki
    os=-proelf
    ;;
openrisc | openrisc-*)
    basic_machine=or32-unknown
    ;;
os400)
    basic_machine=powerpc-ibm
    os=-os400
    ;;
OSE68000 | ose68000)
    basic_machine=m68000-ericsson
    os=-ose
    ;;
os68k)
    basic_machine=m68k-none
    os=-os68k
    ;;
pa-hitachi)
    basic_machine=hppa1.1-hitachi
    os=-hiuxwe2
    ;;
paragon)
    basic_machine=i860-intel
    os=-osf
    ;;
parisc)
    basic_machine=hppa-unknown
    os=-linux
    ;;
parisc-*)
    basic_machine=hppa-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    os=-linux

```

```

        ;;
pbd)
    basic_machine=sparc-tti
        ;;
pbb)
    basic_machine=m68k-tti
        ;;
pc532 | pc532-*)
    basic_machine=ns32k-pc532
        ;;
pc98)
    basic_machine=i386-pc
        ;;
pc98-*)
    basic_machine=i386-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
pentium | p5 | k5 | k6 | nexgen | viac3)
    basic_machine=i586-pc
        ;;
pentiumpro | p6 | 6x86 | athlon | athlon_*)
    basic_machine=i686-pc
        ;;
pentiumii | pentium2 | pentiumiii | pentium3)
    basic_machine=i686-pc
        ;;
pentium4)
    basic_machine=i786-pc
        ;;
pentium-* | p5-* | k5-* | k6-* | nexgen-* | viac3-*)
    basic_machine=i586-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
pentiumpro-* | p6-* | 6x86-* | athlon-*)
    basic_machine=i686-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
pentiumii-* | pentium2-* | pentiumiii-* | pentium3-*)
    basic_machine=i686-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
pentium4-*)
    basic_machine=i786-`echo $basic_machine | sed 's/^[^-]*-
//'\`
        ;;
pn)
    basic_machine=pn-gould
        ;;
power)
    basic_machine=power-ibm
        ;;
ppc | ppcbe)
    basic_machine=powerpc-unknown
        ;;

```

```

ppc-* | ppcbe-*)
    basic_machine=powerpc-`echo $basic_machine | sed 's/^[^-]*-
//'\`
    ;;
ppcle | powerpclittle | ppc-le | powerpc-little)
    basic_machine=powerpcle-unknown
    ;;
ppcle-* | powerpclittle-*)
    basic_machine=powerpcle-`echo $basic_machine | sed 's/^[^-
]*-//'\`
    ;;
ppc64)
    basic_machine=powerpc64-unknown
    ;;
ppc64-*)
    basic_machine=powerpc64-`echo $basic_machine | sed
's/^[^-]*-//'\`
    ;;
ppc64le | powerpc64little | ppc64-le | powerpc64-little)
    basic_machine=powerpc64le-unknown
    ;;
ppc64le-* | powerpc64little-*)
    basic_machine=powerpc64le-`echo $basic_machine | sed
's/^[^-]*-//'\`
    ;;
ps2)
    basic_machine=i386-ibm
    ;;
pw32)
    basic_machine=i586-unknown
    os=-pw32
    ;;
rdos)
    basic_machine=i386-pc
    os=-rdos
    ;;
rom68k)
    basic_machine=m68k-rom68k
    os=-c0ff
    ;;
rm[46]00)
    basic_machine=mips-siemens
    ;;
rtpc | rtpc-*)
    basic_machine=romp-ibm
    ;;
s390 | s390-*)
    basic_machine=s390-ibm
    ;;
s390x | s390x-*)
    basic_machine=s390x-ibm
    ;;
sa29200)
    basic_machine=a29k-amd

```

```

        os=-udi
        ;;
sb1)
    basic_machine=mipsisa64sb1-unknown
    ;;
sb1el)
    basic_machine=mipsisa64sb1el-unknown
    ;;
sde)
    basic_machine=mipsisa32-sde
    os=-elf
    ;;
sei)
    basic_machine=mips-sei
    os=-seiux
    ;;
sequent)
    basic_machine=i386-sequent
    ;;
sh)
    basic_machine=sh-hitachi
    os=-hms
    ;;
sh5el)
    basic_machine=sh5le-unknown
    ;;
sh64)
    basic_machine=sh64-unknown
    ;;
sparclite-wrs | simso-wrs)
    basic_machine=sparclite-wrs
    os=-vxworks
    ;;
sps7)
    basic_machine=m68k-bull
    os=-sysv2
    ;;
spur)
    basic_machine=spur-unknown
    ;;
st2000)
    basic_machine=m68k-tandem
    ;;
stratus)
    basic_machine=i860-stratus
    os=-sysv4
    ;;
strongarm-* | thumb-*)
    basic_machine=arm-`echo $basic_machine | sed 's/^[^-]*-//'`
    ;;
sun2)
    basic_machine=m68000-sun

```

```
;;
sun2os3)
    basic_machine=m68000-sun
    os=-sunos3
;;
sun2os4)
    basic_machine=m68000-sun
    os=-sunos4
;;
sun3os3)
    basic_machine=m68k-sun
    os=-sunos3
;;
sun3os4)
    basic_machine=m68k-sun
    os=-sunos4
;;
sun4os3)
    basic_machine=sparc-sun
    os=-sunos3
;;
sun4os4)
    basic_machine=sparc-sun
    os=-sunos4
;;
sun4sol2)
    basic_machine=sparc-sun
    os=-solaris2
;;
sun3 | sun3-*)
    basic_machine=m68k-sun
;;
sun4)
    basic_machine=sparc-sun
;;
sun386 | sun386i | roadrunner)
    basic_machine=i386-sun
;;
sv1)
    basic_machine=sv1-cray
    os=-unicos
;;
symmetry)
    basic_machine=i386-sequent
    os=-dynix
;;
t3e)
    basic_machine=alphaev5-cray
    os=-unicos
;;
t90)
    basic_machine=t90-cray
```

```
        os=-unicos
        ;;
tile*)
    basic_machine=$basic_machine-unknown
    os=-linux-gnu
    ;;
tx39)
    basic_machine=mipstx39-unknown
    ;;
tx39el)
    basic_machine=mipstx39el-unknown
    ;;
toad1)
    basic_machine=pdp10-xkl
    os=-tops20
    ;;
tower | tower-32)
    basic_machine=m68k-ncr
    ;;
tpf)
    basic_machine=s390x-ibm
    os=-tpf
    ;;
udi29k)
    basic_machine=a29k-amd
    os=-udi
    ;;
ultra3)
    basic_machine=a29k-nyu
    os=-sym1
    ;;
v810 | necv810)
    basic_machine=v810-nec
    os=-none
    ;;
vaxv)
    basic_machine=vax-dec
    os=-sysv
    ;;
vms)
    basic_machine=vax-dec
    os=-vms
    ;;
vpp* | vx | vx-*)
    basic_machine=f301-fujitsu
    ;;
vxworks960)
    basic_machine=i960-wrs
    os=-vxworks
    ;;
vxworks68)
    basic_machine=m68k-wrs
```



```

        os=-vxworks
        ;;
vxworks29k)
    basic_machine=a29k-wrs
    os=-vxworks
    ;;
w65*)
    basic_machine=w65-wdc
    os=-none
    ;;
w89k-*)
    basic_machine=hppa1.1-winbond
    os=-proelf
    ;;
xbox)
    basic_machine=i686-pc
    os=-mingw32
    ;;
xps | xps100)
    basic_machine=xps100-honeywell
    ;;
xscale-* | xscalee[b1]-*)
    basic_machine=`echo $basic_machine | sed 's/^xscale/arm/'`
    ;;
ymp)
    basic_machine=ymp-cray
    os=-unicos
    ;;
z8k-*-coff)
    basic_machine=z8k-unknown
    os=-sim
    ;;
z80-*-coff)
    basic_machine=z80-unknown
    os=-sim
    ;;
none)
    basic_machine=none-none
    os=-none
    ;;

```

Here we handle the default manufacturer of certain CPU types. It is in

some cases the only manufacturer, in others, it is the most popular.

```

w89k)
    basic_machine=hppa1.1-winbond
    ;;
op50n)
    basic_machine=hppa1.1-oki
    ;;
op60c)
    basic_machine=hppa1.1-oki

```

```

        ;;
romp)
    basic_machine=romp-ibm
    ;;
mmix)
    basic_machine=mmix-knuth
    ;;
rs6000)
    basic_machine=rs6000-ibm
    ;;
vax)
    basic_machine=vax-dec
    ;;
pdp10)
    # there are many clones, so DEC is not a safe bet
    basic_machine=pdp10-unknown
    ;;
pdp11)
    basic_machine=pdp11-dec
    ;;
we32k)
    basic_machine=we32k-att
    ;;
sh[1234] | sh[24]a | sh[24]aeb | sh[34]eb | sh[1234]le |
sh[23]ele)
    basic_machine=sh-unknown
    ;;
sparc | sparcv8 | sparcv9 | sparcv9b | sparcv9v)
    basic_machine=sparc-sun
    ;;
cydra)
    basic_machine=cydra-cydrome
    ;;
orion)
    basic_machine=orion-highlevel
    ;;
orion105)
    basic_machine=clipper-highlevel
    ;;
mac | mpw | mac-mpw)
    basic_machine=m68k-apple
    ;;
pmac | pmac-mpw)
    basic_machine=powerpc-apple
    ;;
*-unknown)
    # Make sure to match an already-canonicalized machine name.
    ;;
*)
    echo Invalid configuration \`${1}\': machine
\`${basic_machine}\' not recognized 1>&2
    exit 1

```

```

                ;;
esac

# Here we canonicalize certain aliases for manufacturers.
case $basic_machine in
    *-digital*)
        basic_machine=`echo $basic_machine | sed
's/digital.*/dec/'`
                ;;
    *-commodore*)
        basic_machine=`echo $basic_machine | sed
's/commodore.*/cbm/'`
                ;;
    *)
                ;;
esac

# Decode manufacturer-specific aliases for certain operating systems.

if [ x"$os" != x"" ]
then
case $os in
    # First match some system type aliases
    # that might get confused with valid system types.
    # -solaris* is a basic system type, with this one exception.
    -auroraux)
        os=-auroraux
                ;;
    -solaris1 | -solaris1.*)
        os=`echo $os | sed -e 's|solaris1|sunos4|'`
                ;;
    -solaris)
        os=-solaris2
                ;;
    -svr4*)
        os=-sysv4
                ;;
    -unixware*)
        os=-sysv4.2uw
                ;;
    -gnu/linux*)
        os=`echo $os | sed -e 's|gnu/linux|linux-gnu|'`
                ;;
    # First accept the basic system types.
    # The portable systems comes first.
    # Each alternative MUST END IN A *, to match a version number.
    # -sysv* is not here because it comes later, after sysvr4.
    -gnu* | -bsd* | -mach* | -minix* | -genix* | -ultrix* | -irix* \
        | -*vms* | -sco* | -esix* | -isc* | -aix* | -cnk* | -sunos
| -sunos[34]*\
        | -hpux* | -unos* | -osf* | -luna* | -dgux* | -auroraux* |
-solaris* \

```

```

    | -sym* | -kopensolaris* \
    | -amigaos* | -amigados* | -msdos* | -newsos* | -unicos* |
-aof* \
    | -aos* | -aros* \
    | -nindy* | -vxsim* | -vxworks* | -ebmon* | -hms* | -mvs* \
    | -clix* | -riscos* | -uniplus* | -iris* | -rtu* | -xenix*
\
    | -hiux* | -386bsd* | -knetbsd* | -mirbsd* | -netbsd* \
    | -bitrig* | -openbsd* | -solidbsd* \
    | -ekkobsd* | -kfreebsd* | -freebsd* | -riscix* | -lynxos*
\
    | -bosx* | -nextstep* | -cxux* | -aout* | -elf* | -oabi* \
    | -ptx* | -coff* | -ecoff* | -winnt* | -domain* | -vsta* \
    | -udi* | -eabi* | -lites* | -ieee* | -go32* | -aux* \
    | -chorusos* | -chorusrdb* | -cegcc* \
rtems* \
    | -cygwin* | -msys* | -pe* | -psos* | -moss* | -proelf* | -
    | -mingw32* | -mingw64* | -linux-gnu* | -linux-android* \
    | -linux-newlib* | -linux-uclibc* \
    | -uxpv* | -beos* | -mpeix* | -udk* \
    | -interix* | -uwin* | -mks* | -rhapsody* | -darwin* | -
opened* \
    | -openstep* | -oskit* | -conix* | -pw32* | -nonstopux* \
    | -storm-chaos* | -tops10* | -tenex* | -tops20* | -its* \
    | -os2* | -vos* | -palmos* | -uclinux* | -nucleus* \
    | -morphos* | -superux* | -rtmk* | -rtmk-nova* | -windiss*
\
    | -powermax* | -dnix* | -nx6 | -nx7 | -sei* | -dragonfly* \
    | -skyos* | -haiku* | -rdos* | -toppers* | -drops* | -es*)
# Remember, each alternative MUST END IN *, to match a version
number.
;;
-qnx*)
    case $basic_machine in
        x86-* | i*86-*)
            ;;
        *)
            os=-nto$os
            ;;
    esac
;;
-nto-qnx*)
;;
-nto*)
    os=`echo $os | sed -e 's|nto|nto-qnx|'`
;;
-sim | -es1800* | -hms* | -xray | -os68k* | -none* | -v88r* \
| -windows* | -osx | -abug | -netware* | -os9* | -beos* | -
haiku* \
| -macos* | -mpw* | -magic* | -mmixware* | -mon960* | -
lnews*)
;;

```

```
-mac*)
    os=`echo $os | sed -e 's|mac|macos|'`
    ;;
-linux-dietlibc)
    os=-linux-dietlibc
    ;;
-linux*)
    os=`echo $os | sed -e 's|linux|linux-gnu|'`
    ;;
-sunos5*)
    os=`echo $os | sed -e 's|sunos5|solaris2|'`
    ;;
-sunos6*)
    os=`echo $os | sed -e 's|sunos6|solaris3|'`
    ;;
-opened*)
    os=-openedition
    ;;
-os400*)
    os=-os400
    ;;
-wince*)
    os=-wince
    ;;
-osfrose*)
    os=-osfrose
    ;;
-osf*)
    os=-osf
    ;;
-utek*)
    os=-bsd
    ;;
-dynix*)
    os=-bsd
    ;;
-acis*)
    os=-aos
    ;;
-atheos*)
    os=-atheos
    ;;
-syllable*)
    os=-syllable
    ;;
-386bsd)
    os=-bsd
    ;;
-ctix* | -uts*)
    os=-sysv
    ;;
-nova*)
```

```

        os=-rtmk-nova
        ;;
-ns2 )
        os=-nextstep2
        ;;
-nsk*)
        os=-nsk
        ;;
# Preserve the version number of sinix5.
-sinix5.*)
        os=`echo $os | sed -e 's|sinix|sysv|'`
        ;;
-sinix*)
        os=-sysv4
        ;;
-tpf*)
        os=-tpf
        ;;
-triton*)
        os=-sysv3
        ;;
-oss*)
        os=-sysv3
        ;;
-svr4)
        os=-sysv4
        ;;
-svr3)
        os=-sysv3
        ;;
-sysvr4)
        os=-sysv4
        ;;
# This must come after -sysvr4.
-sysv*)
        ;;
-ose*)
        os=-ose
        ;;
-es1800*)
        os=-ose
        ;;
-xenix)
        os=-xenix
        ;;
-*mint | -mint[0-9]* | -*MiNT | -MiNT[0-9]*)
        os=-mint
        ;;
-aros*)
        os=-aros
        ;;
-kaos*)

```

```

        os=-kaos
        ;;
-zvmoe)
        os=-zvmoe
        ;;
-dicos*)
        os=-dicos
        ;;
-nacl*)
        ;;
-none)
        ;;
*)
        # Get rid of the '-' at the beginning of $os.
        os=`echo $os | sed 's/[^-]*-//'\`
        echo Invalid configuration ``$1\`: system ``$os\` not
recognized 1>&2
        exit 1
        ;;
esac
else

# Here we handle the default operating systems that come with various
machines.
# The value should be what the vendor currently ships out the door
with their
# machine or put another way, the most popular os provided with the
machine.

# Note that if you're going to try to match "-MANUFACTURER" here (say,
# "-sun"), then you have to tell the case statement up towards the top
# that MANUFACTURER isn't an operating system. Otherwise, code above
# will signal an error saying that MANUFACTURER isn't an operating
# system, and we'll never get to this point.

case $basic_machine in
score-*)
        os=-elf
        ;;
spu-*)
        os=-elf
        ;;
*-acorn)
        os=-riscix1.2
        ;;
arm*-rebel)
        os=-linux
        ;;
arm*-semi)
        os=-aout
        ;;
c4x-* | tic4x-*)

```

```

        os=-coff
        ;;
hexagon-*)
        os=-elf
        ;;
tic54x-*)
        os=-coff
        ;;
tic55x-*)
        os=-coff
        ;;
tic6x-*)
        os=-coff
        ;;
# This must come before the *-dec entry.
pdp10-*)
        os=-tops20
        ;;
pdp11-*)
        os=-none
        ;;
*-dec | vax-*)
        os=-ultrix4.2
        ;;
m68*-apollo)
        os=-domain
        ;;
i386-sun)
        os=-sunos4.0.2
        ;;
m68000-sun)
        os=-sunos3
        ;;
m68*-cisco)
        os=-aout
        ;;
mep-*)
        os=-elf
        ;;
mips*-cisco)
        os=-elf
        ;;
mips*-*)
        os=-elf
        ;;
or32-*)
        os=-coff
        ;;
*-tti)      # must be before sparc entry or we get the wrong os.
        os=-sysv3
        ;;
sparc-* | *-sun)

```



```
        os=-sunos4.1.1
        ;;
*-be)
        os=-beos
        ;;
*-haiku)
        os=-haiku
        ;;
*-ibm)
        os=-aix
        ;;
*-knuth)
        os=-mmixware
        ;;
*-wec)
        os=-proelf
        ;;
*-winbond)
        os=-proelf
        ;;
*-oki)
        os=-proelf
        ;;
*-hp)
        os=-hpux
        ;;
*-hitachi)
        os=-hiux
        ;;
i860-* | *-att | *-ncr | *-altos | *-motorola | *-convergent)
        os=-sysv
        ;;
*-cbm)
        os=-amigaos
        ;;
*-dg)
        os=-dgux
        ;;
*-dolphin)
        os=-sysv3
        ;;
m68k-ccur)
        os=-rtu
        ;;
m88k-omron*)
        os=-luna
        ;;
*-next )
        os=-nextstep
        ;;
*-sequent)
        os=-ptx
```

```
        ;;
*-crds)
    os=-unos
        ;;
*-ns)
    os=-genix
        ;;
i370-*)
    os=-mvs
        ;;
*-next)
    os=-nextstep3
        ;;
*-gould)
    os=-sysv
        ;;
*-highlevel)
    os=-bsd
        ;;
*-encore)
    os=-bsd
        ;;
*-sgi)
    os=-irix
        ;;
*-siemens)
    os=-sysv4
        ;;
*-masscomp)
    os=-rtu
        ;;
f30[01]-fujitsu | f700-fujitsu)
    os=-uxpv
        ;;
*-rom68k)
    os=-coff
        ;;
*-*bug)
    os=-coff
        ;;
*-apple)
    os=-macos
        ;;
*-atari*)
    os=-mint
        ;;
*)
    os=-none
        ;;
esac
fi
```

```
# Here we handle the case where we know the os, and the CPU type, but
not the
# manufacturer. We pick the logical manufacturer.
vendor=unknown
case $basic_machine in
  *-unknown)
    case $os in
      -riscix*)
        vendor=acorn
        ;;
      -sunos*)
        vendor=sun
        ;;
      -cnk*|-aix*)
        vendor=ibm
        ;;
      -beos*)
        vendor=be
        ;;
      -hpux*)
        vendor=hp
        ;;
      -mpeix*)
        vendor=hp
        ;;
      -hiux*)
        vendor=hitachi
        ;;
      -unos*)
        vendor=crds
        ;;
      -dgux*)
        vendor=dg
        ;;
      -luna*)
        vendor=omron
        ;;
      -genix*)
        vendor=ns
        ;;
      -mvs* | -opened*)
        vendor=ibm
        ;;
      -os400*)
        vendor=ibm
        ;;
      -ptx*)
        vendor=sequent
        ;;
      -tpf*)
        vendor=ibm
        ;;
```

```

        -vxsim* | -vxworks* | -windiss*)
            vendor=wrs
            ;;
        -aux*)
            vendor=apple
            ;;
        -hms*)
            vendor=hitachi
            ;;
        -mpw* | -macos*)
            vendor=apple
            ;;
        -*mint | -mint[0-9]* | -*MiNT | -MiNT[0-9]*)
            vendor=atari
            ;;
        -vos*)
            vendor=stratus
            ;;
    esac
    basic_machine=`echo $basic_machine | sed
"s/unknown/$vendor/"`
    ;;
esac

echo $basic_machine$os
exit

# Local variables:
# eval: (add-hook 'write-file-hooks 'time-stamp)
# time-stamp-start: "timestamp="
# time-stamp-format: "%:y-%02m-%02d"
# time-stamp-end: ""
# End:

File = configure

#!/bin/sh
# Guess values for system-dependent variables and create Makefiles.
# Generated by GNU Autoconf 2.69 for dbus-glib 0.100.2.
#
# Report bugs to
<https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus&component=GLib>.
#
#
# Copyright (C) 1992-1996, 1998-2012 Free Software Foundation, Inc.
#
# This configure script is free software; the Free Software Foundation
# gives unlimited permission to copy, distribute and modify it.

```



```

        case $arg in #(
            *"$as_nl"*)
        expr "X$arg" : "X\\(.*\\)$as_nl";
        arg=`expr "X$arg" : ".*$as_nl\\(.*\\)"`;
        esac;
        expr "X$arg" : "X\\(.*\\)" | tr -d "$as_nl"
    ,

    export as_echo_n_body
    as_echo_n='sh -c $as_echo_n_body as_echo'
fi
export as_echo_body
as_echo='sh -c $as_echo_body as_echo'
fi

# The user is always right.
if test "${PATH_SEPARATOR+set}" != set; then
    PATH_SEPARATOR=:
    (PATH='/bin;/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 && {
        (PATH='/bin:/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 ||
            PATH_SEPARATOR=';'
    }
fi

# IFS
# We need space, tab and new line, in precisely that order. Quoting
is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " "$as_nl"

# Find who we are. Look in the path if we contain no directory
separator.
as_myself=
case $0 in #(
    *[\ \ /]* ) as_myself=$0 ;;
    *) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done
IFS=$as_save_IFS

;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
    as_myself=$0

```

```

fi
if test ! -f "$sas_myself"; then
    $sas_echo "$sas_myself: error: cannot find myself; rerun with an
absolute file name" >&2
    exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
# 1"
# suppresses any "Segmentation fault" message there. '(' could
# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH
do eval test x\${$as_var+set} = xset \
    && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

# NLS nuisances.
LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# Use a proper internal environment variable to ensure we don't fall
# into an infinite loop, continuously re-executing ourselves.
if test x"${_as_can_reexec}" != xno && test "x$CONFIG_SHELL" != x;
then
    _as_can_reexec=no; export _as_can_reexec;
    # We cannot yet assume a decent shell, so we have to provide a
# neutralization value for shells without unset; and this also
# works around shells that cannot unset nonexistent variables.
# Preserve -v and -x to the replacement shell.
BASH_ENV=/dev/null
ENV=/dev/null
(unset BASH_ENV) >/dev/null 2>&1 && unset BASH_ENV ENV
case $- in # (((
    *v*x* | *x*v* ) as_opts=-vx ;;
    *v* ) as_opts=-v ;;
    *x* ) as_opts=-x ;;
    * ) as_opts= ;;
esac
exec $CONFIG_SHELL $as_opts "$sas_myself" ${1+"$@"}
# Admittedly, this is quite paranoid, since all the known shells bail
# out after a failed `exec'.
$sas_echo "$0: could not re-execute with $CONFIG_SHELL" >&2

```

```

as_fn_exit 255
fi
# We don't want this to propagate to other subprocesses.
{ _as_can_reexec=; unset _as_can_reexec;}
if test "x$CONFIG_SHELL" = x; then
  as_bourne_compatible="if test -n \"\${ZSH_VERSION+set}\" && (emulate
sh) >/dev/null 2>&1; then :
  emulate sh
  NULLCMD=:
  # Pre-4.2 versions of Zsh do word splitting on \"${1+\"$@\"}\", which
  # is contrary to our usage. Disable this feature.
  alias -g \"\${1+\"$@\"}\"='\"$@\"'
  setopt NO_GLOB_SUBST
else
  case \"(set -o) 2>/dev/null\" in #(
  *posix*) :
    set -o posix ;; #(
  *) :
    ;;
esac
fi
"
  as_required="as_fn_return () { (exit \$1); }
as_fn_success () { as_fn_return 0; }
as_fn_failure () { as_fn_return 1; }
as_fn_ret_success () { return 0; }
as_fn_ret_failure () { return 1; }

exitcode=0
as_fn_success || { exitcode=1; echo as_fn_success failed.; }
as_fn_failure && { exitcode=1; echo as_fn_failure succeeded.; }
as_fn_ret_success || { exitcode=1; echo as_fn_ret_success failed.; }
as_fn_ret_failure && { exitcode=1; echo as_fn_ret_failure succeeded.; }
}
if ( set x; as_fn_ret_success y && test x = \"\$1\" ); then :

else
  exitcode=1; echo positional parameters were not saved.
fi
test x\$exitcode = x0 || exit 1
test -x / || exit 1"
  as_suggested="
as_lineno_1=";as_suggested=$as_suggested$LINENO;as_suggested=$as_sugge
sted" as_lineno_1a=\$LINENO

as_lineno_2=";as_suggested=$as_suggested$LINENO;as_suggested=$as_sugge
sted" as_lineno_2a=\$LINENO
  eval 'test \"x\$as_lineno_1'\$as_run'\" !=
\"x\$as_lineno_2'\$as_run'\" &&
  test \"x\`expr \$as_lineno_1'\$as_run' + 1\`\\" =
\"x\$as_lineno_2'\$as_run'\" || exit 1
test \"\$( ( 1 + 1 ) ) = 2 || exit 1

```



```

        if test "x$CONFIG_SHELL" != x; then :
export CONFIG_SHELL
        # We cannot yet assume a decent shell, so we have to
provide a
# neutralization value for shells without unset; and this also
# works around shells that cannot unset nonexistent variables.
# Preserve -v and -x to the replacement shell.
BASH_ENV=/dev/null
ENV=/dev/null
(unset BASH_ENV) >/dev/null 2>&1 && unset BASH_ENV ENV
case $- in # (((
*v*x* | *x*v* ) as_opts=-vx ;;
*v* ) as_opts=-v ;;
*x* ) as_opts=-x ;;
* ) as_opts= ;;
esac
exec $CONFIG_SHELL $as_opts "$as_myself" ${1+"$@"}
# Admittedly, this is quite paranoid, since all the known shells bail
# out after a failed `exec`.
$as_echo "$0: could not re-execute with $CONFIG_SHELL" >&2
exit 255
fi

        if test x$as_have_required = xno; then :
$as_echo "$0: This script requires a shell more modern than all"
$as_echo "$0: the shells that I found on your system."
if test x${ZSH_VERSION+set} = xset ; then
        $as_echo "$0: In particular, zsh $ZSH_VERSION has bugs and should"
        $as_echo "$0: be upgraded to zsh 4.3.4 or later."
else
        $as_echo "$0: Please tell bug-autoconf@gnu.org and
$0:
https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus&component=Glib
$0: about your system, including any error possibly output
$0: before this message. Then install a modern shell, or
$0: manually run the script under such a shell if you do
$0: have one."
        fi
        exit 1
fi
fi
fi
SHELL=${CONFIG_SHELL-/bin/sh}
export SHELL
# Unset more variables known to interfere with behavior of common
tools.
CLICOLOR_FORCE= GREP_OPTIONS=
unset CLICOLOR_FORCE GREP_OPTIONS

## ----- ##

```



```

$as_echo X"$sas_dir" |
  sed '/^X\(. *[^/]\)\.\.\/* [^/][^/]*\.*$/ {
    s//\1/
    q
  }
/^X\(\.\.\.\)\ [^/].*/ {
  s//\1/
  q
}
/^X\(\.\.\.\)$/ {
  s//\1/
  q
}
/^X\(\.\.\)\.*$/ {
  s//\1/
  q
}
s/.*\/./; q'`
  test -d "$sas_dir" && break
done
  test -z "$sas_dirs" || eval "mkdir $sas_dirs"
} || test -d "$sas_dir" || as_fn_error $? "cannot create directory
$sas_dir"

} # as_fn_mkdir_p

# as_fn_executable_p FILE
# -----
# Test if FILE is an executable regular file.
as_fn_executable_p ()
{
  test -f "$1" && test -x "$1"
} # as_fn_executable_p
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
VAR. Take
# advantage of any shell optimizations that allow amortized linear
growth over
# repeated appends, instead of the typical quadratic growth present in
naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\$as_var = x12") 2>/dev/null;
then :
  eval 'as_fn_append ()
  {
    eval $1+=\$2
  }'
else
  as_fn_append ()
  {

```

```

    eval $1=\$$1\$$2
}
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
the
# global $as_val. Take advantage of shells that can avoid forks. The
arguments
# must be portable across $(( )) and expr.
if (eval "test \${( 1 + 1 )} = 2") 2>/dev/null; then :
    eval 'as_fn_arith ()
    {
        as_val=${( $* )}
    }'
else
    as_fn_arith ()
    {
        as_val=`expr "$@" || test $? -eq 1`
    }
fi # as_fn_arith

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
are
# provided, also output the error to LOG_FD, referencing LINENO. Then
exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
    as_status=$1; test $as_status -eq 0 && as_status=1
    if test "$4"; then
        as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
        $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
    fi
    $as_echo "$as_me: error: $2" >&2
    as_fn_exit $as_status
} # as_fn_error

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
"X/"; then

```

```

    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\/\([^\/]*\)/*$' \|| \
X"$0" : 'X\(/\/\) '$' \|| \
X"$0" : 'X\(/\/\) '$' \|| . 2>/dev/null ||
$as_echo X/"$0" |
    sed '/^.*\/\([^\/]*\)\/*$/{
        s//\1/
        q
    }
/^X\/\(/\/\/\) $/{
        s//\1/
        q
    }
/^X\/\(/\/\) .*/{
        s//\1/
        q
    }
s/.*\/./; q'`

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyz'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

as_lineno_1=$LINENO as_lineno_1a=$LINENO
as_lineno_2=$LINENO as_lineno_2a=$LINENO
eval 'test "x$as_lineno_1'$as_run'" != "x$as_lineno_2'$as_run'" &&
test "x`expr $as_lineno_1'$as_run' + 1`" = "x$as_lineno_2'$as_run'"'
|| {
    # Blame Lee E. McMahon (1931-1989) for sed's syntax.  :-)
    sed -n '
        p
        /[$]LINENO/=
        ' <$as_myself |
        sed '
            s/[$]LINENO.*/&-/
            t lineno

```

```

        b
        :lineno
        N
        :loop
        s/[ $\$$ ]LINENO\([ $\^$ ' $\$$ as_cr_alnum'_].*\n\)\(.*\)/\2\1\2/
        t loop
        s/-\n.*//
        ' > $\$$ as_me.lineno &&
chmod +x " $\$$ as_me.lineno" ||
{  $\$$ as_echo " $\$$ as_me: error: cannot create  $\$$ as_me.lineno; rerun with
a POSIX shell" >&2; as_fn_exit 1; }

# If we had to re-execute with $CONFIG_SHELL, we're ensured to have
# already done that, so ensure we don't try to do so again and fall
# in an infinite loop. This has already happened in practice.
_as_can_reexec=no; export _as_can_reexec
# Don't try to exec as it changes  $\$$ [0], causing all sort of problems
# (the dirname of  $\$$ [0] is not the place where we might find the
# original and so on. Autoconf is especially sensitive to this).
. " $\$$ as_me.lineno"
# Exit status is that of the last command.
exit
}

ECHO_C= ECHO_N= ECHO_T=
case `echo -n x` in #((((
-n*)
  case `echo 'xy\c'` in
  *c*) ECHO_T=' ';; # ECHO_T is single tab character.
  xy) ECHO_C='\c';;
  *) echo `echo ksh88 bug on AIX 6.1` > /dev/null
     ECHO_T=' ';;
  esac;;
*)
  ECHO_N='-n';;
esac

rm -f conf $\$$  $\$$  conf $\$$  $\$$ .exe conf $\$$  $\$$ .file
if test -d conf $\$$  $\$$ .dir; then
  rm -f conf $\$$  $\$$ .dir/conf $\$$  $\$$ .file
else
  rm -f conf $\$$  $\$$ .dir
  mkdir conf $\$$  $\$$ .dir 2>/dev/null
fi
if (echo >conf $\$$  $\$$ .file) 2>/dev/null; then
  if ln -s conf $\$$  $\$$ .file conf $\$$  $\$$  2>/dev/null; then
    as_ln_s='ln -s'
    # ... but there are two gotchas:
    # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
    # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
    executable.
    # In both cases, we have to default to `cp -pR'.

```

```

    ln -s conf$$.$file conf$$.$dir 2>/dev/null && test ! -f conf$$.$exe
||
    as_ln_s='cp -pR'
elif ln conf$$.$file conf$$ 2>/dev/null; then
    as_ln_s=ln
else
    as_ln_s='cp -pR'
fi
else
    as_ln_s='cp -pR'
fi
rm -f conf$$ conf$$.$exe conf$$.$dir/conf$$.$file conf$$.$file
rmdir conf$$.$dir 2>/dev/null

if mkdir -p . 2>/dev/null; then
    as_mkdir_p='mkdir -p "$as_dir"'
else
    test -d ./-p && rmdir ./-p
    as_mkdir_p=false
fi

as_test_x='test -x'
as_executable_p=as_fn_executable_p

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed
'y%*$as_cr_letters%P$as_cr_LETTERS%;s%[^_$as_cr_alnum]%%_g'"

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_$as_cr_alnum]%%_g'"

SHELL=${CONFIG_SHELL-/bin/sh}

test -n "$DJDIR" || exec 7<&0 </dev/null
exec 6>&1

# Name of the host.
# hostname on some systems (SVR3.2, old GNU/Linux) returns a bogus
exit status,
# so uname gets run too.
ac_hostname=`(hostname || uname -n) 2>/dev/null | sed 1q`

#
# Initializations.
#
ac_default_prefix=/usr/local
ac_clean_files=
ac_config_libobj_dir=.
LIBOBJS=
cross_compiling=no
subdirs=

```



```
MFLAGS=
MAKEFLAGS=

# Identity of this package.
PACKAGE_NAME='dbus-glib'
PACKAGE_TARNAME='dbus-glib'
PACKAGE_VERSION='0.100.2'
PACKAGE_STRING='dbus-glib 0.100.2'
PACKAGE_BUGREPORT='https://bugs.freedesktop.org/enter_bug.cgi?product=
dbus&component=GLib'
PACKAGE_URL=''

# Factoring default headers for most tests.
ac_includes_default="\
#include <stdio.h>
#ifdef HAVE_SYS_TYPES_H
# include <sys/types.h>
#endif
#ifdef HAVE_SYS_STAT_H
# include <sys/stat.h>
#endif
#ifdef STDC_HEADERS
# include <stdlib.h>
# include <stddef.h>
#else
# ifdef HAVE_STDLIB_H
# include <stdlib.h>
# endif
#endif
#ifdef HAVE_STRING_H
# if !defined STDC_HEADERS && defined HAVE_MEMORY_H
# include <memory.h>
# endif
# include <string.h>
#endif
#ifdef HAVE_STRINGS_H
# include <strings.h>
#endif
#ifdef HAVE_INTTYPES_H
# include <inttypes.h>
#endif
#ifdef HAVE_STDINT_H
# include <stdint.h>
#endif
#ifdef HAVE_UNISTD_H
# include <unistd.h>
#endif"

ac_subst_vars='am__EXEEXT_FALSE
am__EXEEXT_TRUE
LTLIBOBJS
LIBOBJS'
```

TEST_SOCKET_DIR
ABSOLUTE_TOP_BUILDDIR
TEST_SLEEP_FOREVER_BINARY
TEST_SEGFAULT_BINARY
TEST_EXIT_BINARY
TEST_INTERFACES_SERVICE_BINARY
TEST_CORE_SERVICE_BINARY
TEST_SHELL_SERVICE_BINARY
TEST_SERVICE_BINARY
TEST_SERVICE_DIR
EXPANDED_DATADIR
EXPANDED_LIBDIR
EXPANDED_BINDIR
EXPANDED_SYSCONFDIR
EXPANDED_LOCALSTATEDIR
GTK_DOC_USE_REBASE_FALSE
GTK_DOC_USE_REBASE_TRUE
GTK_DOC_USE_LIBTOOL_FALSE
GTK_DOC_USE_LIBTOOL_TRUE
GTK_DOC_BUILD_PDF_FALSE
GTK_DOC_BUILD_PDF_TRUE
GTK_DOC_BUILD_HTML_FALSE
GTK_DOC_BUILD_HTML_TRUE
ENABLE_GTK_DOC_FALSE
ENABLE_GTK_DOC_TRUE
GTKDOC_DEPS_LIBS
GTKDOC_DEPS_CFLAGS
HTML_DIR
GTKDOC_MKPDF
GTKDOC_REBASE
GTKDOC_CHECK
DBUS_GLIB_TOOL_LIBS
DBUS_GLIB_TOOL_CFLAGS
GLIB_GENMARSHAL
HAVE_GLIB_THREADS_FALSE
HAVE_GLIB_THREADS_TRUE
DBUS_GLIB_THREADS_LIBS
DBUS_GLIB_THREADS_CFLAGS
DBUS_GLIB_LIBS
DBUS_GLIB_CFLAGS
DBUS_LIBS
DBUS_CFLAGS
PKG_CONFIG_LIBDIR
PKG_CONFIG_PATH
PKG_CONFIG
DBUS_PATH_OR_ABSTRACT
OTOOL64
OTOOL
LIPO
NMEDIT
DSYMUTIL
MANIFEST_TOOL

RANLIB
ac_ct_AR
AR
DLLTOOL
OBJDUMP
LN_S
NM
ac_ct_DUMPBIN
DUMPBIN
LD
FGREP
SED
LIBTOOL
DBUS_BUILD_TESTS_FALSE
DBUS_BUILD_TESTS_TRUE
DBUS_BINDING_TOOL
DBUS_BASH_COMPLETION_FALSE
DBUS_BASH_COMPLETION_TRUE
EGREP
GREP
CPP
am__fastdepCC_FALSE
am__fastdepCC_TRUE
CCDEPMODE
am__nodep
AMDEPBACKSLASH
AMDEP_FALSE
AMDEP_TRUE
am__quote
am__include
DEPDIR
OBJEXT
EXEEXT
ac_ct_CC
CPPFLAGS
LDFLAGS
CFLAGS
CC
LT_AGE
LT_REVISION
LT_CURRENT
AM_BACKSLASH
AM_DEFAULT_VERBOSITY
AM_DEFAULT_V
AM_V
MAINT
MAINTAINER_MODE_FALSE
MAINTAINER_MODE_TRUE
am__untar
am__tar
AMTAR
am__leading_dot

SET_MAKE
AWK
mkdir_p
MKDIR_P
INSTALL_STRIP_PROGRAM
STRIP
install_sh
MAKEINFO
AUTOHEADER
AUTOMAKE
AUTOCONF
ACLOCAL
VERSION
PACKAGE
CYGPATH_W
am__isrc
INSTALL_DATA
INSTALL_SCRIPT
INSTALL_PROGRAM
host_os
host_vendor
host_cpu
host
build_os
build_vendor
build_cpu
build
target_alias
host_alias
build_alias
LIBS
ECHO_T
ECHO_N
ECHO_C
DEFS
mandir
localedir
libdir
psdir
pdfdir
dvidir
htmldir
infodir
docdir
oldincludedir
includedir
localstatedir
sharedstatedir
sysconfdir
datadir
datarootdir
libexecdir

```
sbindir
bindir
program_transform_name
prefix
exec_prefix
PACKAGE_URL
PACKAGE_BUGREPORT
PACKAGE_STRING
PACKAGE_VERSION
PACKAGE_TARNAME
PACKAGE_NAME
PATH_SEPARATOR
SHELL'
ac_subst_files=''
ac_user_opts='
enable_option_checking
enable_maintainer_mode
enable_silent_rules
enable_dependency_tracking
enable_tests
enable_ansi
enable_verbose_mode
enable_asserts
enable_checks
enable_gcov
enable_bash_completion
with_test_socket_dir
with_introspect_xml
with_dbus_binding_tool
enable_shared
enable_static
with_pic
enable_fast_install
with_gnu_ld
with_libtool_sysroot
enable_libtool_lock
with_html_dir
enable_gtk_doc
enable_gtk_doc_html
enable_gtk_doc_pdf
'
        ac_precious_vars='build_alias
host_alias
target_alias
CC
CFLAGS
LDFLAGS
LIBS
CPPFLAGS
CPP
PKG_CONFIG
PKG_CONFIG_PATH
```

```
PKG_CONFIG_LIBDIR
DBUS_CFLAGS
DBUS_LIBS
DBUS_GLIB_CFLAGS
DBUS_GLIB_LIBS
DBUS_GLIB_THREADS_CFLAGS
DBUS_GLIB_THREADS_LIBS
GTKDOC_DEPS_CFLAGS
GTKDOC_DEPS_LIBS'
```

```
# Initialize some variables set by options.
ac_init_help=
ac_init_version=false
ac_unrecognized_opts=
ac_unrecognized_sep=
# The variables have the same names as the options, with
# dashes changed to underlines.
cache_file=/dev/null
exec_prefix=NONE
no_create=
no_recursion=
prefix=NONE
program_prefix=NONE
program_suffix=NONE
program_transform_name=s,x,x,
silent=
site=
srcdir=
verbose=
x_includes=NONE
x_libraries=NONE
```

```
# Installation directory options.
# These are left unexpanded so users can "make install
exec_prefix=/foo"
# and all the variables that are supposed to be based on exec_prefix
# by default will actually change.
# Use braces instead of parens because sh, perl, etc. also accept
them.
# (The list follows the same order as the GNU Coding Standards.)
bindir='${exec_prefix}/bin'
sbindir='${exec_prefix}/sbin'
libexecdir='${exec_prefix}/libexec'
datarootdir='${prefix}/share'
datadir='${datarootdir}'
sysconfdir='${prefix}/etc'
sharedstatedir='${prefix}/com'
localstatedir='${prefix}/var'
includedir='${prefix}/include'
oldincludedir='/usr/include'
docdir='${datarootdir}/doc/${PACKAGE_TARNAME}'
```

```

infodir='${datarootdir}/info'
htmldir='${docdir}'
dvidir='${docdir}'
pdfdir='${docdir}'
psdir='${docdir}'
libdir='${exec_prefix}/lib'
localedir='${datarootdir}/locale'
mandir='${datarootdir}/man'

ac_prev=
ac_dashdash=
for ac_option
do
  # If the previous option needs an argument, assign it.
  if test -n "$ac_prev"; then
    eval $ac_prev=\$ac_option
    ac_prev=
    continue
  fi

  case $ac_option in
    *=?*) ac_optarg=`expr "X$ac_option" : '[^=]*=\.(\.*\)'` ;;
    *)    ac_optarg= ;;
  esac

  # Accept the important Cygnus configure options, so we can diagnose
  typos.

  case $ac_dashdash$ac_option in
    --)
      ac_dashdash=yes ;;

    -bindir | --bindir | --bindi | --bind | --bin | --bi)
      ac_prev=bindir ;;
    -bindir=* | --bindir=* | --bindi=* | --bind=* | --bin=* | --bi=*)
      bindir=$ac_optarg ;;

    -build | --build | --buil | --bui | --bu)
      ac_prev=build_alias ;;
    -build=* | --build=* | --buil=* | --bui=* | --bu=*)
      build_alias=$ac_optarg ;;

    -cache-file | --cache-file | --cache-fil | --cache-fi \
    | --cache-f | --cache- | --cache | --cach | --cac | --ca | --c)
      ac_prev=cache_file ;;
    -cache-file=* | --cache-file=* | --cache-fil=* | --cache-fi=* \
    | --cache-f=* | --cache-==* | --cache=* | --cach=* | --cac=* | --ca=*
    | --c=*)
      cache_file=$ac_optarg ;;

    --config-cache | -C)

```

```

    cache_file=config.cache ;;

-datadir | --datadir | --datadi | --datad)
    ac_prev=datadir ;;
-datadir=* | --datadir=* | --datadi=* | --datad=*)
    datadir=$ac_optarg ;;

-datarootdir | --datarootdir | --datarootdi | --datarootd | --
dataroot \
| --dataroot | --dataroot | --dataroot)
    ac_prev=datarootdir ;;
-datarootdir=* | --datarootdir=* | --datarootdi=* | --datarootd=* \
| --dataroot=* | --dataroot=* | --dataroot=* | --dataroot=*)
    datarootdir=$ac_optarg ;;

-disable-* | --disable-*)
    ac_useropt=`expr "x$ac_option" : 'x-*disable-\(.*\) '`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
        as_fn_error $? "invalid feature name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
        *)
"enable_$ac_useropt"
"*) ;;
        *)
ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--
disable-$ac_useropt_orig"
        ac_unrecognized_sep=', ';;
    esac
    eval enable_$ac_useropt=no ;;

-docdir | --docdir | --docdi | --doc | --do)
    ac_prev=docdir ;;
-docdir=* | --docdir=* | --docdi=* | --doc=* | --do=*)
    docdir=$ac_optarg ;;

-dvidir | --dvidir | --dvidi | --dvid | --dvi | --dv)
    ac_prev=dvidir ;;
-dvidir=* | --dvidir=* | --dvidi=* | --dvid=* | --dvi=* | --dv=*)
    dvidir=$ac_optarg ;;

-enable-* | --enable-*)
    ac_useropt=`expr "x$ac_option" : 'x-*enable-\([^=]*\) '`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
        as_fn_error $? "invalid feature name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
        *)

```



```

"enable_${ac_useropt}"
"*) ;;
    *)
ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--
enable-${ac_useropt}_orig"
    ac_unrecognized_sep=', ';;
    esac
    eval enable_${ac_useropt}=\${ac_optarg} ;;

-exec-prefix | --exec_prefix | --exec-prefix | --exec-prefi \
| --exec-pref | --exec-pre | --exec-pr | --exec-p | --exec- \
| --exec | --exe | --ex)
    ac_prev=exec_prefix ;;
-exec-prefix=* | --exec_prefix=* | --exec-prefix=* | --exec-prefi=*
\
| --exec-pref=* | --exec-pre=* | --exec-pr=* | --exec-p=* | --exec-
=* \
| --exec=* | --exe=* | --ex=*)
    exec_prefix=${ac_optarg} ;;

-gas | --gas | --ga | --g)
    # Obsolete; use --with-gas.
    with_gas=yes ;;

-help | --help | --hel | --he | -h)
    ac_init_help=long ;;
-help=r* | --help=r* | --hel=r* | --he=r* | -hr*)
    ac_init_help=recursive ;;
-help=s* | --help=s* | --hel=s* | --he=s* | -hs*)
    ac_init_help=short ;;

-host | --host | --hos | --ho)
    ac_prev=host_alias ;;
-host=* | --host=* | --hos=* | --ho=*)
    host_alias=${ac_optarg} ;;

-htmldir | --htmldir | --htmlidi | --htmlid | --html | --htm | --ht)
    ac_prev=htmldir ;;
-htmldir=* | --htmldir=* | --htmlidi=* | --htmlid=* | --html=* | --
htm=* \
| --ht=*)
    htmldir=${ac_optarg} ;;

-includedir | --includedir | --includedi | --included | --include \
| --includ | --inclu | --incl | --inc)
    ac_prev=includedir ;;
-includedir=* | --includedir=* | --includedi=* | --included=* | --
include=* \
| --includ=* | --inclu=* | --incl=* | --inc=*)
    includedir=${ac_optarg} ;;

-infodir | --infodir | --infodi | --infod | --info | --inf)

```

```

    ac_prev=infodir ;;
    -infodir=* | --infodir=* | --infodi=* | --infod=* | --info=* | --
inf=*)
    infodir=$ac_optarg ;;

    -libdir | --libdir | --libdi | --libd)
    ac_prev=libdir ;;
    -libdir=* | --libdir=* | --libdi=* | --libd=*)
    libdir=$ac_optarg ;;

    -libexecdir | --libexecdir | --libexecdi | --libexecd | --libexec \
| --libexe | --libex | --libe)
    ac_prev=libexecdir ;;
    -libexecdir=* | --libexecdir=* | --libexecdi=* | --libexecd=* | --
libexec=* \
| --libexe=* | --libex=* | --libe=*)
    libexecdir=$ac_optarg ;;

    -localedir | --localedir | --localedi | --localed | --locale)
    ac_prev=localedir ;;
    -localedir=* | --localedir=* | --localedi=* | --localed=* | --
locale=*)
    localedir=$ac_optarg ;;

    -localstatedir | --localstatedir | --localstatedi | --localstated \
| --localstate | --localstat | --localsta | --localst | --locals)
    ac_prev=localstatedir ;;
    -localstatedir=* | --localstatedir=* | --localstatedi=* | --
localstated=* \
| --localstate=* | --localstat=* | --localsta=* | --localst=* | --
locals=*)
    localstatedir=$ac_optarg ;;

    -mandir | --mandir | --mandi | --mand | --man | --ma | --m)
    ac_prev=mandir ;;
    -mandir=* | --mandir=* | --mandi=* | --mand=* | --man=* | --ma=* | -
-m=*)
    mandir=$ac_optarg ;;

    -nfp | --nfp | --nf)
    # Obsolete; use --without-fp.
    with_fp=no ;;

    -no-create | --no-create | --no-creat | --no-crea | --no-cre \
| --no-cr | --no-c | -n)
    no_create=yes ;;

    -no-recursion | --no-recursion | --no-recursio | --no-recursi \
| --no-recurs | --no-recur | --no-recu | --no-rec | --no-re | --no-
r)
    no_recursion=yes ;;

```

```

-oldincludedir | --oldincludedir | --oldincludedi | --oldincluded \
| --oldinclude | --oldinclud | --oldinclu | --oldincl | --oldinc \
| --oldin | --oldi | --old | --ol | --o)
    ac_prev=oldincludedir ;;
-oldincludedir=* | --oldincludedir=* | --oldincludedi=* | --
oldincluded=* \
| --oldinclude=* | --oldinclud=* | --oldinclu=* | --oldincl=* | --
oldinc=* \
| --oldin=* | --oldi=* | --old=* | --ol=* | --o=*)
    oldincludedir=$ac_optarg ;;

-prefix | --prefix | --prefi | --pref | --pre | --pr | --p)
    ac_prev=prefix ;;
-prefix=* | --prefix=* | --prefi=* | --pref=* | --pre=* | --pr=* | -
-p=*)
    prefix=$ac_optarg ;;

-program-prefix | --program-prefix | --program-prefi | --program-
pref \
| --program-pre | --program-pr | --program-p)
    ac_prev=program_prefix ;;
-program-prefix=* | --program-prefix=* | --program-prefi=* \
| --program-pref=* | --program-pre=* | --program-pr=* | --program-
p=*)
    program_prefix=$ac_optarg ;;

-program-suffix | --program-suffix | --program-suffi | --program-
suff \
| --program-suf | --program-su | --program-s)
    ac_prev=program_suffix ;;
-program-suffix=* | --program-suffix=* | --program-suffi=* \
| --program-suff=* | --program-suf=* | --program-su=* | --program-
s=*)
    program_suffix=$ac_optarg ;;

-program-transform-name | --program-transform-name \
| --program-transform-nam | --program-transform-na \
| --program-transform-n | --program-transform- \
| --program-transform | --program-transfor \
| --program-transfo | --program-transf \
| --program-trans | --program-tran \
| --progr-tra | --program-tr | --program-t)
    ac_prev=program_transform_name ;;
-program-transform-name=* | --program-transform-name=* \
| --program-transform-nam=* | --program-transform-na=* \
| --program-transform-n=* | --program-transform-=* \
| --program-transform=* | --program-transfor=* \
| --program-transfo=* | --program-transf=* \
| --program-trans=* | --program-tran=* \
| --progr-tra=* | --program-tr=* | --program-t=*)
    program_transform_name=$ac_optarg ;;

```

```

-pdfdir | --pdfdir | --pdfdi | --pdfd | --pdf | --pd)
    ac_prev=pdfdir ;;
-pdfdir=* | --pdfdir=* | --pdfdi=* | --pdfd=* | --pdf=* | --pd=*)
    pdfdir=$ac_optarg ;;

-psdir | --psdir | --psdi | --psd | --ps)
    ac_prev=psdir ;;
-psdir=* | --psdir=* | --psdi=* | --psd=* | --ps=*)
    psdir=$ac_optarg ;;

-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil)
    silent=yes ;;

-sbindir | --sbindir | --sbindi | --sbind | --sbin | --sbi | --sb)
    ac_prev=sbindir ;;
-sbindir=* | --sbindir=* | --sbindi=* | --sbind=* | --sbin=* \
| --sbi=* | --sb=*)
    sbindir=$ac_optarg ;;

-sharedstatedir | --sharedstatedir | --sharedstatedi \
| --sharedstated | --sharedstate | --sharedstat | --sharedsta \
| --sharedst | --shares | --shared | --share | --share \
| --sha | --sh)
    ac_prev=sharedstatedir ;;
-sharedstatedir=* | --sharedstatedir=* | --sharedstatedi=* \
| --sharedstated=* | --sharedstate=* | --sharedstat=* | --
sharedsta=* \
| --sharedst=* | --shares=* | --shared=* | --share=* | --share=* \
| --sha=* | --sh=*)
    sharedstatedir=$ac_optarg ;;

-site | --site | --sit)
    ac_prev=site ;;
-site=* | --site=* | --sit=*)
    site=$ac_optarg ;;

-srcdir | --srcdir | --srcdi | --srcd | --src | --sr)
    ac_prev=srcdir ;;
-srcdir=* | --srcdir=* | --srcdi=* | --srcd=* | --src=* | --sr=*)
    srcdir=$ac_optarg ;;

-sysconfdir | --sysconfdir | --sysconfdi | --sysconfd | --sysconf \
| --syscon | --sysco | --sysc | --sys | --sy)
    ac_prev=sysconfdir ;;
-sysconfdir=* | --sysconfdir=* | --sysconfdi=* | --sysconfd=* | --
sysconf=* \
| --syscon=* | --sysco=* | --sysc=* | --sys=* | --sy=*)
    sysconfdir=$ac_optarg ;;

-target | --target | --targe | --targ | --tar | --ta | --t)
    ac_prev=target_alias ;;

```

```

-target=* | --target=* | --targe=* | --targ=* | --tar=* | --ta=* | -
-t=*)
    target_alias=$ac_optarg ;;

-v | -verbose | --verbose | --verbos | --verbo | --verb)
    verbose=yes ;;

-version | --version | --versio | --versi | --vers | -V)
    ac_init_version=: ;;

-with-* | --with-*)
    ac_useropt=`expr "x$ac_option" : 'x-*with-\([^=]*\)'`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
        as_fn_error $? "invalid package name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
        *)
            "with_$ac_useropt"
            *) ;;
        *)
    ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--with-
$ac_useropt_orig"
        ac_unrecognized_sep=', ';;
    esac
    eval with_$ac_useropt=\$ac_optarg ;;

-without-* | --without-*)
    ac_useropt=`expr "x$ac_option" : 'x-*without-\(.*\) '`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
        as_fn_error $? "invalid package name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
        *)
            "with_$ac_useropt"
            *) ;;
        *)
    ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--
without-$ac_useropt_orig"
        ac_unrecognized_sep=', ';;
    esac
    eval with_$ac_useropt=no ;;

--x)
    # Obsolete; use --with-x.
    with_x=yes ;;

-x-includes | --x-includes | --x-include | --x-includ | --x-inclu \
| --x-incl | --x-inc | --x-in | --x-i)

```

```

    ac_prev=x_includes ;;
    -x-includes=* | --x-includes=* | --x-include=* | --x-includ=* | --x-
inclu=* \
    | --x-incl=* | --x-inc=* | --x-in=* | --x-i=*)
    x_includes=$ac_optarg ;;

    -x-libraries | --x-libraries | --x-librarie | --x-librari \
    | --x-librar | --x-libra | --x-libr | --x-lib | --x-li | --x-l)
    ac_prev=x_libraries ;;
    -x-libraries=* | --x-libraries=* | --x-librarie=* | --x-librari=* \
    | --x-librar=* | --x-libra=* | --x-libr=* | --x-lib=* | --x-li=* | -
-x-l=*)
    x_libraries=$ac_optarg ;;

    -*) as_fn_error $? "unrecognized option: \`${ac_option}'
Try \`${0} --help' for more information"
    ;;

    *)
    ac_envvar=`expr "x${ac_option}" : 'x\([^=]*\)='`
    # Reject names that are not valid shell variable names.
    case $ac_envvar in #(
        '' | [0-9]* | *[^_${as_cr_alnum}]* )
        as_fn_error $? "invalid variable name: \`${ac_envvar}'" ;;
    esac
    eval $ac_envvar=\${ac_optarg}
    export $ac_envvar ;;

    *)
    # FIXME: should be removed in autoconf 3.0.
    $as_echo "$as_me: WARNING: you should use --build, --host, --
target" >&2
    expr "x${ac_option}" : ".*\[^-._${as_cr_alnum}]" >/dev/null &&
    $as_echo "$as_me: WARNING: invalid host type: ${ac_option}" >&2
    : "${build_alias=${ac_option}} ${host_alias=${ac_option}}
${target_alias=${ac_option}}
    ;;

    esac
done

if test -n "$ac_prev"; then
    ac_option=--`echo $ac_prev | sed 's/_/_/g'`
    as_fn_error $? "missing argument to ${ac_option}"
fi

if test -n "$ac_unrecognized_opts"; then
    case $enable_option_checking in
        no) ;;
        fatal) as_fn_error $? "unrecognized options:
${ac_unrecognized_opts}" ;;
    esac

```

```

    *)      $as_echo "$as_me: WARNING: unrecognized options:
$ac_unrecognized_opts" >&2 ;;
  esac
fi

# Check all directory arguments for consistency.
for ac_var in      exec_prefix prefix bindir sbindir libexecdir
datarootdir \
                datadir sysconfdir sharedstatedir localstatedir includedir
\
                oldincludedir docdir infodir htmdir dvidir pdfdir psdir \
                libdir localedir mandir
do
  eval ac_val=\$$ac_var
  # Remove trailing slashes.
  case $ac_val in
    */ )
      ac_val=`expr "X$ac_val" : 'X\([^/]\)' \| "X$ac_val" :
'X\(.*\)'`
      eval $ac_var=\$ac_val;;
    esac
  # Be sure to have absolute directory names.
  case $ac_val in
    [\\/$]* | ?:[\\/$]* ) continue;;
    NONE | '' ) case $ac_var in *prefix ) continue;; esac;;
  esac
  as_fn_error $? "expected an absolute directory name for --$ac_var:
$ac_val"
done

# There might be people who depend on the old broken behavior: `$host'
# used to hold the argument of --host etc.
# FIXME: To remove some day.
build=$build_alias
host=$host_alias
target=$target_alias

# FIXME: To remove some day.
if test "x$host_alias" != x; then
  if test "x$build_alias" = x; then
    cross_compiling=maybe
  elif test "x$build_alias" != "x$host_alias"; then
    cross_compiling=yes
  fi
fi

ac_tool_prefix=
test -n "$host_alias" && ac_tool_prefix=$host_alias-

test "$silent" = yes && exec 6>/dev/null

```

```

ac_pwd=`pwd` && test -n "$ac_pwd" &&
ac_ls_di=`ls -di .` &&
ac_pwd_ls_di=`cd "$ac_pwd" && ls -di .` ||
  as_fn_error $? "working directory cannot be determined"
test "X$ac_ls_di" = "X$ac_pwd_ls_di" ||
  as_fn_error $? "pwd does not report name of working directory"

# Find the source files, if location was not specified.
if test -z "$srcdir"; then
  ac_srcdir_defaulted=yes
  # Try the directory containing this script, then the parent
  directory.
  ac_confdir=`$as_dirname -- "$as_myself" ||
$as_expr X"$as_myself" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \| \
  X"$as_myself" : 'X\(//\)[^/]' \| \
  X"$as_myself" : 'X\(//\)$' \| \
  X"$as_myself" : 'X\(/\)' \| . 2>/dev/null ||
$as_echo X"$as_myself" |
  sed '/^X\(.*[^/]\)\/*[^/][^/]*/*$/{
    s//\1/
    q
  }
/^X\(\\/\)\)[^/].*${
  s//\1/
  q
}
/^X\(\\/\)\)$/{
  s//\1/
  q
}
/^X\(\\/\).*${
  s//\1/
  q
}
s/.*/./; q'`
  srcdir=$ac_confdir
  if test ! -r "$srcdir/$ac_unique_file"; then
    srcdir=..
  fi
else
  ac_srcdir_defaulted=no
fi
if test ! -r "$srcdir/$ac_unique_file"; then
  test "$ac_srcdir_defaulted" = yes && srcdir="$ac_confdir or .."
  as_fn_error $? "cannot find sources ($ac_unique_file) in $srcdir"
fi
ac_msg="sources are in $srcdir, but `cd $srcdir` does not work"
ac_abs_confdir=`(
  cd "$srcdir" && test -r ".$ac_unique_file" || as_fn_error $?
"$ac_msg"
  pwd)`

```



```

# When building in place, set srcdir=.
if test "$ac_abs_confdir" = "$ac_pwd"; then
  srcdir=.
fi
# Remove unnecessary trailing slashes from srcdir.
# Double slashes in file names in object file debugging info
# mess up M-x gdb in Emacs.
case $srcdir in
*/) srcdir=`expr "X$srcdir" : 'X\([^/]\)' \| "X$srcdir" :
'X\(.*\)'`;;
esac
for ac_var in $ac_precious_vars; do
  eval ac_env_${ac_var}_set=\${${ac_var}_set}
  eval ac_env_${ac_var}_value=\${${ac_var}_value}
  eval ac_cv_env_${ac_var}_set=\${${ac_var}_set}
  eval ac_cv_env_${ac_var}_value=\${${ac_var}_value}
done

#
# Report the --help message.
#
if test "$ac_init_help" = "long"; then
  # Omit some internal or obsolete options to make the list less
  imposing.
  # This message is too long to be a string in the A/UX 3.1 sh.
  cat <<_ACEOF
\`configure' configures dbus-glib 0.100.2 to adapt to many kinds of
systems.

```

Usage: \$0 [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as VAR=VALUE. See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:

-h, --help	display this help and exit
--help=short	display options specific to this package
--help=recursive	display the short help of all the included packages
-V, --version	display version information and exit
-q, --quiet, --silent	do not print '\`checking ...' messages
--cache-file=FILE	cache test results in FILE [disabled]
-C, --config-cache	alias for '\`--cache-file=config.cache'
-n, --no-create	do not create output files
--srcdir=DIR	find the sources in DIR [configure dir or '\`..']

Installation directories:

```

--prefix=PREFIX          install architecture-independent files in
PREFIX
                          [$ac_default_prefix]
--exec-prefix=EPREFIX    install architecture-dependent files in
EPREFIX
                          [PREFIX]

```

By default, `\`make install'` will install all the files in `\`$ac_default_prefix/bin'`, `\`$ac_default_prefix/lib'` etc. You can specify an installation prefix other than `\`$ac_default_prefix'` using `\`--prefix'`, for instance `\`--prefix=~$HOME'`.

For better control, use the options below.

Fine tuning of the installation directories:

```

--bindir=DIR             user executables [EPREFIX/bin]
--sbindir=DIR           system admin executables [EPREFIX/sbin]
--libexecdir=DIR       program executables [EPREFIX/libexec]
--sysconfdir=DIR       read-only single-machine data [PREFIX/etc]
--sharedstatedir=DIR   modifiable architecture-independent data
[PREFIX/com]
--localstatedir=DIR    modifiable single-machine data [PREFIX/var]
--libdir=DIR           object code libraries [EPREFIX/lib]
--includedir=DIR      C header files [PREFIX/include]
--oldincludedir=DIR   C header files for non-gcc [/usr/include]
--datarootdir=DIR     read-only arch.-independent data root
[PREFIX/share]
--datadir=DIR         read-only architecture-independent data
[DATAROOTDIR]
--infodir=DIR        info documentation [DATAROOTDIR/info]
--localedir=DIR     locale-dependent data [DATAROOTDIR/locale]
--mandir=DIR        man documentation [DATAROOTDIR/man]
--docdir=DIR        documentation root [DATAROOTDIR/doc/dbus-
glib]
--htmldir=DIR       html documentation [DOCDIR]
--dvidir=DIR        dvi documentation [DOCDIR]
--pdfdir=DIR        pdf documentation [DOCDIR]
--psdir=DIR         ps documentation [DOCDIR]
_ACEOF

```

```

cat <<\_ACEOF

```

Program names:

```

--program-prefix=PREFIX    prepend PREFIX to installed
program names
--program-suffix=SUFFIX    append SUFFIX to installed
program names
--program-transform-name=PROGRAM  run sed PROGRAM on installed
program names

```

System types:

```
--build=BUILD      configure for building on BUILD [guessed]
--host=HOST        cross-compile to build programs to run on HOST
[BUILD]
_ACEOF
fi
```

```
if test -n "$ac_init_help"; then
```

```
  case $ac_init_help in
    short | recursive ) echo "Configuration of dbus-glib 0.100.2:>";;
    esac
  cat <<\_ACEOF
```

Optional Features:

```
--disable-option-checking  ignore unrecognized --enable/--with
options
--disable-FEATURE          do not include FEATURE (same as --enable-
FEATURE=no)
--enable-FEATURE[=ARG]    include FEATURE [ARG=yes]
--enable-maintainer-mode  enable make rules and dependencies not
useful (and
                           sometimes confusing) to the casual installer
--enable-silent-rules     less verbose build output (undo: "make V=1")
--disable-silent-rules   verbose build output (undo: "make V=0")
--enable-dependency-tracking
                           do not reject slow dependency extractors
--disable-dependency-tracking
                           speeds up one-time build
--enable-tests            enable unit test code
--enable-ansi             enable -ansi -pedantic gcc flags
--enable-verbose-mode    support verbose debug mode
--enable-asserts         include assertion checks
--enable-checks          include sanity checks on public API
--enable-gcov            compile with coverage profiling
instrumentation (gcc
                           only)
--enable-bash-completion  install bash completion scripts
--enable-shared[=PKGS]   build shared libraries [default=yes]
--enable-static[=PKGS]   build static libraries [default=yes]
--enable-fast-install[=PKGS]
                           optimize for fast installation [default=yes]
--disable-libtool-lock   avoid locking (might break parallel builds)
--enable-gtk-doc         use gtk-doc to build documentation
[[default=no]]
--enable-gtk-doc-html    build documentation in html format
[[default=yes]]
--enable-gtk-doc-pdf     build documentation in pdf format
[[default=no]]
```

Optional Packages:

```

--with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
--without-PACKAGE      do not use PACKAGE (same as --with-
PACKAGE=no)
--with-test-socket-dir=dirname
                        Where to put sockets for make check
--with-introspect-xml=filename
                        Pass in a pregenerated dbus daemon
introspection xml
                        file (as generated by 'dbus-daemon --
introspect') to
                        use instead of querying the installed dbus
daemon
--with-dbus-binding-tool=filename
                        Use external dbus-binding-tool program
--with-pic[=PKGS]      try to use only PIC/non-PIC objects
[default=use
                        both]
--with-gnu-ld           assume the C compiler uses GNU ld
[default=no]
--with-libtool-sysroot=DIR Search for dependent libraries within DIR
                        (or the compiler's sysroot if not specified).
--with-html-dir=PATH   path to installed docs

```

Some influential environment variables:

```

CC          C compiler command
CFLAGS      C compiler flags
LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in
a
            nonstandard directory <lib dir>
LIBS        libraries to pass to the linker, e.g. -l<library>
CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include
dir> if
            you have headers in a nonstandard directory <include
dir>
CPP         C preprocessor
PKG_CONFIG  path to pkg-config utility
PKG_CONFIG_PATH
            directories to add to pkg-config's search path
PKG_CONFIG_LIBDIR
            path overriding pkg-config's built-in search path
DBUS_CFLAGS C compiler flags for DBUS, overriding pkg-config
DBUS_LIBS   linker flags for DBUS, overriding pkg-config
DBUS_GLIB_CFLAGS
            C compiler flags for DBUS_GLIB, overriding pkg-config
DBUS_GLIB_LIBS
            linker flags for DBUS_GLIB, overriding pkg-config
DBUS_GLIB_THREADS_CFLAGS
            C compiler flags for DBUS_GLIB_THREADS, overriding pkg-
config
DBUS_GLIB_THREADS_LIBS
            linker flags for DBUS_GLIB_THREADS, overriding pkg-
config

```

```
GTKDOC_DEPS_CFLAGS
    C compiler flags for GTKDOC_DEPS, overriding pkg-config
GTKDOC_DEPS_LIBS
    linker flags for GTKDOC_DEPS, overriding pkg-config
```

Use these variables to override the choices made by `configure' or to help it to find libraries and programs with nonstandard names/locations.

Report bugs to

<https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLib>.

_ACEOF

ac_status=\$?

fi

```
if test "$ac_init_help" = "recursive"; then
  # If there are subdirs, report their specific --help.
  for ac_dir in : $ac_subdirs_all; do test "x$ac_dir" = x: && continue
    test -d "$ac_dir" ||
      { cd "$srcdir" && ac_pwd=`pwd` && srcdir=. && test -d "$ac_dir";
    } ||
      continue
    ac_builddir=.

```

```
case "$ac_dir" in
```

```
.) ac_dir_suffix= ac_top_builddir_sub=. ac_top_build_prefix= ;;
```

```
*)
```

```
  ac_dir_suffix=`$as_echo "$ac_dir" | sed 's|^\.([\//]||)`
```

```
  # A ".." for each directory in $ac_dir_suffix.
```

```
  ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/[^\//]*|/..|g;s|/||`
```

```
  case $ac_top_builddir_sub in
```

```
  "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
```

```
  *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
```

```
  esac ;;
```

```
esac
```

```
ac_abs_top_builddir=$ac_pwd
```

```
ac_abs_builddir=$ac_pwd$ac_dir_suffix
```

```
# for backward compatibility:
```

```
ac_top_builddir=$ac_top_build_prefix
```

```
case $srcdir in
```

```
.) # We are building in place.
```

```
  ac_srcdir=.
```

```
  ac_top_srcdir=$ac_top_builddir_sub
```

```
  ac_abs_top_srcdir=$ac_pwd ;;
```

```
[\//]* |?:[\//]* ) # Absolute name.
```

```
  ac_srcdir=$srcdir$ac_dir_suffix;
```

```
  ac_top_srcdir=$srcdir
```

```
  ac_abs_top_srcdir=$srcdir ;;
```

```
*) # Relative name.
```

```

    ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
    ac_top_srcdir=$ac_top_build_prefix$srcdir
    ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

    cd "$ac_dir" || { ac_status=$?; continue; }
    # Check for gusted configure.
    if test -f "$ac_srcdir/configure.gnu"; then
        echo &&
        $SHELL "$ac_srcdir/configure.gnu" --help=recursive
    elif test -f "$ac_srcdir/configure"; then
        echo &&
        $SHELL "$ac_srcdir/configure" --help=recursive
    else
        $as_echo "$as_me: WARNING: no configuration information is in
$ac_dir" >&2
        fi || ac_status=$?
        cd "$ac_pwd" || { ac_status=$?; break; }
    done
fi

test -n "$ac_init_help" && exit $ac_status
if $ac_init_version; then
    cat <<\_ACEOF
dbus-glib configure 0.100.2
generated by GNU Autoconf 2.69

Copyright (C) 2012 Free Software Foundation, Inc.
This configure script is free software; the Free Software Foundation
gives unlimited permission to copy, distribute and modify it.
_ACEOF
    exit
fi

## ----- ##
## Autoconf initialization. ##
## ----- ##

# ac_fn_c_try_compile LINENO
# -----
# Try to compile conftest.$ac_ext, and return whether this succeeded.
ac_fn_c_try_compile ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    rm -f conftest.$ac_objext
    if { { ac_try="$ac_compile"
case "($ac_try" in
    *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
    *) ac_try_echo=$ac_try;;
esac

```

```

eval ac_try_echo="\\"$as_me:${as_lineno-$LINENO}: $ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_compile") 2>confptest.err
  ac_status=$?
  if test -s confptest.err; then
    grep -v '^ *+' confptest.err >confptest.er1
    cat confptest.er1 >&5
    mv -f confptest.er1 confptest.err
  fi
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; } && {
    test -z "$ac_c_werror_flag" ||
    test ! -s confptest.err
  } && test -s confptest.$ac_objext; then :
  ac_retval=0
else
  $as_echo "$as_me: failed program was:" >&5
  sed 's/^/| /' confptest.$ac_ext >&5

  ac_retval=1
fi
eval $as_lineno_stack; ${as_lineno_stack:+} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_c_try_compile

# ac_fn_c_try_link LINENO
# -----
# Try to link confptest.$ac_ext, and return whether this succeeded.
ac_fn_c_try_link ()
{
  as_lineno=${as_lineno-"$1"}
  as_lineno_stack=as_lineno_stack=$as_lineno_stack
  rm -f confptest.$ac_objext confptest$ac_exeext
  if { { ac_try="$ac_link"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\\"$as_me:${as_lineno-$LINENO}: $ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link") 2>confptest.err
  ac_status=$?
  if test -s confptest.err; then
    grep -v '^ *+' confptest.err >confptest.er1
    cat confptest.er1 >&5
    mv -f confptest.er1 confptest.err
  fi
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; } && {
    test -z "$ac_c_werror_flag" ||
    test ! -s confptest.err

```

```

        } && test -s confptest$ac_exeext && {
        test "$cross_compiling" = yes ||
        test -x confptest$ac_exeext
        }; then :
    ac_retval=0
else
    $as_echo "$as_me: failed program was:" >&5
    sed 's/^/| /' confptest.$ac_ext >&5

        ac_retval=1
fi
# Delete the IPA/IPO (Inter Procedural Analysis/Optimization)
information
# created by the PGI compiler (confptest_ipa8_confptest.o), as it
would
# interfere with the next link command; also delete a directory that
is
# left behind by Apple's compiler. We do this before executing the
actions.
rm -rf confptest.dSYM confptest_ipa8_confptest.o
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_c_try_link

# ac_fn_c_try_cpp LINENO
# -----
# Try to preprocess confptest.$ac_ext, and return whether this
succeeded.
ac_fn_c_try_cpp ()
{
    as_lineno=${as_lineno-"$1"}
    as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if { { ac_try="$ac_cpp confptest.$ac_ext"
case "($ac_try" in
    *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
    *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: $ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_cpp confptest.$ac_ext") 2>confptest.err
    ac_status=$?
    if test -s confptest.err; then
        grep -v '^ *+' confptest.err >confptest.er1
        cat confptest.er1 >&5
        mv -f confptest.er1 confptest.err
    fi
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; } > confptest.i && {
        test -z "$ac_c_preproc_warn_flag$ac_c_werror_flag" ||
        test ! -s confptest.err
    }; then :

```



```

    ac_retval=0
else
    $as_echo "$as_me: failed program was:" >&5
sed 's/^/| /' conftest.$ac_ext >&5

    ac_retval=1
fi
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_c_try_cpp

# ac_fn_c_try_run LINENO
# -----
# Try to link conftest.$ac_ext, and return whether this succeeded.
Assumes
# that executables *can* be run.
ac_fn_c_try_run ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if { { ac_try="$ac_link"
case "((($ac_try" in
    *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
    *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_link") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
    test $ac_status = 0; } && { ac_try='./conftest$ac_exeext'
    { { case "((($ac_try" in
    *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
    *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_try") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
    test $ac_status = 0; }; }; then :
    ac_retval=0
else
    $as_echo "$as_me: program exited with status $ac_status" >&5
    $as_echo "$as_me: failed program was:" >&5
sed 's/^/| /' conftest.$ac_ext >&5

    ac_retval=$ac_status
fi
rm -rf conftest.dSYM conftest_ipa8_conftest.oo
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno

```

```

    as_fn_set_status $ac_retval

} # ac_fn_c_try_run

# ac_fn_c_check_header_compile LINENO HEADER VAR INCLUDES
# -----
# Tests whether HEADER exists and can be compiled using the include
files in
# INCLUDES, setting the cache variable VAR accordingly.
ac_fn_c_check_header_compile ()
{
    as_lineno=${as_lineno-"$1"}
    as_lineno_stack=as_lineno_stack=$as_lineno_stack
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
    if eval \${$3+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
#include <$2>
_ACEOF
    if ac_fn_c_try_compile "$LINENO"; then :
        eval "$3=yes"
    else
        eval "$3=no"
    fi
    rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
    fi
    eval ac_res=\${$3}
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
        eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
} # ac_fn_c_check_header_compile

# ac_fn_c_check_func LINENO FUNC VAR
# -----
# Tests whether FUNC exists, setting the cache variable VAR
accordingly
ac_fn_c_check_func ()
{
    as_lineno=${as_lineno-"$1"}
    as_lineno_stack=as_lineno_stack=$as_lineno_stack
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
    if eval \${$3+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext

```

```

/* end confdefs.h.  */
/* Define $2 to an innocuous variant, in case <limits.h> declares $2.
   For example, HP-UX 11i <limits.h> declares gettimeofday.  */
#define $2 innocuous_$2

/* System header to define __stub macros and hopefully few prototypes,
   which can conflict with char $2 (); below.
   Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
   <limits.h> exists even on freestanding compilers.  */

#ifdef __STDC__
#include <limits.h>
#else
#include <assert.h>
#endif

#undef $2

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char $2 ();
/* The GNU C library defines this for functions which it implements
   to always fail with ENOSYS.  Some functions are actually named
   something starting with __ and the normal name is an alias.  */
#ifdef __stub_$2 || defined __stub__$2
choke me
#endif

int
main ()
{
return $2 ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
eval "$3=yes"
else
eval "$3=no"
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
fi
eval ac_res=\${$3}
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }

```

```

eval $as_lineno_stack; ${as_lineno_stack:+} unset as_lineno

} # ac_fn_c_check_func

# ac_fn_c_check_header_mongrel LINENO HEADER VAR INCLUDES
# -----
# Tests whether HEADER exists, giving a warning if it cannot be
# compiled using
# the include files in INCLUDES and setting the cache variable VAR
# accordingly.
ac_fn_c_check_header_mongrel ()
{
    as_lineno=${as_lineno-"$1"}
    as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if eval \${$3+:} false; then :
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
    if eval \${$3+:} false; then :
        $as_echo_n "(cached) " >&6
    fi
    eval ac_res=\${$3}
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
    else
        # Is the header compilable?
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking $2 usability" >&5
$as_echo_n "checking $2 usability... " >&6; }
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
        /* end confdefs.h. */
        $4
        #include <$2>
        _ACEOF
        if ac_fn_c_try_compile "$LINENO"; then :
            ac_header_compiler=yes
        else
            ac_header_compiler=no
        fi
        rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_header_compiler"
>&5
$as_echo "$ac_header_compiler" >&6; }

        # Is the header present?
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking $2 presence" >&5
$as_echo_n "checking $2 presence... " >&6; }
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
        /* end confdefs.h. */
        #include <$2>
        _ACEOF
        if ac_fn_c_try_cpp "$LINENO"; then :
            ac_header_preproc=yes

```

```

else
  ac_header_preproc=no
fi
rm -f conftest.err conftest.i conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_header_preproc"
>&5
$as_echo "$ac_header_preproc" >&6; }

# So?  What about this header?
case $ac_header_compiler:$ac_header_preproc:$ac_c_preproc_warn_flag in
#((
  yes:no: )
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: accepted by
the compiler, rejected by the preprocessor!" >&5
$as_echo "$as_me: WARNING: $2: accepted by the compiler, rejected by
the preprocessor!" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: proceeding
with the compiler's result" >&5
$as_echo "$as_me: WARNING: $2: proceeding with the compiler's result"
>&2;}
    ;;
  no:yes:* )
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: present but
cannot be compiled" >&5
$as_echo "$as_me: WARNING: $2: present but cannot be compiled" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2:      check
for missing prerequisite headers?" >&5
$as_echo "$as_me: WARNING: $2:      check for missing prerequisite
headers?" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: see the
Autoconf documentation" >&5
$as_echo "$as_me: WARNING: $2: see the Autoconf documentation" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2:      section
\"Present But Cannot Be Compiled\"" >&5
$as_echo "$as_me: WARNING: $2:      section \"Present But Cannot Be
Compiled\"" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: proceeding
with the compiler's result" >&5
$as_echo "$as_me: WARNING: $2: proceeding with the compiler's result"
>&2;}
    ( $as_echo "## -----
----- ##
## Report this to
https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus&component=GLib
##
## -----
----- ##"
) | sed "s/^/$as_me: WARNING:      /" >&2
    ;;
esac
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }

```



```
/bin/universe          = `(/bin/universe) 2>/dev/null          || echo
unknown`
```

```
_ASUNAME
```

```
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  $as_echo "PATH: $as_dir"
done
IFS=$as_save_IFS

} >&5
```

```
cat >&5 <<_ACEOF
```

```
## ----- ##
## Core tests. ##
## ----- ##
```

```
_ACEOF
```

```
# Keep a trace of the command line.
# Strip out --no-create and --no-recursion so they do not pile up.
# Strip out --silent because we don't want to record it for future
runs.
# Also quote any args containing shell meta-characters.
# Make two passes to allow for proper duplicate-argument suppression.
ac_configure_args=
ac_configure_args0=
ac_configure_args1=
ac_must_keep_next=false
for ac_pass in 1 2
do
  for ac_arg
  do
    case $ac_arg in
      -no-create | --no-c* | -n | -no-recursion | --no-r*) continue ;;
      -q | -quiet | --quiet | --quie | --qui | --qu | --q \
      | -silent | --silent | --silen | --sile | --sil)
        continue ;;
      *\'*)
        ac_arg=`$as_echo "$ac_arg" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'/g"` ;;
    esac
    case $ac_pass in
      1) as_fn_append ac_configure_args0 " '$ac_arg' " ;;
      2)
        as_fn_append ac_configure_args1 " '$ac_arg' "
```

```

    if test $ac_must_keep_next = true; then
ac_must_keep_next=false # Got value, back to normal.
    else
case $ac_arg in
    *=* | --config-cache | -C | -disable-* | --disable-* \
    | -enable-* | --enable-* | -gas | --g* | -nfp | --nf* \
    | -q | -quiet | --q* | -silent | --sil* | -v | -verb* \
    | -with-* | --with-* | -without-* | --without-* | --x)
        case "$ac_configure_args0 " in
            "$ac_configure_args1"* " '$ac_arg' "*" ) continue ;;
        esac
        ;;
    -* ) ac_must_keep_next=true ;;
esac
fi
as_fn_append ac_configure_args " '$ac_arg'"
;;
esac
done
done
{ ac_configure_args0=; unset ac_configure_args0;}
{ ac_configure_args1=; unset ac_configure_args1;}

# When interrupted or exit'd, cleanup temporary files, and complete
# config.log. We remove comments because anyway the quotes in there
# would cause problems or look ugly.
# WARNING: Use '\'' to represent an apostrophe within the trap.
# WARNING: Do not start the trap code with a newline, due to a FreeBSD
4.0 bug.
trap 'exit_status=$?
    # Save into config.log some information that might help in
debugging.
    {
        echo

        $as_echo "## ----- ##"
## Cache variables. ##
## ----- ##"
        echo
        # The following way of writing the cache mishandles newlines in
values,
        (
            for ac_var in `(set) 2>&1 | sed -n '\''s/^\([a-zA-Z_][a-zA-Z0-
9_]*\)=.*/\1/p'\''`; do
                eval ac_val=\${$ac_var}
                case $ac_val in #(
                    *${as_nl}*)
                        case $ac_var in #(
                            *_cv_*) { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: cache
variable $ac_var contains a newline" >&5
$as_echo "$as_me: WARNING: cache variable $ac_var contains a newline"
>&2;} ;;

```



```

    esac
    case $ac_var in #(
    _ | IFS | as_nl) ;; #(
    BASH_ARGV | BASH_SOURCE) eval $ac_var= ;; #(
    *) { eval $ac_var=; unset $ac_var; } ;;
    esac ;;
esac
done
(set) 2>&1 |
case $as_nl` (ac_space='\' \'\''; set) 2>&1` in #(
*${as_nl}ac_space= \ *)
sed -n \
"s/'\'/'\'\'\\\\'\'\'\'/g;

s/^\([\$_as_cr_alnum]*_cv_[$_as_cr_alnum]*\\)=\\(.\\*\\)/\\1='\'\'\'2\'\'\'
/p"
    ;; #(
    *)
    sed -n "/^[\$_as_cr_alnum]*_cv_[$_as_cr_alnum]*=/p"
    ;;
esac |
sort
)
echo

$as_echo "## ----- ##"
## Output variables. ##
## ----- ##"
echo
for ac_var in $ac_subst_vars
do
    eval ac_val=\$$ac_var
    case $ac_val in
    *\'\'\'*) ac_val=`$as_echo "$ac_val" | sed
"s/'\'/'\'\'\\\\'\'\'\'/g`; ;
    esac
    $as_echo "$ac_var='\'\'$ac_val\'\'\'
done | sort
echo

if test -n "$ac_subst_files"; then
    $as_echo "## ----- ##"
## File substitutions. ##
## ----- ##"
echo
for ac_var in $ac_subst_files
do
    eval ac_val=\$$ac_var
    case $ac_val in
    *\'\'\'*) ac_val=`$as_echo "$ac_val" | sed
"s/'\'/'\'\'\\\\'\'\'\'/g`; ;
    esac

```

```

    $as_echo "$ac_var='\''$ac_val'\''"
    done | sort
    echo
fi

if test -s confdefs.h; then
    $as_echo "## ----- ##"
## confdefs.h. ##
## ----- ##"
    echo
    cat confdefs.h
    echo
fi
test "$ac_signal" != 0 &&
    $as_echo "$as_me: caught signal $ac_signal"
    $as_echo "$as_me: exit $exit_status"
} >&5
rm -f core *.core core.confptest.* &&
rm -f -r confptest* confdefs* conf$$* $ac_clean_files &&
exit $exit_status
' 0
for ac_signal in 1 2 13 15; do
    trap 'ac_signal='$ac_signal'; as_fn_exit 1' $ac_signal
done
ac_signal=0

# confdefs.h avoids OS command line length limits that DEFS can
# exceed.
rm -f -r confptest* confdefs.h

$as_echo "/* confdefs.h */" > confdefs.h

# Predefined preprocessor variables.

cat >>confdefs.h <<_ACEOF
#define PACKAGE_NAME "$PACKAGE_NAME"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_TARNAME "$PACKAGE_TARNAME"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_VERSION "$PACKAGE_VERSION"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_STRING "$PACKAGE_STRING"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_BUGREPORT "$PACKAGE_BUGREPORT"

```

```

_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_URL "$PACKAGE_URL"
_ACEOF

# Let the site file select an alternate cache file if it wants to.
# Prefer an explicitly selected file to automatically selected ones.
ac_site_file1=NONE
if test -n "$CONFIG_SITE"; then
  # We do not want a PATH search for config.site.
  case $CONFIG_SITE in #((
    -*) ac_site_file1=./$CONFIG_SITE;;
    */*) ac_site_file1=$CONFIG_SITE;;
    *) ac_site_file1=./$CONFIG_SITE;;
  esac
fi
for ac_site_file in $ac_site_file1
do
  test "x$ac_site_file" = xNONE && continue
  if test /dev/null != "$ac_site_file" && test -r "$ac_site_file";
then
  { $as_echo "$as_me:${as_lineno-$LINENO}: loading site script
$ac_site_file" >&5
$as_echo "$as_me: loading site script $ac_site_file" >&6;}
  sed 's/^/| /' "$ac_site_file" >&5
  . "$ac_site_file" \
    || { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in
\`$ac_pwd':" >&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error $? "failed to load site script $ac_site_file
See \`config.log' for more details" "$LINENO" 5; }
  fi
done

if test -r "$cache_file"; then
  # Some versions of bash will fail to source /dev/null (special files
  # actually), so we avoid doing that. DJGPP emulates it as a regular
  file.
  if test /dev/null != "$cache_file" && test -f "$cache_file"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: loading cache
$cache_file" >&5
$as_echo "$as_me: loading cache $cache_file" >&6;}
    case $cache_file in
      [\\/] * | ?:[\\/] * ) . "$cache_file";;
      *) . "$cache_file";;
    esac
  fi
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: creating cache $cache_file"
>&5

```

```

$as_echo "$as_me: creating cache $cache_file" >&6;}
  >$cache_file
fi

# Check that the precious variables saved in the cache have kept the
# same
# value.
ac_cache_corrupted=false
for ac_var in $ac_precious_vars; do
  eval ac_old_set=\$ac_cv_env_${ac_var}_set
  eval ac_new_set=\$ac_env_${ac_var}_set
  eval ac_old_val=\$ac_cv_env_${ac_var}_value
  eval ac_new_val=\$ac_env_${ac_var}_value
  case $ac_old_set,$ac_new_set in
    set,)
      { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' was
set to \`${ac_old_val}' in the previous run" >&5
$as_echo "$as_me: error: \`${ac_var}' was set to \`${ac_old_val}' in the
previous run" >&2;}
      ac_cache_corrupted=: ;;
    ,set)
      { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' was
not set in the previous run" >&5
$as_echo "$as_me: error: \`${ac_var}' was not set in the previous run"
>&2;}
      ac_cache_corrupted=: ;;
    ,);)
      *)
        if test "x$ac_old_val" != "x$ac_new_val"; then
          # differences in whitespace do not lead to failure.
          ac_old_val_w=`echo x $ac_old_val`
          ac_new_val_w=`echo x $ac_new_val`
          if test "$ac_old_val_w" != "$ac_new_val_w"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' has
changed since the previous run:" >&5
$as_echo "$as_me: error: \`${ac_var}' has changed since the previous
run:" >&2;}
            ac_cache_corrupted=:
          else
            { $as_echo "$as_me:${as_lineno-$LINENO}: warning: ignoring
whitespace changes in \`${ac_var}' since the previous run:" >&5
$as_echo "$as_me: warning: ignoring whitespace changes in \`${ac_var}'
since the previous run:" >&2;}
            eval $ac_var=\$ac_old_val
          fi
          { $as_echo "$as_me:${as_lineno-$LINENO}: former value:
\`${ac_old_val}'" >&5
$as_echo "$as_me: former value: \`${ac_old_val}'" >&2;}
          { $as_echo "$as_me:${as_lineno-$LINENO}: current value:
\`${ac_new_val}'" >&5
$as_echo "$as_me: current value: \`${ac_new_val}'" >&2;}
          fi;;

```

```

esac
# Pass precious variables to config.status.
if test "$ac_new_set" = set; then
  case $ac_new_val in
    *\'*) ac_arg=$ac_var=`$as_echo "$ac_new_val" | sed
"s/'/'\\\'\\\'\\\'\\\'/g"` ;;
    *) ac_arg=$ac_var=$ac_new_val ;;
  esac
  case " $ac_configure_args " in
    *\' $ac_arg \' *) ;; # Avoid dups. Use of quotes ensures
accuracy.
    *) as_fn_append ac_configure_args " \'$ac_arg\'" ;;
  esac
fi
done
if $ac_cache_corrupted; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':" >&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
  { $as_echo "$as_me:${as_lineno-$LINENO}: error: changes in the
environment can compromise the build" >&5
$as_echo "$as_me: error: changes in the environment can compromise the
build" >&2;}
  as_fn_error $? "run `make distclean' and/or `rm $cache_file' and
start over" "$LINENO" 5
fi
## ----- ##
## Main body of script. ##
## ----- ##

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

ac_aux_dir=
for ac_dir in "$srcdir" "$srcdir/.." "$srcdir/../../."; do
  if test -f "$ac_dir/install-sh"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/install-sh -c"
    break
  elif test -f "$ac_dir/install.sh"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/install.sh -c"
    break
  elif test -f "$ac_dir/shtool"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/shtool install -c"
    break
  fi
done

```

```

    fi
done
if test -z "$ac_aux_dir"; then
    as_fn_error $? "cannot find install-sh, install.sh, or shtool in
\"$srcdir\" \"$srcdir/..\" \"$srcdir/../../\"" "$LINENO" 5
fi

# These three variables are undocumented and unsupported,
# and are intended to be withdrawn in a future Autoconf release.
# They can cause serious problems if a builder's source tree is in a
directory
# whose full name contains unusual characters.
ac_config_guess="$SHELL $ac_aux_dir/config.guess" # Please don't use
this var.
ac_config_sub="$SHELL $ac_aux_dir/config.sub" # Please don't use this
var.
ac_configure="$SHELL $ac_aux_dir/configure" # Please don't use this
var.

# Make sure we can run config.sub.
$SHELL "$ac_aux_dir/config.sub" sun4 >/dev/null 2>&1 ||
    as_fn_error $? "cannot run $SHELL $ac_aux_dir/config.sub" "$LINENO"
5

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking build system type"
>&5
$as_echo_n "checking build system type... " >&6; }
if ${ac_cv_build+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_build_alias=$build_alias
test "x$ac_build_alias" = x &&
    ac_build_alias=`$SHELL "$ac_aux_dir/config.guess"`
test "x$ac_build_alias" = x &&
    as_fn_error $? "cannot guess build type; you must specify one"
"$LINENO" 5
ac_cv_build=`$SHELL "$ac_aux_dir/config.sub" $ac_build_alias` ||
    as_fn_error $? "$SHELL $ac_aux_dir/config.sub $ac_build_alias
failed" "$LINENO" 5
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_build" >&5
$as_echo "$ac_cv_build" >&6; }
case $ac_cv_build in
*-*-*) ;;
*) as_fn_error $? "invalid value of canonical build" "$LINENO" 5;;
esac
build=$ac_cv_build
ac_save_IFS=$IFS; IFS='- '
set x $ac_cv_build
shift

```

```

build_cpu=$1
build_vendor=$2
shift; shift
# Remember, the first character of IFS is used to create $*,
# except with old shells:
build_os=$*
IFS=$ac_save_IFS
case $build_os in *\ *) build_os=`echo "$build_os" | sed 's/ /-/g'`;
esac

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking host system type"
>&5
$as_echo_n "checking host system type... " >&6; }
if ${ac_cv_host+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "x$host_alias" = x; then
    ac_cv_host=$ac_cv_build
  else
    ac_cv_host=`$SHELL "$ac_aux_dir/config.sub" $host_alias` ||
      as_fn_error $? "$SHELL $ac_aux_dir/config.sub $host_alias failed"
"$LINENO" 5
  fi
fi

```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_host" >&5
$as_echo "$ac_cv_host" >&6; }
case $ac_cv_host in
*-*-*) ;;
*) as_fn_error $? "invalid value of canonical host" "$LINENO" 5;;
esac
host=$ac_cv_host
ac_save_IFS=$IFS; IFS='- '
set x $ac_cv_host
shift
host_cpu=$1
host_vendor=$2
shift; shift
# Remember, the first character of IFS is used to create $*,
# except with old shells:
host_os=$*
IFS=$ac_save_IFS
case $host_os in *\ *) host_os=`echo "$host_os" | sed 's/ /-/g'`;
esac

```

```

am__api_version='1.12'

```

```

# Find a good install program. We prefer a C program (faster),
# so one script is as good as another. But avoid the broken or

```

```

# incompatible versions:
# SysV /etc/install, /usr/sbin/install
# SunOS /usr/etc/install
# IRIX /sbin/install
# AIX /bin/install
# AmigaOS /C/install, which installs bootblocks on floppy discs
# AIX 4 /usr/bin/installbsd, which doesn't work without a -g flag
# AFS /usr/afsws/bin/install, which mishandles nonexistent args
# SVR4 /usr/ucb/install, which tries to use the nonexistent group
"staff"
# OS/2's system install, which has a completely different semantic
# ./install, which can be erroneously created by make from
./install.sh.
# Reject install programs that cannot install multiple files.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for a BSD-compatible
install" >&5
$as_echo_n "checking for a BSD-compatible install... " >&6; }
if test -z "$INSTALL"; then
if ${ac_cv_path_install+:} false; then :
  $as_echo_n "(cached) " >&6
else
  as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
  for as_dir in $PATH
  do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    # Account for people who put trailing slashes in PATH elements.
case $as_dir/ in #(
  ./ | ../ | /[cC]/* | \
  /etc/* | /usr/sbin/* | /usr/etc/* | /sbin/* | /usr/afsws/bin/* | \
  ?:[\\/]os2[\\/]install[\\/] * | ?:[\\/]OS2[\\/]INSTALL[\\/] * | \
  /usr/ucb/* ) ;;
*)
  # OSF1 and SCO ODT 3.0 have their own names for install.
  # Don't use installbsd from OSF since it installs stuff as root
  # by default.
  for ac_prog in ginstall scoinst install; do
    for ac_exec_ext in ' $ac_executable_extensions; do
      if as_fn_executable_p "$as_dir/$ac_prog$ac_exec_ext"; then
        if test $ac_prog = install &&
          grep dspmsg "$as_dir/$ac_prog$ac_exec_ext" >/dev/null 2>&1;
then
          # AIX install. It has an incompatible calling convention.
          :
          elif test $ac_prog = install &&
            grep pwplus "$as_dir/$ac_prog$ac_exec_ext" >/dev/null 2>&1;
then
            # program-specific install script used by HP pwplus--don't
            use.
            :
          else
            rm -rf conftest.one conftest.two conftest.dir

```



```

        echo one > confptest.one
        echo two > confptest.two
        mkdir confptest.dir
        if "$sas_dir/$sas_prog$sas_exec_ext" -c confptest.one
confptest.two "`pwd`/confptest.dir" &&
            test -s confptest.one && test -s confptest.two &&
            test -s confptest.dir/confptest.one &&
            test -s confptest.dir/confptest.two
        then
            ac_cv_path_install="$sas_dir/$sas_prog$sas_exec_ext -c"
            break 3
        fi
    fi
done
done
;;
esac

done
IFS=$sas_save_IFS

rm -rf confptest.one confptest.two confptest.dir

fi
if test "${ac_cv_path_install+set}" = set; then
    INSTALL=$ac_cv_path_install
else
    # As a last resort, use the slow shell script.  Don't cache a
    # value for INSTALL within a source directory, because that will
    # break other packages using the cache if that directory is
    # removed, or if the value is a relative name.
    INSTALL=$ac_install_sh
fi
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $INSTALL" >&5
$sas_echo "$INSTALL" >&6; }

# Use test -z because SunOS4 sh mishandles braces in ${var-val}.
# It thinks the first close brace ends the variable substitution.
test -z "$INSTALL_PROGRAM" && INSTALL_PROGRAM='${INSTALL}'

test -z "$INSTALL_SCRIPT" && INSTALL_SCRIPT='${INSTALL}'

test -z "$INSTALL_DATA" && INSTALL_DATA='${INSTALL} -m 644'

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking whether build
environment is sane" >&5
$sas_echo_n "checking whether build environment is sane... " >&6; }
# Reject unsafe characters in $srcdir or the absolute working
directory
# name.  Accept space and tab only in the latter.

```

```

am_lf='
'
case `pwd` in
  *[\\"#\$\&\'`$am_lf]*)
    as_fn_error $? "unsafe absolute working directory name" "$LINENO"
5;;
esac
case $srcdir in
  *[\\"#\$\&\'`$am_lf\ \]*)
    as_fn_error $? "unsafe srcdir value: '$srcdir'" "$LINENO" 5;;
esac

# Do 'set' in a subshell so we don't clobber the current shell's
# arguments.  Must try -L first in case configure is actually a
# symlink; some systems play weird games with the mod time of symlinks
# (eg FreeBSD returns the mod time of the symlink's containing
# directory).
if (
  am_has_slept=no
  for am_try in 1 2; do
    echo "timestamp, slept: $am_has_slept" > conftest.file
    set X `ls -Lt "$srcdir/configure" conftest.file 2> /dev/null`
    if test "$*" = "X"; then
      # -L didn't work.
      set X `ls -t "$srcdir/configure" conftest.file`
    fi
    if test "$*" != "X $srcdir/configure conftest.file" \
      && test "$*" != "X conftest.file $srcdir/configure"; then

      # If neither matched, then we have a broken ls.  This can happen
      # if, for instance, CONFIG_SHELL is bash and it inherits a
      # broken ls alias from the environment.  This has actually
      # happened.  Such a system could not be considered "sane".
      as_fn_error $? "ls -t appears to fail.  Make sure there is not a
broken
alias in your environment" "$LINENO" 5
    fi
    if test "$2" = conftest.file || test $am_try -eq 2; then
      break
    fi
    # Just in case.
    sleep 1
    am_has_slept=yes
  done
  test "$2" = conftest.file
)
then
  # Ok.
  :
else
  as_fn_error $? "newly created file is older than distributed files!
Check your system clock" "$LINENO" 5

```

```

fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: yes" >&5
$sas_echo "yes" >&6; }
# If we didn't sleep, we still need to ensure time stamps of
config.status and
# generated files are strictly newer.
am_sleep_pid=
if grep 'slept: no' conftest.file >/dev/null 2>&1; then
  ( sleep 1 ) &
  am_sleep_pid=$!
fi

rm -f conftest.file

test "$program_prefix" != NONE &&

program_transform_name="s^&$program_prefix&;$program_transform_name"
# Use a double $ so make ignores it.
test "$program_suffix" != NONE &&

program_transform_name="s\&$program_suffix&;$program_transform_name"
# Double any \ or $.
# By default was `s,x,x', remove it if useless.
ac_script='s/[\\$]/&&/g;s/;/s,x,x,$//'
program_transform_name=`$sas_echo "$program_transform_name" | sed
"$ac_script"`

# expand $ac_aux_dir to an absolute path
am_aux_dir=`cd $ac_aux_dir && pwd`

if test x"${MISSING+set}" != xset; then
  case $am_aux_dir in
*\ * | *\ *)
    MISSING="\${SHELL} \"$am_aux_dir/missing\"" ;;
*)
    MISSING="\${SHELL} $am_aux_dir/missing" ;;
esac
fi
# Use eval to expand $SHELL
if eval "$MISSING --run true"; then
  am_missing_run="$MISSING --run "
else
  am_missing_run=
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: WARNING: 'missing' script
is too old or missing" >&5
$sas_echo "$sas_me: WARNING: 'missing' script is too old or missing"
>&2;}
fi

if test x"${install_sh}" != xset; then
  case $am_aux_dir in
*\ * | *\ *)

```

```

    install_sh="\${SHELL} '$am_aux_dir/install-sh'" ;;
*)
    install_sh="\${SHELL} $am_aux_dir/install-sh"
esac
fi

# Installed binaries are usually stripped using 'strip' when the user
# run "make install-strip". However 'strip' might not be the right
# tool to use in cross-compilation environments, therefore Automake
# will honor the 'STRIP' environment variable to overrule this
program.
if test "$cross_compiling" != no; then
  if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}strip", so it can be a
    program name with args.
    set dummy ${ac_tool_prefix}strip; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_STRIP+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      if test -n "$STRIP"; then
        ac_cv_prog_STRIP="$STRIP" # Let the user override the test.
      else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_STRIP="${ac_tool_prefix}strip"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
STRIP=$ac_cv_prog_STRIP
if test -n "$STRIP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $STRIP" >&5
  $as_echo "$STRIP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
  $as_echo "no" >&6; }
fi

```

```

fi
if test -z "$ac_cv_prog_STRIP"; then
  ac_ct_STRIP=$STRIP
  # Extract the first word of "strip", so it can be a program name
  with args.
  set dummy strip; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_STRIP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_STRIP"; then
      ac_cv_prog_ac_ct_STRIP="$ac_ct_STRIP" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_STRIP="strip"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

    fi
  fi
  ac_ct_STRIP=$ac_cv_prog_ac_ct_STRIP
  if test -n "$ac_ct_STRIP"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_STRIP" >&5
    $as_echo "$ac_ct_STRIP" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
    $as_echo "no" >&6; }
  fi

  if test "x$ac_ct_STRIP" = x; then
    STRIP=":"
  else
    case $cross_compiling:$ac_tool_warned in
    yes:)
      { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
      not prefixed with host triplet" >&5
      $as_echo "$as_me: WARNING: using cross tools not prefixed with host
      triplet" >&2;}
    ac_tool_warned=yes ;;
  }

```

```

esac
    STRIP=$ac_ct_STRIP
    fi
else
    STRIP="$ac_cv_prog_STRIP"
fi

fi

INSTALL_STRIP_PROGRAM="\$(install_sh) -c -s"

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for a thread-safe
mkdir -p" >&5
$as_echo_n "checking for a thread-safe mkdir -p... " >&6; }
if test -z "$MKDIR_P"; then
    if ${ac_cv_path_mkdir+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
        for as_dir in $PATH$PATH_SEPARATOR/opt/sfw/bin
        do
            IFS=$as_save_IFS
            test -z "$as_dir" && as_dir=.
            for ac_prog in mkdir gmkdir; do
                for ac_exec_ext in '' $ac_executable_extensions; do
                    as_fn_executable_p "$as_dir/$ac_prog$ac_exec_ext" || continue
                    case `"$as_dir/$ac_prog$ac_exec_ext" --version 2>&1` in #(
                        'mkdir (GNU coreutils) '* | \
                        'mkdir (coreutils) '* | \
                        'mkdir (fileutils) '4.1*)
                        ac_cv_path_mkdir=$as_dir/$ac_prog$ac_exec_ext
                        break 3;;
                    esac
                done
            done
        done
        IFS=$as_save_IFS
    fi

    fi

    test -d ./--version && rmdir ./--version
    if test "${ac_cv_path_mkdir+set}" = set; then
        MKDIR_P="$ac_cv_path_mkdir -p"
    else
        # As a last resort, use the slow shell script. Don't cache a
        # value for MKDIR_P within a source directory, because that will
        # break other packages using the cache if that directory is
        # removed, or if the value is a relative name.
        MKDIR_P="$ac_install_sh -d"
    fi
fi

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $MKDIR_P" >&5
$as_echo "$MKDIR_P" >&6; }

```

```

for ac_prog in gawk mawk nawk awk
do
  # Extract the first word of "$ac_prog", so it can be a program name
  with args.
  set dummy $ac_prog; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_AWK+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$AWK"; then
      ac_cv_prog_AWK="$AWK" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_AWK="$ac_prog"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      AWK=$ac_cv_prog_AWK
      if test -n "$AWK"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $AWK" >&5
        $as_echo "$AWK" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi

      test -n "$AWK" && break
    done

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether ${MAKE-make}
sets \$(MAKE)" >&5
    $as_echo_n "checking whether ${MAKE-make} sets \$(MAKE)... " >&6; }
    set x ${MAKE-make}
    ac_make=`$as_echo "$2" | sed 's/+/p/g; s/[^a-zA-Z0-9_]/_/g'`
    if eval \${ac_cv_prog_make_${ac_make}_set+:} false; then :
      $as_echo_n "(cached) " >&6

```

```

else
  cat >confptest.make <<\_ACEOF
SHELL = /bin/sh
all:
  @echo '@@@%=%$(MAKE)=@@@%%'
\_ACEOF
# GNU make sometimes prints "make[1]: Entering ...", which would
confuse us.
case `${MAKE-make} -f confptest.make 2>/dev/null` in
  *@@@%=?*=@@%*%)
    eval ac_cv_prog_make_${ac_make}_set=yes;;
  *)
    eval ac_cv_prog_make_${ac_make}_set=no;;
esac
rm -f confptest.make
fi
if eval test \${ac_cv_prog_make_${ac_make}_set} = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
  SET_MAKE=
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  SET_MAKE="MAKE=${MAKE-make}"
fi

rm -rf .tst 2>/dev/null
mkdir .tst 2>/dev/null
if test -d .tst; then
  am__leading_dot=.
else
  am__leading_dot=_
fi
rmdir .tst 2>/dev/null

if test "`cd $srcdir && pwd`" != "`pwd`"; then
  # Use -I$(srcdir) only when $(srcdir) != ., so that make's output
  # is not polluted with repeated "-I."
  am__isrc=' -I$(srcdir)'
  # test to see if srcdir already configured
  if test -f $srcdir/config.status; then
    as_fn_error $? "source directory already configured; run \"make
distclean\" there first" "$LINENO" 5
  fi
fi

# test whether we have cygpath
if test -z "$CYGPATH_W"; then
  if (cygpath --version) >/dev/null 2>/dev/null; then
    CYGPATH_W='cygpath -w'
  else
    CYGPATH_W=echo
  fi
fi

```



```
fi
fi
```

```
# Define the identity of the package.
PACKAGE='dbus-glib'
VERSION='0.100.2'
```

```
cat >>confdefs.h <<_ACEOF
#define PACKAGE "$PACKAGE"
_ACEOF
```

```
cat >>confdefs.h <<_ACEOF
#define VERSION "$VERSION"
_ACEOF
```

```
# Some tools Automake needs.
```

```
ACLOCAL=${ACLOCAL-"${am_missing_run}aclocal-${am__api_version}"}
```

```
AUTOCONF=${AUTOCONF-"${am_missing_run}autoconf"}
```

```
AUTOMAKE=${AUTOMAKE-"${am_missing_run}automake-${am__api_version}"}
```

```
AUTOHEADER=${AUTOHEADER-"${am_missing_run}autoheader"}
```

```
MAKEINFO=${MAKEINFO-"${am_missing_run}makeinfo"}
```

```
# For better backward compatibility.  To be removed once Automake
1.9.x
# dies out for good.  For more background, see:
# <http://lists.gnu.org/archive/html/automake/2012-07/msg00001.html>
# <http://lists.gnu.org/archive/html/automake/2012-07/msg00014.html>
mkdir_p='${MKDIR_P}'
```

```
# We need awk for the "check" target.  The system "awk" is bad on
# some platforms.
# Always define AMTAR for backward compatibility.  Yes, it's still
used
# in the wild :-( We should find a proper way to deprecate it ...
AMTAR='${TAR-tar}'
```

```
am__tar='${TAR-tar} chof - "$$tardir"' am__untar='${TAR-tar} xf -'
```

```

ac_config_headers="$ac_config_headers config.h"

# Honor aclocal flags
ACLOCAL="$ACLOCAL $ACLOCAL_FLAGS"

## must come before we use the $USE_MAINTAINER_MODE variable later

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to enable
maintainer-specific portions of Makefiles" >&5
$as_echo_n "checking whether to enable maintainer-specific portions of
Makefiles... " >&6; }
  # Check whether --enable-maintainer-mode was given.
if test "${enable_maintainer_mode+set}" = set; then :
  enableval=$enable_maintainer_mode; USE_MAINTAINER_MODE=$enableval
else
  USE_MAINTAINER_MODE=no
fi

  { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$USE_MAINTAINER_MODE" >&5
$as_echo "$USE_MAINTAINER_MODE" >&6; }
  if test $USE_MAINTAINER_MODE = yes; then
    MAINTAINER_MODE_TRUE=
    MAINTAINER_MODE_FALSE='#'
  else
    MAINTAINER_MODE_TRUE='#'
    MAINTAINER_MODE_FALSE=
  fi

  MAINT=$MAINTAINER_MODE_TRUE

# Check whether --enable-silent-rules was given.
if test "${enable_silent_rules+set}" = set; then :
  enableval=$enable_silent_rules;
fi

case $enable_silent_rules in # (((
  yes) AM_DEFAULT_VERBOSITY=0;;
  no) AM_DEFAULT_VERBOSITY=1;;
  *) AM_DEFAULT_VERBOSITY=0;;
esac
am_make=${MAKE-make}
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $am_make
supports nested variables" >&5

```

```

$as_echo_n "checking whether $am_make supports nested variables... "
>&6; }
if ${am_cv_make_support_nested_variables+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if $as_echo 'TRUE=$(BAR$(V))
BAR0=false
BAR1=true
V=1
am__doit:
  @$(TRUE)
.PHONY: am__doit' | $am_make -f - >/dev/null 2>&1; then
  am_cv_make_support_nested_variables=yes
else
  am_cv_make_support_nested_variables=no
fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_make_support_nested_variables" >&5
$as_echo "$am_cv_make_support_nested_variables" >&6; }
if test $am_cv_make_support_nested_variables = yes; then
  AM_V='$(V) '
  AM_DEFAULT_V='$(AM_DEFAULT_VERBOSITY) '
else
  AM_V=$AM_DEFAULT_VERBOSITY
  AM_DEFAULT_V=$AM_DEFAULT_VERBOSITY
fi
AM_BACKSLASH='\ '

# libtool versioning - this applies to libdbus
#
# See
http://sources.redhat.com/autobook/autobook/autobook\_91.html#SEC91 for
details
#

## increment if the interface has additions, changes, removals.
LT_CURRENT=4

## increment any time the source changes; set to
## 0 if you increment CURRENT
LT_REVISION=2

## increment if any interfaces have been added; set to 0
## if any interfaces have been changed or removed. removal has
## precedence over adding, so set to 0 if both happened.
LT_AGE=2

```

```

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}gcc", so it can be a
  program name with args.
  set dummy ${ac_tool_prefix}gcc; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_CC+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$CC"; then
      ac_cv_prog_CC="$CC" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ` $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_CC="${ac_tool_prefix}gcc"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      CC=$ac_cv_prog_CC
      if test -n "$CC"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
        $as_echo "$CC" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi

      fi
      if test -z "$ac_cv_prog_CC"; then
        ac_ct_CC=$CC

```

```

# Extract the first word of "gcc", so it can be a program name with
args.
set dummy gcc; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_CC+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_CC"; then
    ac_cv_prog_ac_ct_CC="$ac_ct_CC" # Let the user override the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_ac_ct_CC="gcc"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
ac_ct_CC=$ac_cv_prog_ac_ct_CC
if test -n "$ac_ct_CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_CC" >&5
$as_echo "$ac_ct_CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_CC" = x; then
    CC=""
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    CC=$ac_ct_CC
  fi
else

```

```

    CC="$ac_cv_prog_CC"
fi

if test -z "$CC"; then
    if test -n "$ac_tool_prefix"; then
        # Extract the first word of "${ac_tool_prefix}cc", so it can be a
        program name with args.
        set dummy ${ac_tool_prefix}cc; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
        $as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_prog_CC+:} false; then :
            $as_echo_n "(cached) " >&6
        else
            if test -n "$CC"; then
                ac_cv_prog_CC="$CC" # Let the user override the test.
            else
                as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
                for as_dir in $PATH
                do
                    IFS=$as_save_IFS
                    test -z "$as_dir" && as_dir=.
                    for ac_exec_ext in ' ' $ac_executable_extensions; do
                        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                            ac_cv_prog_CC="${ac_tool_prefix}cc"
                            $as_echo "$as_me:${as_lineno-$LINENO}: found
                            $as_dir/$ac_word$ac_exec_ext" >&5
                            break 2
                        fi
                    done
                done
                IFS=$as_save_IFS
            fi
        fi
        CC=$ac_cv_prog_CC
        if test -n "$CC"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
            $as_echo "$CC" >&6; }
        else
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
            $as_echo "no" >&6; }
        fi
    fi
fi

if test -z "$CC"; then
    # Extract the first word of "cc", so it can be a program name with
    args.
    set dummy cc; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }

```

```

if ${ac_cv_prog_CC+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$CC"; then
    ac_cv_prog_CC="$CC" # Let the user override the test.
  else
    ac_prog_rejected=no
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    if test "$as_dir/$ac_word$ac_exec_ext" = "/usr/ucb/cc"; then
      ac_prog_rejected=yes
      continue
    fi
    ac_cv_prog_CC="cc"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

if test $ac_prog_rejected = yes; then
  # We found a bogon in the path, so make sure we never use it.
  set dummy $ac_cv_prog_CC
  shift
  if test $# != 0; then
    # We chose a different compiler from the bogus one.
    # However, it has the same basename, so the bogon will be chosen
    # first if we set CC to just the basename; use the full file name.
    shift
    ac_cv_prog_CC="$as_dir/$ac_word${1+' '}$@"
  fi
fi
fi
fi
fi
CC=$ac_cv_prog_CC
if test -n "$CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
  $as_echo "$CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
  $as_echo "no" >&6; }
fi

fi

```

```

if test -z "$CC"; then
  if test -n "$ac_tool_prefix"; then
    for ac_prog in cl.exe
    do
      # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
      be a program name with args.
      set dummy $ac_tool_prefix$ac_prog; ac_word=$2
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $as_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_CC+:} false; then :
        $as_echo_n "(cached) " >&6
      else
        if test -n "$CC"; then
          ac_cv_prog_CC="$CC" # Let the user override the test.
        else
          as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
          for as_dir in $PATH
          do
            IFS=$as_save_IFS
            test -z "$as_dir" && as_dir=.
            for ac_exec_ext in ' $ac_executable_extensions; do
              if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                ac_cv_prog_CC="$ac_tool_prefix$ac_prog"
                $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                break 2
              fi
            done
          done
          IFS=$as_save_IFS

          fi
          fi
          CC=$ac_cv_prog_CC
          if test -n "$CC"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
            $as_echo "$CC" >&6; }
          else
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
            $as_echo "no" >&6; }
          fi

          test -n "$CC" && break
        done
      fi
    if test -z "$CC"; then
      ac_ct_CC=$CC
      for ac_prog in cl.exe
      do
        # Extract the first word of "$ac_prog", so it can be a program name
        with args.

```



```

set dummy $ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_CC+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_CC"; then
    ac_cv_prog_ac_ct_CC="$ac_ct_CC" # Let the user override the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_ac_ct_CC="$ac_prog"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
ac_ct_CC=$ac_cv_prog_ac_ct_CC
if test -n "$ac_ct_CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_CC" >&5
$as_echo "$ac_ct_CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  test -n "$ac_ct_CC" && break
done

  if test "x$ac_ct_CC" = x; then
    CC=""
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    CC=$ac_ct_CC

```

```

    fi
fi

fi

test -z "$CC" && { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in
\`$ac_pwd':" >&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error $? "no acceptable C compiler found in $PATH
See \`config.log' for more details" "$LINENO" 5; }

# Provide some information about the compiler.
$as_echo "$as_me:${as_lineno-$LINENO}: checking for C compiler
version" >&5
set X $ac_compile
ac_compiler=$2
for ac_option in --version -v -V -qversion; do
  { { ac_try="$ac_compiler $ac_option >&5"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\`$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_compiler $ac_option >&5") 2>conftest.err
  ac_status=$?
  if test -s conftest.err; then
    sed '10a\
... rest of stderr output deleted ...
    10q' conftest.err >conftest.er1
    cat conftest.er1 >&5
  fi
  rm -f conftest.er1 conftest.err
  $as_echo "$as_me:${as_lineno-$LINENO}: \`$? = $ac_status" >&5
  test $ac_status = 0; }
done

cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
ac_clean_files_save=$ac_clean_files
ac_clean_files="$ac_clean_files a.out a.out.dSYM a.exe b.out"
# Try to create an executable without -o first, disregard a.out.

```

```

# It will help us diagnose broken compilers, and finding out an
intuition
# of exeext.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the C
compiler works" >&5
$as_echo_n "checking whether the C compiler works... " >&6; }
ac_link_default=`$as_echo "$ac_link" | sed 's/ -o *conftest[^\ ]*//'\`

# The possible output files:
ac_files="a.out confctest.exe confctest a.exe a_out.exe b.out
confctest.*"

ac_rmfiles=
for ac_file in $ac_files
do
  case $ac_file in
    *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj ) ;;
    * ) ac_rmfiles="$ac_rmfiles $ac_file";;
  esac
done
rm -f $ac_rmfiles

if { { ac_try="$ac_link_default"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link_default") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; }; then :
  # Autoconf-2.13 could set the ac_cv_exeext variable to `no'.
# So ignore a value of `no', otherwise this would lead to `EXEEXT =
no'
# in a Makefile. We should not override ac_cv_exeext if it was
cached,
# so that the user can short-circuit this test for compilers unknown
to
# Autoconf.
for ac_file in $ac_files ''
do
  test -f "$ac_file" || continue
  case $ac_file in
    *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj )
      ;;
    [ab].out )
      # We found the default executable, but exeext='' is most
      # certainly right.

```

```

        break;;
    *.* )
        if test "${ac_cv_exeext+set}" = set && test "$ac_cv_exeext" !=
no;
        then ;; else
            ac_cv_exeext=`expr "$ac_file" : '[^.]*\(\..\*\)`
        fi
        # We set ac_cv_exeext here because the later test for it is not
        # safe: cross compilers may not add the suffix if given an `-o'
        # argument, so we may need to know it at that point already.
        # Even if this section looks crufty: it has the advantage of
        # actually working.
        break;;
    * )
        break;;
esac
done
test "$ac_cv_exeext" = no && ac_cv_exeext=

else
    ac_file=''
fi
if test -z "$ac_file"; then :
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
$as_echo "$as_me: failed program was:" >&5
sed 's/^/| /' conftest.$ac_ext >&5

    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':" >&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error 77 "C compiler cannot create executables
See `config.log' for more details" "$LINENO" 5; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for C compiler
default output file name" >&5
$as_echo_n "checking for C compiler default output file name... " >&6;
}
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_file" >&5
$as_echo "$ac_file" >&6; }
ac_exeext=$ac_cv_exeext

rm -f -r a.out a.out.dSYM a.exe conftest$ac_cv_exeext b.out
ac_clean_files=$ac_clean_files_save
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for suffix of
executables" >&5
$as_echo_n "checking for suffix of executables... " >&6; }
if { { ac_try="$ac_link"
case "($ac_try" in
    *\`* | *\`* | *\`*) ac_try_echo=\$ac_try;;

```

```

*) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\\"$as_me:${as_lineno-$LINENO}: $ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; }; then :
  # If both `confptest.exe' and `confptest' are `present' (well,
  observable)
  # catch `confptest.exe'. For instance with Cygwin, `ls confptest' will
  # work properly (i.e., refer to `confptest.exe'), while it won't with
  # `rm'.
  for ac_file in confptest.exe confptest confptest.*; do
    test -f "$ac_file" || continue
    case $ac_file in
      *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj ) ;;
      *.* ) ac_cv_exeext=`expr "$ac_file" : '[^.]*(\..*)'`
        break;;
      * ) break;;
    esac
  done
else
  { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \"\$ac_pwd':"
>&5
  $as_echo "$as_me: error: in \"\$ac_pwd':" >&2;}
  as_fn_error $? "cannot compute suffix of executables: cannot compile
  and link
  See `config.log' for more details" "$LINENO" 5; }
fi
rm -f confptest confptest$ac_cv_exeext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_exeext" >&5
$as_echo "$ac_cv_exeext" >&6; }

rm -f confptest.$ac_ext
EXEEXT=$ac_cv_exeext
ac_exeext=$EXEEXT
cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */
#include <stdio.h>
int
main ()
{
FILE *f = fopen ("confptest.out", "w");
return ferror (f) || fclose (f) != 0;

;
return 0;
}
_ACEOF
ac_clean_files="$ac_clean_files confptest.out"

```



```

    $sas_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
rm -f confptest.o confptest.obj
if { { ac_try="$ac_compile"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: $ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_compile") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; }; then :
  for ac_file in confptest.o confptest.obj confptest.*; do
  test -f "$ac_file" || continue;
  case $ac_file in
    *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM ) ;;
    *) ac_cv_objext=`expr "$ac_file" : '.*\\.\\(.*\\)'`
      break;;
  esac
done
else
  $as_echo "$as_me: failed program was:" >&5
  sed 's/^/| /' confptest.$ac_ext >&5

{ { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':" >&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error $? "cannot compute suffix of object files: cannot compile
See `config.log' for more details" "$LINENO" 5; }
fi
rm -f confptest.$ac_cv_objext confptest.$ac_ext
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_objext" >&5
$as_echo "$ac_cv_objext" >&6; }
OBJEXT=$ac_cv_objext
ac_objext=$OBJEXT
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether we are using
the GNU C compiler" >&5

```

```

$as_echo_n "checking whether we are using the GNU C compiler... " >&6;
}
if ${ac_cv_c_compiler_gnu+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{
#ifdef __GNUC__
  choke me
#endif

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_compiler_gnu=yes
else
  ac_compiler_gnu=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
ac_cv_c_compiler_gnu=$ac_compiler_gnu

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_c_compiler_gnu" >&5
$as_echo "$ac_cv_c_compiler_gnu" >&6; }
if test $ac_compiler_gnu = yes; then
  GCC=yes
else
  GCC=
fi
ac_test_CFLAGS=${CFLAGS+set}
ac_save_CFLAGS=$CFLAGS
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CC accepts
-g" >&5
$as_echo_n "checking whether $CC accepts -g... " >&6; }
if ${ac_cv_prog_cc_g+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_save_c_werror_flag=$ac_c_werror_flag
  ac_c_werror_flag=yes
  ac_cv_prog_cc_g=no
  CFLAGS="-g"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int

```



```

main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_prog_cc_g=yes
else
    CFLAGS=""
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :

else
    ac_c_werror_flag=$ac_save_c_werror_flag
    CFLAGS="-g"
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_prog_cc_g=yes
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
    ac_c_werror_flag=$ac_save_c_werror_flag
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_prog_cc_g" >&5
$as_echo "$ac_cv_prog_cc_g" >&6; }
if test "$ac_test_CFLAGS" = set; then
    CFLAGS=$ac_save_CFLAGS

```

```

elif test $ac_cv_prog_cc_g = yes; then
  if test "$GCC" = yes; then
    CFLAGS="-g -O2"
  else
    CFLAGS="-g"
  fi
else
  if test "$GCC" = yes; then
    CFLAGS="-O2"
  else
    CFLAGS=
  fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $CC option to
accept ISO C89" >&5
$as_echo_n "checking for $CC option to accept ISO C89... " >&6; }
if ${ac_cv_prog_cc_c89+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_cv_prog_cc_c89=no
ac_save_CC=$CC
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdarg.h>
#include <stdio.h>
struct stat;
/* Most of the following tests are stolen from RCS 5.7's src/conf.sh.
*/
struct buf { int x; };
FILE * (*rcsopen) (struct buf *, struct stat *, int);
static char *e (p, i)
    char **p;
    int i;
{
    return p[i];
}
static char *f (char * (*g) (char **, int), char **p, ...)
{
    char *s;
    va_list v;
    va_start (v,p);
    s = g (p, va_arg (v,int));
    va_end (v);
    return s;
}

/* OSF 4.0 Compaq cc is some sort of almost-ANSI by default.  It has
function prototypes and stuff, but not '\xHH' hex character
constants.
These don't provoke an error unfortunately, instead are silently
treated
as 'x'.  The following induces an error, until -std is added to get

```

```

proper ANSI mode.  Curiously '\x00'!='x' always comes out true, for
an
array size at least.  It's necessary to write '\x00'==0 to get
something
that's true only with -std.  */
int osf4_cc_array ['\x00' == 0 ? 1 : -1];

/* IBM C 6 for AIX is almost-ANSI by default, but it replaces macro
parameters
inside strings and character constants.  */
#define FOO(x) 'x'
int xlc6_cc_array[FOO(a) == 'x' ? 1 : -1];

int test (int i, double x);
struct s1 {int (*f) (int a);};
struct s2 {int (*f) (double a);};
int pairnames (int, char **, FILE *(*) (struct buf *, struct stat *,
int), int, int);
int argc;
char **argv;
int
main ()
{
return f (e, argv, 0) != argv[0] || f (e, argv, 1) != argv[1];
;
return 0;
}
__ACEOF
for ac_arg in ' -qlanglvl=extc89 -qlanglvl=ansi -std \
-Ae "-Aa -D_HPUX_SOURCE" "-Xc -D__EXTENSIONS__"
do
CC="$ac_save_CC $ac_arg"
if ac_fn_c_try_compile "$LINENO"; then :
ac_cv_prog_cc_c89=$ac_arg
fi
rm -f core conftest.err conftest.$ac_objext
test "x$ac_cv_prog_cc_c89" != "xno" && break
done
rm -f conftest.$ac_ext
CC=$ac_save_CC

fi
# AC_CACHE_VAL
case "x$ac_cv_prog_cc_c89" in
x)
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: none needed" >&5
$as_echo "none needed" >&6; } ;;
xno)
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: unsupported" >&5
$as_echo "unsupported" >&6; } ;;
*)
CC="$CC $ac_cv_prog_cc_c89"

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_prog_cc_c89" >&5
$as_echo "$ac_cv_prog_cc_c89" >&6; } ;;
esac
if test "x$ac_cv_prog_cc_c89" != xno; then :

fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
DEPDIR="`${am__leading_dot}deps"

ac_config_commands="$ac_config_commands depfiles"

am_make=${MAKE-make}
cat > confinc << 'END'
am__doit:
    @echo this is the am__doit target
.PHONY: am__doit
END
# If we don't find an include directive, just comment out the code.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for style of include
used by $am_make" >&5
$as_echo_n "checking for style of include used by $am_make... " >&6; }
am__include="#"
am__quote=
_am_result=none
# First try GNU make style include.
echo "include confinc" > confmf
# Ignore all kinds of additional output from 'make'.
case ` $am_make -s -f confmf 2> /dev/null ` in #(
*the\ am__doit\ target*)
    am__include=include
    am__quote=
    _am_result=GNU
    ;;
esac
# Now try BSD make style include.
if test "$am__include" = "#"; then
    echo '.include "confinc"' > confmf
    case ` $am_make -s -f confmf 2> /dev/null ` in #(
*the\ am__doit\ target*)
        am__include=.include
        am__quote=""
        _am_result=BSD
        ;;
    esac

```

```

fi

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $_am_result" >&5
$sas_echo "$_am_result" >&6; }
rm -f confinc confmf

# Check whether --enable-dependency-tracking was given.
if test "${enable_dependency_tracking+set}" = set; then :
  enableval=$enable_dependency_tracking;
fi

if test "x$enable_dependency_tracking" != xno; then
  am_depcomp="$ac_aux_dir/depcomp"
  AMDEPBACKSLASH='\'
  am__nodep='_no'
fi
if test "x$enable_dependency_tracking" != xno; then
  AMDEP_TRUE=
  AMDEP_FALSE='#'
else
  AMDEP_TRUE='#'
  AMDEP_FALSE=
fi

depcc="$CC"  am_compiler_list=

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking dependency style of
$depcc" >&5
$sas_echo_n "checking dependency style of $depcc... " >&6; }
if ${am_cv_CC_dependencies_compiler_type+set} false; then :
  $sas_echo_n "(cached) " >&6
else
  if test -z "$AMDEP_TRUE" && test -f "$am_depcomp"; then
    # We make a subdir and do the tests there.  Otherwise we can end up
    # making bogus files that we don't know about and never remove.  For
    # instance it was reported that on HP-UX the gcc test will end up
    # making a dummy file named 'D' -- because '-MD' means "put the
output
    # in D".
    rm -rf confptest.dir
    mkdir confptest.dir
    # Copy depcomp to subdir because otherwise we won't find it if we're
    # using a relative directory.
    cp "$am_depcomp" confptest.dir
    cd confptest.dir
    # We will build objects and dependencies in a subdirectory because
    # it helps to detect inapplicable dependency modes.  For instance
    # both Tru64's cc and ICC support -MD to output dependencies as a
    # side effect of compilation, but ICC will put the dependencies in

```

```

# the current directory while Tru64 will put them in the object
# directory.
mkdir sub

am_cv_CC_dependencies_compiler_type=none
if test "$am_compiler_list" = ""; then
    am_compiler_list=`sed -n 's/^#*\([a-zA-Z0-9]*\))$/\1/p' <
./depcomp`
fi
am__universal=false
case " $depcc " in #(
    *\ -arch\ *\ -arch\ *) am__universal=true ;;
    esac

for depmode in $am_compiler_list; do
    # Setup a source with many dependencies, because some compilers
    # like to wrap large dependency lists on column 80 (with \), and
    # we should not choose a depcomp mode which is confused by this.
    #
    # We need to recreate these files for each test, as the compiler
may
    # overwrite some of them when testing with obscure command lines.
    # This happens at least with the AIX C compiler.
    : > sub/confctest.c
    for i in 1 2 3 4 5 6; do
        echo '#include "conf tst'$i'.h"' >> sub/confctest.c
        # Using ": > sub/conf tst$i.h" creates only sub/conf tst1.h with
        # Solaris 10 /bin/sh.
        echo '/* dummy */' > sub/conf tst$i.h
    done
    echo "${am__include} ${am__quote}sub/confctest.Po${am__quote}" >
confmf

    # We check with '-c' and '-o' for the sake of the "dashmstdout"
    # mode. It turns out that the SunPro C++ compiler does not
properly
    # handle '-M -o', and we need to detect this. Also, some Intel
    # versions had trouble with output in subdirs.
    am__obj=sub/confctest.${OBJEXT-o}
    am__minus_obj="-o $am__obj"
    case $depmode in
    gcc)
        # This depmode causes a compiler race in universal mode.
        test "$am__universal" = false || continue
        ;;
    nosideeffect)
        # After this tag, mechanisms are not by side-effect, so they'll
        # only be used when explicitly requested.
        if test "x$enable_dependency_tracking" = xyes; then
            continue
        else
            break
        fi
    esac
done

```

```

    fi
    ;;
msvc7 | msvc7msys | msvisualcpp | msvcmsys)
    # This compiler won't grok '-c -o', but also, the minuso test
has
    # not run yet.  These depmodes are late enough in the game, and
    # so weak that their functioning should not be impacted.
    am__obj=confptest.${OBJEXT-o}
    am__minus_obj=
    ;;
none) break ;;
esac
if depmode=$depmode \
    source=sub/confptest.c object=$am__obj \
    depfile=sub/confptest.Po tmpdepfile=sub/confptest.TPo \
    $SHELL ./depcomp $depcc -c $am__minus_obj sub/confptest.c \
    >/dev/null 2>confptest.err &&
    grep sub/confftst1.h sub/confptest.Po > /dev/null 2>&1 &&
    grep sub/confftst6.h sub/confptest.Po > /dev/null 2>&1 &&
    grep $am__obj sub/confptest.Po > /dev/null 2>&1 &&
    ${MAKE-make} -s -f confmf > /dev/null 2>&1; then
    # icc doesn't choke on unknown options, it will just issue
warnings
    # or remarks (even with -Werror).  So we grep stderr for any
message
    # that says an option was ignored or not supported.
    # When given -MP, icc 7.0 and 7.1 complain thusly:
    #   icc: Command line warning: ignoring option '-M'; no argument
required
    # The diagnosis changed in icc 8.0:
    #   icc: Command line remark: option '-MP' not supported
    if (grep 'ignoring option' confptest.err ||
        grep 'not supported' confptest.err) >/dev/null 2>&1; then ;;
else
    am_cv_CC_dependencies_compiler_type=$depmode
    break
    fi
    fi
done

    cd ..
    rm -rf confptest.dir
else
    am_cv_CC_dependencies_compiler_type=none
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_CC_dependencies_compiler_type" >&5
$as_echo "$am_cv_CC_dependencies_compiler_type" >&6; }
CCDEPMODE=depmode=$am_cv_CC_dependencies_compiler_type

```

```

if
  test "x$enable_dependency_tracking" != xno \
  && test "$am_cv_CC_dependencies_compiler_type" = gcc3; then
  am__fastdepCC_TRUE=
  am__fastdepCC_FALSE='#'
else
  am__fastdepCC_TRUE='#'
  am__fastdepCC_FALSE=
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for library
containing strerror" >&5
$as_echo_n "checking for library containing strerror... " >&6; }
if ${ac_cv_search_strerror+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_func_search_save_LIBS=$LIBS
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
  /* end confdefs.h. */

  /* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char strerror ();
int
main ()
{
return strerror ();
;
return 0;
}
_ACEOF
for ac_lib in ' cposix; do
  if test -z "$ac_lib"; then
    ac_res="none required"
  else
    ac_res=-l$ac_lib
    LIBS="-l$ac_lib $ac_func_search_save_LIBS"
  fi
  if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_search_strerror=$ac_res
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext
  if ${ac_cv_search_strerror+:} false; then :
  break
fi

```



```

done
if ${ac_cv_search_strerror+:} false; then :

else
  ac_cv_search_strerror=no
fi
rm conftest.$ac_ext
LIBS=$ac_func_search_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_search_strerror" >&5
$as_echo "$ac_cv_search_strerror" >&6; }
ac_res=$ac_cv_search_strerror
if test "$ac_res" != no; then :
  test "$ac_res" = "none required" || LIBS="$ac_res $LIBS"

fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to run the C
preprocessor" >&5
$as_echo_n "checking how to run the C preprocessor... " >&6; }
# On Suns, sometimes $CPP names a directory.
if test -n "$CPP" && test -d "$CPP"; then
  CPP=
fi
if test -z "$CPP"; then
  if ${ac_cv_prog_CPP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    # Double quotes because CPP needs to be expanded
    for CPP in "$CC -E" "$CC -E -traditional-cpp" "/lib/cpp"
    do
      ac_preproc_ok=false
      for ac_c_preproc_warn_flag in ' yes
do
  # Use a header file that comes with gcc, so configuring glibc
  # with a fresh cross-compiler works.
  # Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
  # <limits.h> exists even on freestanding compilers.
  # On the NeXT, cc -E runs the code through the compiler's parser,
  # not just through cpp. "Syntax error" is here to catch this case.
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
#ifdef __STDC__
# include <limits.h>
#else

```

```

# include <assert.h>
#endif

                Syntax error
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :

else
    # Broken: fails on valid input.
    continue
fi
rm -f confptest.err confptest.i confptest.$ac_ext

    # OK, works on sane cases.  Now check whether nonexistent headers
    # can be detected and how.
    cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h.  */
#include <ac_nonexistent.h>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
    # Broken: success on invalid input.
    continue
else
    # Passes both tests.
    ac_preproc_ok=:
    break
fi
rm -f confptest.err confptest.i confptest.$ac_ext

done
# Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
rm -f confptest.i confptest.err confptest.$ac_ext
if $ac_preproc_ok; then :
    break
fi

    done
    ac_cv_prog_CPP=$CPP

fi
    CPP=$ac_cv_prog_CPP
else
    ac_cv_prog_CPP=$CPP
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $CPP" >&5
$as_echo "$CPP" >&6; }
ac_preproc_ok=false
for ac_c_preproc_warn_flag in ' yes
do
    # Use a header file that comes with gcc, so configuring glibc
    # with a fresh cross-compiler works.
    # Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
    # <limits.h> exists even on freestanding compilers.

```

```

    # On the NeXT, cc -E runs the code through the compiler's parser,
    # not just through cpp. "Syntax error" is here to catch this case.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
#ifdef __STDC__
# include <limits.h>
#else
# include <assert.h>
#endif
                Syntax error

_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :

else
    # Broken: fails on valid input.
    continue
fi
rm -f conftest.err conftest.i conftest.$ac_ext

    # OK, works on sane cases.  Now check whether nonexistent headers
    # can be detected and how.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
#include <ac_nonexistent.h>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
    # Broken: success on invalid input.
    continue
else
    # Passes both tests.
    ac_preproc_ok=:
    break
fi
rm -f conftest.err conftest.i conftest.$ac_ext

done
# Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
rm -f conftest.i conftest.err conftest.$ac_ext
if $ac_preproc_ok; then :

else
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`${ac_pwd}':"
    >&5
    $as_echo "$as_me: error: in \`${ac_pwd}':" >&2;}
    as_fn_error $? "C preprocessor \`${CPP}` fails sanity check
    See \`${config.log}` for more details" "$LINENO" 5; }
fi

ac_ext=c
ac_cpp='${CPP} ${CPPFLAGS}'
ac_compile='${CC} -c ${CFLAGS} ${CPPFLAGS} conftest.$ac_ext >&5'

```

```

ac_link='$CC -o confptest$sac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
confptest.$sac_ext $LIBS >&5'
ac_compiler_gnu=$sac_cv_c_compiler_gnu

{ $sac_echo "$sac_me:${as_lineno-$LINENO}: checking for grep that
handles long lines and -e" >&5
$sac_echo_n "checking for grep that handles long lines and -e... " >&6;
}
if ${ac_cv_path_GREP+:} false; then :
  $sac_echo_n "(cached) " >&6
else
  if test -z "$GREP"; then
    ac_path_GREP_found=false
    # Loop through the user's path and test for each of PROGNAMES_LIST
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
    for as_dir in $PATH$PATH_SEPARATOR/usr/xpg4/bin
    do
      IFS=$as_save_IFS
      test -z "$as_dir" && as_dir=.
      for ac_prog in grep ggrep; do
        for ac_exec_ext in ' ' $sac_executable_extensions; do
          ac_path_GREP="$as_dir/$ac_prog$sac_exec_ext"
          as_fn_executable_p "$ac_path_GREP" || continue
        # Check for GNU ac_path_GREP and select it if it is found.
        # Check for GNU $ac_path_GREP
        case `"$ac_path_GREP" --version 2>&1` in
        *GNU*)
          ac_cv_path_GREP="$ac_path_GREP" ac_path_GREP_found=;;
        *)
          ac_count=0
          $sac_echo_n 0123456789 >"confptest.in"
          while :
          do
            cat "confptest.in" "confptest.in" >"confptest.tmp"
            mv "confptest.tmp" "confptest.in"
            cp "confptest.in" "confptest.nl"
            $sac_echo 'GREP' >> "confptest.nl"
            "$ac_path_GREP" -e 'GREP$' -e '-(cannot match)-' < "confptest.nl"
            >"confptest.out" 2>/dev/null || break
            diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
            as_fn_arith $ac_count + 1 && ac_count=$as_val
            if test $ac_count -gt ${ac_path_GREP_max-0}; then
              # Best one so far, save it but keep looking for a better one
              ac_cv_path_GREP="$ac_path_GREP"
              ac_path_GREP_max=$ac_count
            fi
            # 10*(2^10) chars as input seems more than enough
            test $ac_count -gt 10 && break
          done
          rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
        esac
      done
    done
  fi

```



```

    cp "confptest.in" "confptest.nl"
    $as_echo 'EGREP' >> "confptest.nl"
    "$ac_path_EGREP" 'EGREP$' < "confptest.nl" >"confptest.out"
2>/dev/null || break
diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
as_fn_arith $ac_count + 1 && ac_count=$as_val
if test $ac_count -gt ${ac_path_EGREP_max-0}; then
    # Best one so far, save it but keep looking for a better one
    ac_cv_path_EGREP="$ac_path_EGREP"
    ac_path_EGREP_max=$ac_count
fi
# 10*(2^10) chars as input seems more than enough
test $ac_count -gt 10 && break
done
rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
esac

    $ac_path_EGREP_found && break 3
done
done
done
IFS=$as_save_IFS
if test -z "$ac_cv_path_EGREP"; then
    as_fn_error $? "no acceptable egrep could be found in
$PATH$PATH_SEPARATOR/usr/xpg4/bin" "$LINENO" 5
fi
else
    ac_cv_path_EGREP=$EGREP
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_EGREP"
>&5
$as_echo "$ac_cv_path_EGREP" >&6; }
EGREP="$ac_cv_path_EGREP"

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for ANSI C header
files" >&5
$as_echo_n "checking for ANSI C header files... " >&6; }
if ${ac_cv_header_stdcl+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <float.h>

int

```

```

main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_header_stdc=yes
else
    ac_cv_header_stdc=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

if test $ac_cv_header_stdc = yes; then
    # SunOS 4.x string.h does not declare mem*, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <string.h>

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "memchr" >/dev/null 2>&1; then :

else
    ac_cv_header_stdc=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdc = yes; then
    # ISC 2.0.2 stdlib.h does not declare free, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdlib.h>

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "free" >/dev/null 2>&1; then :

else
    ac_cv_header_stdc=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdc = yes; then
    # /bin/cc in Irix-4.0.5 gets non-ANSI ctype macros unless using -
ansi.
    if test "$cross_compiling" = yes; then :

```

```

:
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <ctype.h>
#include <stdlib.h>
#if ((' ' & 0x0FF) == 0x020)
# define ISLOWER(c) ('a' <= (c) && (c) <= 'z')
# define TOUPPER(c) (ISLOWER(c) ? 'A' + ((c) - 'a') : (c))
#else
# define ISLOWER(c) \
    (('a' <= (c) && (c) <= 'i') \
    || ('j' <= (c) && (c) <= 'r') \
    || ('s' <= (c) && (c) <= 'z'))
# define TOUPPER(c) (ISLOWER(c) ? ((c) | 0x40) : (c))
#endif

#define XOR(e, f) (((e) && !(f)) || (!(e) && (f)))
int
main ()
{
  int i;
  for (i = 0; i < 256; i++)
    if (XOR (islower (i), ISLOWER (i))
        || toupper (i) != TOUPPER (i))
      return 2;
  return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :

else
  ac_cv_header_stdcl=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$ac_exeext \
  conftest.$ac_objext conftest.beam conftest.$ac_ext
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_header_stdcl"
>&5
$as_echo "$ac_cv_header_stdcl" >&6; }
if test $ac_cv_header_stdcl = yes; then

$as_echo "#define STDC_HEADERS 1" >>confdefs.h

fi

# Check whether --enable-tests was given.
if test "${enable_tests+set}" = set; then :

```



```
    enableval=$enable_tests; enable_tests=$enableval
else
    enable_tests=$USE_MAINTAINER_MODE
fi

# Check whether --enable-ansi was given.
if test "${enable_ansi+set}" = set; then :
    enableval=$enable_ansi; enable_ansi=$enableval
else
    enable_ansi=no
fi

# Check whether --enable-verbose-mode was given.
if test "${enable_verbose_mode+set}" = set; then :
    enableval=$enable_verbose_mode; enable_verbose_mode=$enableval
else
    enable_verbose_mode=$USE_MAINTAINER_MODE
fi

# Check whether --enable-asserts was given.
if test "${enable_asserts+set}" = set; then :
    enableval=$enable_asserts; enable_asserts=$enableval
else
    enable_asserts=$USE_MAINTAINER_MODE
fi

# Check whether --enable-checks was given.
if test "${enable_checks+set}" = set; then :
    enableval=$enable_checks; enable_checks=$enableval
else
    enable_checks=yes
fi

# Check whether --enable-gcov was given.
if test "${enable_gcov+set}" = set; then :
    enableval=$enable_gcov; enable_gcov=$enableval
else
    enable_gcov=no
fi

# Check whether --enable-bash-completion was given.
if test "${enable_bash_completion+set}" = set; then :
    enableval=$enable_bash_completion; enable_bash_completion=$enableval
else
    enable_bash_completion=yes
fi

# Check whether --with-test-socket-dir was given.
if test "${with_test_socket_dir+set}" = set; then :
    withval=$with_test_socket_dir;
fi
```

```

# Check whether --with-introspect-xml was given.
if test "${with_introspect_xml+set}" = set; then :
    withval=$with_introspect_xml;
fi

    if test x$enable_bash_completion = xyes; then
        DBUS_BASH_COMPLETION_TRUE=
        DBUS_BASH_COMPLETION_FALSE='#'
    else
        DBUS_BASH_COMPLETION_TRUE='#'
        DBUS_BASH_COMPLETION_FALSE=
    fi

if test x$enable_bash_completion = xyes; then

$as_echo "#define DBUS_BASH_COMPLETION 1" >>confdefs.h

fi

if test x$enable_verbose_mode = xyes; then

$as_echo "#define DBUS_ENABLE_VERBOSE_MODE 1" >>confdefs.h

fi

# Check whether --with-dbus-binding-tool was given.
if test "${with_dbus_binding_tool+set}" = set; then :
    withval=$with_dbus_binding_tool; DBUS_BINDING_TOOL=$withval
else
    DBUS_BINDING_TOOL=\$(top_builddir)/dbus/dbus-binding-tool
fi

    if test x$enable_tests = xyes; then
        DBUS_BUILD_TESTS_TRUE=
        DBUS_BUILD_TESTS_FALSE='#'
    else
        DBUS_BUILD_TESTS_TRUE='#'
        DBUS_BUILD_TESTS_FALSE=
    fi

if test x$enable_tests = xyes; then

```

```

$as_echo "#define DBUS_BUILD_TESTS 1" >>confdefs.h

fi

if test x$enable_verbose_mode = xyes; then

$as_echo "#define DBUS_ENABLE_VERBOSE_MODE 1" >>confdefs.h

fi
if test x$enable_asserts = xno; then

$as_echo "#define DBUS_DISABLE_ASSERT 1" >>confdefs.h

$as_echo "#define G_DISABLE_ASSERT 1" >>confdefs.h

fi
if test x$enable_checks = xno; then

$as_echo "#define DBUS_DISABLE_CHECKS 1" >>confdefs.h

$as_echo "#define G_DISABLE_CHECKS 1" >>confdefs.h

fi

#### gcc warning flags

if test "x$GCC" = "xyes"; then

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether gcc
understands -Wfloat-equal" >&5
$as_echo_n "checking whether gcc understands -Wfloat-equal... " >&6; }

    ac_save_CFLAGS="$CFLAGS"
    CFLAGS="$CFLAGS -Wfloat-equal"

    cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h.  */

    _ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cc_flag=yes
else
    ac_cc_flag=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$ac_save_CFLAGS"

    if test "x$ac_cc_flag" = "xyes"; then

```

```

    ac_flag_float_equal=yes
else
    ac_flag_float_equal=no
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cc_flag" >&5
$as_echo "$ac_cc_flag" >&6; }

```

```

    case " $CFLAGS " in
*[\ \ ]-Wall[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wall" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wchar-subscripts[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wchar-subscripts" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wmissing-declarations[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wmissing-declarations" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wmissing-prototypes[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wmissing-prototypes" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wnested-externs[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wnested-externs" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wpointer-arith[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wpointer-arith" ;;
esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wcast-align[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wcast-align" ;;
esac

```

```

if test "x$ac_flag_float_equal" = "xyes"; then
    case " $CFLAGS " in
*[\ \ ]-Wfloat-equal[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wfloat-equal" ;;
esac
fi

```

```

case " $CFLAGS " in
*[\ \ ]-Wsign-compare[\ \ ]*) ;;

```

```

*) CFLAGS="$CFLAGS -Wsign-compare" ;;
esac

# This one is special - it's not a warning override.
# http://bugs.freedesktop.org/show_bug.cgi?id=10599
# is the bug for DBus core.
case " $CFLAGS " in
*[\ \ ]-fno-strict-aliasing[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -fno-strict-aliasing" ;;
esac

if test "x$enable_ansi" = "xyes"; then
  case " $CFLAGS " in
  *[\ \ ]-ansi[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -ansi" ;;
  esac

  case " $CFLAGS " in
  *[\ \ ]-D_POSIX_C_SOURCE*) ;;
  *) CFLAGS="$CFLAGS -D_POSIX_C_SOURCE=199309L" ;;
  esac

  case " $CFLAGS " in
  *[\ \ ]-D_BSD_SOURCE[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -D_BSD_SOURCE" ;;
  esac

  case " $CFLAGS " in
  *[\ \ ]-pedantic[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -pedantic" ;;
  esac
fi
if test x$enable_gcov = xyesh; then
  case " $CFLAGS " in
  *[\ \ ]-fprofile-arcs[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -fprofile-arcs" ;;
  esac
  case " $CFLAGS " in
  *[\ \ ]-ftest-coverage[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -ftest-coverage" ;;
  esac

  ## remove optimization
  CFLAGS=`echo "$CFLAGS" | sed -e 's/-O[0-9]*//g'`
fi
else
if test x$enable_gcov = xyesh; then
  as_fn_error $? "--enable-gcov can only be used with gcc" "$LINENO"
5
fi
fi

```



```

$as_echo_n "checking how to print strings... " >&6; }
# Test print first, because it will be a builtin if present.
if test "X`( print -r -- -n ) 2>/dev/null`" = X-n && \
    test "X`print -r -- $ECHO 2>/dev/null`" = "X$ECHO"; then
    ECHO='print -r --'
elif test "X`printf %s $ECHO 2>/dev/null`" = "X$ECHO"; then
    ECHO='printf %s\n'
else
    # Use this function as a fallback that always works.
    func_fallback_echo ()
    {
        eval 'cat <<_LTECHO_EOF'
$1
_LTECHO_EOF'
    }
    ECHO='func_fallback_echo'
fi

# func_echo_all arg...
# Invoke $ECHO with all args, space-separated.
func_echo_all ()
{
    $ECHO ""
}

case "$ECHO" in
    printf*) { $as_echo "$as_me:${as_lineno-$LINENO}: result: printf"
>&5
$as_echo "printf" >&6; } ;;
    print*) { $as_echo "$as_me:${as_lineno-$LINENO}: result: print -r"
>&5
$as_echo "print -r" >&6; } ;;
    *) { $as_echo "$as_me:${as_lineno-$LINENO}: result: cat" >&5
$as_echo "cat" >&6; } ;;
esac

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for a sed that does
not truncate output" >&5

```

```

$as_echo_n "checking for a sed that does not truncate output... " >&6;
}
if ${ac_cv_path_SED+:} false; then :
  $as_echo_n "(cached) " >&6
else
ac_script=s/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa/bbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbb/
  for ac_i in 1 2 3 4 5 6 7; do
    ac_script="$ac_script$as_nl$ac_script"
  done
  echo "$ac_script" 2>/dev/null | sed 99q >confptest.sed
  { ac_script=; unset ac_script;}
  if test -z "$SED"; then
ac_path_SED_found=false
# Loop through the user's path and test for each of PROGMAME-LIST
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_prog in sed gsed; do
    for ac_exec_ext in '$ac_executable_extensions; do
      ac_path_SED="$as_dir/$ac_prog$ac_exec_ext"
      as_fn_executable_p "$ac_path_SED" || continue
# Check for GNU ac_path_SED and select it if it is found.
# Check for GNU $ac_path_SED
case `"$ac_path_SED" --version 2>&1` in
*GNU*)
  ac_cv_path_SED="$ac_path_SED" ac_path_SED_found=;;;
*)
  ac_count=0
  $as_echo_n 0123456789 >"confptest.in"
  while :
  do
    cat "confptest.in" "confptest.in" >"confptest.tmp"
    mv "confptest.tmp" "confptest.in"
    cp "confptest.in" "confptest.nl"
    $as_echo '' >> "confptest.nl"
    "$ac_path_SED" -f confptest.sed < "confptest.nl" >"confptest.out"
2>/dev/null || break
    diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
    as_fn_arith $ac_count + 1 && ac_count=$as_val
    if test $ac_count -gt ${ac_path_SED_max-0}; then
      # Best one so far, save it but keep looking for a better one
      ac_cv_path_SED="$ac_path_SED"
      ac_path_SED_max=$ac_count
    fi
    # 10*(2^10) chars as input seems more than enough
    test $ac_count -gt 10 && break
  done
  rm -f confptest.in confptest.tmp confptest.nl confptest.out;;

```



```

esac

    $ac_path_SED_found && break 3
done
done
done
IFS=$as_save_IFS
if test -z "$ac_cv_path_SED"; then
    as_fn_error $? "no acceptable sed could be found in \$PATH"
"$LINENO" 5
fi
else
    ac_cv_path_SED=$SED
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_SED" >&5
$as_echo "$ac_cv_path_SED" >&6; }
SED="$ac_cv_path_SED"
rm -f confptest.sed

test -z "$SED" && SED=sed
Xsed="$SED -e 1s/^X//"

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for fgrep" >&5
$as_echo_n "checking for fgrep... " >&6; }
if ${ac_cv_path_FGREP+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if echo 'ab*c' | $GREP -F 'ab*c' >/dev/null 2>&1
    then ac_cv_path_FGREP="$GREP -F"
    else
        if test -z "$FGREP"; then
            ac_path_FGREP_found=false
            # Loop through the user's path and test for each of PROGNAMES_LIST
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH$PATH_SEPARATOR/usr/xpg4/bin
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_prog in fgrep; do
                    for ac_exec_ext in ' $ac_executable_extensions; do

```

```

        ac_path_FGREP="$as_dir/$ac_prog$ac_exec_ext"
        as_fn_executable_p "$ac_path_FGREP" || continue
# Check for GNU ac_path_FGREP and select it if it is found.
# Check for GNU $ac_path_FGREP
case `"$ac_path_FGREP" --version 2>&1` in
*GNU*)
    ac_cv_path_FGREP="$ac_path_FGREP" ac_path_FGREP_found=;;;
*)
    ac_count=0
    $as_echo_n 0123456789 >"confptest.in"
    while :
    do
        cat "confptest.in" "confptest.in" >"confptest.tmp"
        mv "confptest.tmp" "confptest.in"
        cp "confptest.in" "confptest.nl"
        $as_echo 'FGREP' >> "confptest.nl"
        "$ac_path_FGREP" FGREP < "confptest.nl" >"confptest.out" 2>/dev/null
    || break
        diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
        as_fn_arith $ac_count + 1 && ac_count=$as_val
        if test $ac_count -gt ${ac_path_FGREP_max-0}; then
            # Best one so far, save it but keep looking for a better one
            ac_cv_path_FGREP="$ac_path_FGREP"
            ac_path_FGREP_max=$ac_count
        fi
        # 10*(2^10) chars as input seems more than enough
        test $ac_count -gt 10 && break
    done
    rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
esac

    $ac_cv_path_FGREP_found && break 3
done
done
done
IFS=$as_save_IFS
if test -z "$ac_cv_path_FGREP"; then
    as_fn_error $? "no acceptable fgrep could be found in
$PATH$PATH_SEPARATOR/usr/xpg4/bin" "$LINENO" 5
fi
else
    ac_cv_path_FGREP=$FGREP
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_FGREP"
>&5
$as_echo "$ac_cv_path_FGREP" >&6; }
FGREP="$ac_cv_path_FGREP"

```

```
test -z "$GREP" && GREP=grep
```

```
# Check whether --with-gnu-ld was given.
if test "${with_gnu_ld+set}" = set; then :
  withval=$with_gnu_ld; test "$withval" = no || with_gnu_ld=yes
else
  with_gnu_ld=no
fi

ac_prog=ld
if test "$GCC" = yes; then
  # Check if gcc -print-prog-name=ld gives a path.
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for ld used by
$CC" >&5
$as_echo_n "checking for ld used by $CC... " >&6; }
  case $host in
    *-*-mingw*)
      # gcc leaves a trailing carriage return which upsets mingw
      ac_prog=`($CC -print-prog-name=ld) 2>&5 | tr -d '\015'` ;;
    *)
      ac_prog=`($CC -print-prog-name=ld) 2>&5` ;;
  esac
  case $ac_prog in
    # Accept absolute paths.
    [[\/*] | ?:[\/*]*)
      re_direlt='/[^\/*][^\/*]*/\.\./'
      # Canonicalize the pathname of ld
      ac_prog=`$ECHO "$ac_prog" | $SED 's%\\%/%g'`
      while $ECHO "$ac_prog" | $GREP "$re_direlt" > /dev/null 2>&1; do
        ac_prog=`$ECHO $ac_prog | $SED "s%$re_direlt%/"`
      done
      test -z "$LD" && LD="$ac_prog"
    ;;
  "")
```

```

    # If it fails, then pretend we aren't using GCC.
    ac_prog=ld
    ;;
*)
    # If it is relative, then search for the first ld in PATH.
    with_gnu_ld=unknown
    ;;
esac
elif test "$with_gnu_ld" = yes; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for GNU ld" >&5
$as_echo_n "checking for GNU ld... " >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for non-GNU ld"
>&5
$as_echo_n "checking for non-GNU ld... " >&6; }
fi
if ${lt_cv_path_LD+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -z "$LD"; then
        lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
        for ac_dir in $PATH; do
            IFS="$lt_save_ifs"
            test -z "$ac_dir" && ac_dir=.
            if test -f "$ac_dir/$ac_prog" || test -f
"$ac_dir/$ac_prog$ac_exeext"; then
                lt_cv_path_LD="$ac_dir/$ac_prog"
                # Check to see if the program is GNU ld.  I'd rather use --
version,
                # but apparently some variants of GNU ld only accept -v.
                # Break only if it was the GNU/non-GNU ld that we prefer.
                case `"$lt_cv_path_LD" -v 2>&1 </dev/null` in
                    *GNU* | *'with BFD'*)
                        test "$with_gnu_ld" != no && break
                        ;;
                    *)
                        test "$with_gnu_ld" != yes && break
                        ;;
                esac
            fi
        done
        IFS="$lt_save_ifs"
    else
        lt_cv_path_LD="$LD" # Let the user override the test with a path.
    fi
fi

LD="$lt_cv_path_LD"
if test -n "$LD"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LD" >&5
$as_echo "$LD" >&6; }
else

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi
test -z "$LD" && as_fn_error $? "no acceptable ld found in \${PATH}"
"$LINENO" 5
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if the linker ($LD)
is GNU ld" >&5
$as_echo_n "checking if the linker ($LD) is GNU ld... " >&6; }
if ${lt_cv_prog_gnu_ld+:} false; then :
  $as_echo_n "(cached) " >&6
else
  # I'd rather use --version here, but apparently some GNU lds only
accept -v.
case `LD -v 2>&1 </dev/null` in
*GNU* | *'with BFD'*)
  lt_cv_prog_gnu_ld=yes
  ;;
*)
  lt_cv_prog_gnu_ld=no
  ;;
esac
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_prog_gnu_ld"
>&5
$as_echo "$lt_cv_prog_gnu_ld" >&6; }
with_gnu_ld=$lt_cv_prog_gnu_ld

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for BSD- or MS-
compatible name lister (nm)" >&5
$as_echo_n "checking for BSD- or MS-compatible name lister (nm)... "
>&6; }
if ${lt_cv_path_NM+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$NM"; then
    # Let the user override the test.
    lt_cv_path_NM="$NM"
  else
    lt_nm_to_check="${ac_tool_prefix}nm"
    if test -n "$ac_tool_prefix" && test "$build" = "$host"; then
      lt_nm_to_check="$lt_nm_to_check nm"
    fi
    for lt_tmp_nm in $lt_nm_to_check; do
      lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR

```

```

for ac_dir in $PATH /usr/ccs/bin/elf /usr/ccs/bin /usr/ucb /bin;
do
    IFS="$lt_save_ifs"
    test -z "$ac_dir" && ac_dir=.
    tmp_nm="$ac_dir/$lt_tmp_nm"
    if test -f "$tmp_nm" || test -f "$tmp_nm$ac_exeext" ; then
        # Check to see if the nm accepts a BSD-compatible flag.
        # Adding the `sed 1q' prevents false positives on HP-UX, which
says:
        # nm: unknown option "B" ignored
        # Tru64's nm complains that /dev/null is an invalid object file
case "$tmp_nm" -B /dev/null 2>&1 | sed '1q'` in
*/dev/null* | *'Invalid file or object type'*)
    lt_cv_path_NM="$tmp_nm -B"
    break
    ;;
*)
    case `"$tmp_nm" -p /dev/null 2>&1 | sed '1q'` in
*/dev/null*)
        lt_cv_path_NM="$tmp_nm -p"
        break
        ;;
*)
        lt_cv_path_NM=${lt_cv_path_NM="$tmp_nm"} # keep the first
match, but
        continue # so that we can try to find one that supports BSD
flags
        ;;
    esac
    ;;
esac
    fi
done
IFS="$lt_save_ifs"
done
: ${lt_cv_path_NM=no}
fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_path_NM" >&5
$as_echo "$lt_cv_path_NM" >&6; }
if test "$lt_cv_path_NM" != "no"; then
    NM="$lt_cv_path_NM"
else
    # Didn't find any BSD compatible name lister, look for dumpbin.
    if test -n "$DUMPBIN"; then :
        # Let the user override the test.
    else
        if test -n "$ac_tool_prefix"; then
            for ac_prog in dumpbin "link -dump"
            do
                # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
be a program name with args.

```

```

set dummy $ac_tool_prefix$ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_DUMPBIN+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$DUMPBIN"; then
    ac_cv_prog_DUMPBIN="$DUMPBIN" # Let the user override the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '$ac_executable_extensions'; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_DUMPBIN="$ac_tool_prefix$ac_prog"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
DUMPBIN=$ac_cv_prog_DUMPBIN
if test -n "$DUMPBIN"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DUMPBIN" >&5
$as_echo "$DUMPBIN" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  test -n "$DUMPBIN" && break
done
fi
if test -z "$DUMPBIN"; then
  ac_ct_DUMPBIN=$DUMPBIN
  for ac_prog in dumpbin "link -dump"
do
  # Extract the first word of "$ac_prog", so it can be a program name
  with args.
  set dummy $ac_prog; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_DUMPBIN+:} false; then :
    $as_echo_n "(cached) " >&6
  else

```

```

    if test -n "$ac_ct_DUMPBIN"; then
      ac_cv_prog_ac_ct_DUMPBIN="$ac_ct_DUMPBIN" # Let the user override
the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_DUMPBIN="$ac_prog"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      ac_ct_DUMPBIN=$ac_cv_prog_ac_ct_DUMPBIN
      if test -n "$ac_ct_DUMPBIN"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DUMPBIN" >&5
$as_echo "$ac_ct_DUMPBIN" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
      fi

      test -n "$ac_ct_DUMPBIN" && break
    done

    if test "x$ac_ct_DUMPBIN" = x; then
      DUMPBIN=":"
    else
      case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
      DUMPBIN=$ac_ct_DUMPBIN
    fi
  fi

  case ` $DUMPBIN -symbols /dev/null 2>&1 | sed '1q' ` in
*COFF*)

```



```

        DUMPBIN="$DUMPBIN -symbols"
        ;;
    *)
        DUMPBIN=:
        ;;
    esac
fi

if test "$DUMPBIN" != ":"; then
    NM="$DUMPBIN"
fi
fi
test -z "$NM" && NM=nm

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking the name lister
($NM) interface" >&5
$as_echo_n "checking the name lister ($NM) interface... " >&6; }
if ${lt_cv_nm_interface+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_nm_interface="BSD nm"
    echo "int some_variable = 0;" > conftest.$ac_ext
    (eval echo "\"\`$as_me:$LINENO: $ac_compile\"" >&5)
    (eval "$ac_compile" 2>conftest.err)
    cat conftest.err >&5
    (eval echo "\"\`$as_me:$LINENO: $NM \\\"conftest.$ac_objext\\\"\""
>&5)
    (eval "$NM \"conftest.$ac_objext\" 2>conftest.err > conftest.out)
    cat conftest.err >&5
    (eval echo "\"\`$as_me:$LINENO: output\"" >&5)
    cat conftest.out >&5
    if $GREP 'External.*some_variable' conftest.out > /dev/null; then
        lt_cv_nm_interface="MS dumpbin"
    fi
    rm -f conftest*
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_nm_interface"
>&5
$as_echo "$lt_cv_nm_interface" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether ln -s works"
>&5
$as_echo_n "checking whether ln -s works... " >&6; }
LN_S=$as_ln_s
if test "$LN_S" = "ln -s"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }

```

```

else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no, using $LN_S"
  >&5
  $as_echo "no, using $LN_S" >&6; }
fi

# find the maximum length of command line arguments
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking the maximum length
of command line arguments" >&5
$as_echo_n "checking the maximum length of command line arguments... "
>&6; }
if ${lt_cv_sys_max_cmd_len+:} false; then :
  $as_echo_n "(cached) " >&6
else
  i=0
  teststring="ABCD"

  case $build_os in
  msdosdjgpp*)
    # On DJGPP, this test can blow up pretty badly due to problems in
    libc
    # (any single argument exceeding 2000 bytes causes a buffer
    overrun
    # during glob expansion). Even if it were fixed, the result of
    this
    # check would be larger than it should be.
    lt_cv_sys_max_cmd_len=12288; # 12K is about right
    ;;

  gnu*)
    # Under GNU Hurd, this test is not required because there is
    # no limit to the length of command line arguments.
    # Libtool will interpret -1 as no limit whatsoever
    lt_cv_sys_max_cmd_len=-1;
    ;;

  cygwin* | mingw* | cegcc*)
    # On Win9x/ME, this test blows up -- it succeeds, but takes
    # about 5 minutes as the teststring grows exponentially.
    # Worse, since 9x/ME are not pre-emptively multitasking,
    # you end up with a "frozen" computer, even though with patience
    # the test eventually succeeds (with a max line length of 256k).
    # Instead, let's just punt: use the minimum linelength reported by
    # all of the supported platforms: 8192 (on NT/2K/XP).
    lt_cv_sys_max_cmd_len=8192;
    ;;

  mint*)
    # On MiNT this can take a long time and run out of memory.
    lt_cv_sys_max_cmd_len=8192;
    ;;

```

```

amigaos*)
# On AmigaOS with pdksh, this test takes hours, literally.
# So we just punt and use a minimum line length of 8192.
lt_cv_sys_max_cmd_len=8192;
;;

netbsd* | freebsd* | openbsd* | darwin* | dragonfly*)
# This has been around since 386BSD, at least. Likely further.
if test -x /sbin/sysctl; then
  lt_cv_sys_max_cmd_len=`/sbin/sysctl -n kern.argmax`
elif test -x /usr/sbin/sysctl; then
  lt_cv_sys_max_cmd_len=`/usr/sbin/sysctl -n kern.argmax`
else
  lt_cv_sys_max_cmd_len=65536      # usable default for all BSDs
fi
# And add a safety zone
lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
;;

interix*)
# We know the value 262144 and hardcode it with a safety zone
# (like BSD)
lt_cv_sys_max_cmd_len=196608
;;

os2*)
# The test takes a long time on OS/2.
lt_cv_sys_max_cmd_len=8192
;;

osf*)
# Dr. Hans Ekkehard Plesser reports seeing a kernel panic running
configure
# due to this test when exec_disable_arg_limit is 1 on Tru64. It
is not
# nice to cause kernel panics so lets avoid the loop below.
# First set a reasonable default.
lt_cv_sys_max_cmd_len=16384
#
if test -x /sbin/sysconfig; then
  case ` /sbin/sysconfig -q proc exec_disable_arg_limit` in
    *1*) lt_cv_sys_max_cmd_len=-1 ;;
  esac
fi
;;

sco3.2v5*)
lt_cv_sys_max_cmd_len=102400
;;

sysv5* | sco5v6* | sysv4.2uw2*)
kargmax=`grep ARG_MAX /etc/conf/cf.d/stune 2>/dev/null`
if test -n "$kargmax"; then

```

```

    lt_cv_sys_max_cmd_len=`echo $kargmax | sed 's/.*[          ]//'\`
else
    lt_cv_sys_max_cmd_len=32768
fi
;;
*)
lt_cv_sys_max_cmd_len=`(getconf ARG_MAX) 2> /dev/null`
if test -n "$lt_cv_sys_max_cmd_len"; then
    lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
    lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
else
    # Make teststring a little bigger before we do anything with it.
    # a 1K string should be a reasonable start.
    for i in 1 2 3 4 5 6 7 8 ; do
        teststring=$teststring$teststring
    done
    SHELL=${SHELL-${CONFIG_SHELL-/bin/sh}}
    # If test is not a shell built-in, we'll probably end up
computing a
    # maximum length that is only half of the actual maximum length,
but
    # we can't tell.
    while { test "X"`env echo "$teststring$teststring" 2>/dev/null`
\
        = "X$teststring$teststring"; } >/dev/null 2>&1 &&
        test $i != 17 # 1/2 MB should be enough
    do
        i=`expr $i + 1`
        teststring=$teststring$teststring
    done
    # Only check the string length outside the loop.
    lt_cv_sys_max_cmd_len=`expr "X$teststring" : ".*" 2>&1`
    teststring=
    # Add a significant safety factor because C++ compilers can tack
on
    # massive amounts of additional arguments before passing them to
the
    # linker. It appears as though 1/2 is a usable value.
    lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 2`
fi
;;
esac

fi

if test -n $lt_cv_sys_max_cmd_len ; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_sys_max_cmd_len" >&5
$as_echo "$lt_cv_sys_max_cmd_len" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: none" >&5
$as_echo "none" >&6; }

```

```

fi
max_cmd_len=${lt_cv_sys_max_cmd_len

: ${CP="cp -f"}
: ${MV="mv -f"}
: ${RM="rm -f"}

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the shell
understands some XSI constructs" >&5
$as_echo_n "checking whether the shell understands some XSI
constructs... " >&6; }
# Try some XSI features
xsi_shell=no
( _lt_dummy="a/b/c"
  test
"$_{lt_dummy}##*/},${_lt_dummy%/*},${_lt_dummy#??}"${_lt_dummy%$_lt_du
mmy"} , \
    = c,a/b,b/c, \
    && eval 'test $(( 1 + 1 )) -eq 2 \
    && test "${#_lt_dummy}" -eq 5' ) >/dev/null 2>&1 \
    && xsi_shell=yes
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $xsi_shell" >&5
$as_echo "$xsi_shell" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the shell
understands \"+=\"" >&5
$as_echo_n "checking whether the shell understands \"+=\"... " >&6; }
lt_shell_append=no
( foo=bar; set foo baz; eval "$1+=\$2" && test "$foo" = barbaz ) \
  >/dev/null 2>&1 \
  && lt_shell_append=yes
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_shell_append" >&5
$as_echo "$lt_shell_append" >&6; }

if ( (MAIL=60; unset MAIL) || exit) >/dev/null 2>&1; then
  lt_unset=unset
else
  lt_unset=false
fi

# test EBCDIC or ASCII

```

```

case `echo X|tr X '\101'` in
A) # ASCII based system
  # \n is not interpreted correctly by Solaris 8 /usr/ucb/tr
  lt_SP2NL='tr \040 \012'
  lt_NL2SP='tr \015\012 \040\040'
  ;;
*) # EBCDIC based system
  lt_SP2NL='tr \100 \n'
  lt_NL2SP='tr \r\n \100\100'
  ;;
esac

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to convert
$build file names to $host format" >&5
$as_echo_n "checking how to convert $build file names to $host
format... " >&6; }
if ${lt_cv_to_host_file_cmd+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $host in
  *-*-mingw* )
    case $build in
      *-*-mingw* ) # actually msys
        lt_cv_to_host_file_cmd=func_convert_file_msys_to_w32
        ;;
      *-*-cygwin* )
        lt_cv_to_host_file_cmd=func_convert_file_cygwin_to_w32
        ;;
      * ) # otherwise, assume *nix
        lt_cv_to_host_file_cmd=func_convert_file_nix_to_w32
        ;;
    esac
  ;;
  *-*-cygwin* )
    case $build in
      *-*-mingw* ) # actually msys
        lt_cv_to_host_file_cmd=func_convert_file_msys_to_cygwin
        ;;
      *-*-cygwin* )
        lt_cv_to_host_file_cmd=func_convert_file_noop
        ;;
      * ) # otherwise, assume *nix
        lt_cv_to_host_file_cmd=func_convert_file_nix_to_cygwin
        ;;
    esac
  ;;

```

```
    esac
    ;;
* ) # unhandled hosts (and "normal" native builds)
    lt_cv_to_host_file_cmd=func_convert_file_noop
    ;;
esac
```

```
fi
```

```
to_host_file_cmd=$lt_cv_to_host_file_cmd
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_to_host_file_cmd" >&5
$as_echo "$lt_cv_to_host_file_cmd" >&6; }
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to convert
$build file names to toolchain format" >&5
$as_echo_n "checking how to convert $build file names to toolchain
format... " >&6; }
```

```
if ${lt_cv_to_tool_file_cmd+:} false; then :
  $as_echo_n "(cached) " >&6
```

```
else
```

```
  #assume ordinary cross tools, or native build.
```

```
lt_cv_to_tool_file_cmd=func_convert_file_noop
```

```
case $host in
```

```
  *-*-mingw* )
```

```
    case $build in
```

```
      *-*-mingw* ) # actually msys
```

```
        lt_cv_to_tool_file_cmd=func_convert_file_msys_to_w32
```

```
        ;;
```

```
    esac
```

```
    ;;
```

```
esac
```

```
fi
```

```
to_tool_file_cmd=$lt_cv_to_tool_file_cmd
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_to_tool_file_cmd" >&5
$as_echo "$lt_cv_to_tool_file_cmd" >&6; }
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $LD option to
reload object files" >&5
```

```
$as_echo_n "checking for $LD option to reload object files... " >&6; }
```

```
if ${lt_cv_ld_reload_flag+:} false; then :
```

```

    $sas_echo_n "(cached) " >&6
else
    lt_cv_ld_reload_flag='-r'
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$lt_cv_ld_reload_flag" >&5
$sas_echo "$lt_cv_ld_reload_flag" >&6; }
reload_flag=$lt_cv_ld_reload_flag
case $reload_flag in
"" | " ") ;;
*) reload_flag=" $reload_flag" ;;
esac
reload_cmds='$LD$reload_flag -o $output$reload_objs'
case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
    if test "$GCC" != yes; then
        reload_cmds=false
    fi
    ;;
darwin*)
    if test "$GCC" = yes; then
        reload_cmds='$LTCC $LTCFLAGS -nostdlib ${wl}-r -o
$output$reload_objs'
    else
        reload_cmds='$LD$reload_flag -o $output$reload_objs'
    fi
    ;;
esac

```

```

if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}objdump", so it can be
    a program name with args.
    set dummy ${ac_tool_prefix}objdump; ac_word=$2
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $sas_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_OBJDUMP+:} false; then :
        $sas_echo_n "(cached) " >&6
    else
        if test -n "$OBJDUMP"; then
            ac_cv_prog_OBJDUMP="$OBJDUMP" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do

```



```

IFS=$as_save_IFS
test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
  ac_cv_prog_OBJDUMP="{ac_tool_prefix}objdump"
  $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
  break 2
fi
done
done
IFS=$as_save_IFS

fi
fi
OBJDUMP=$ac_cv_prog_OBJDUMP
if test -n "$OBJDUMP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OBJDUMP" >&5
$as_echo "$OBJDUMP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_OBJDUMP"; then
  ac_ct_OBJDUMP=$OBJDUMP
  # Extract the first word of "objdump", so it can be a program name
  with args.
  set dummy objdump; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_OBJDUMP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_OBJDUMP"; then
      ac_cv_prog_ac_ct_OBJDUMP="$ac_ct_OBJDUMP" # Let the user override
the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
          for ac_exec_ext in ' $ac_executable_extensions; do
if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
  ac_cv_prog_ac_ct_OBJDUMP="objdump"
  $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
  break 2
fi
fi

```

```

done
  done
IFS=$as_save_IFS

fi
fi
ac_ct_OBJDUMP=$ac_cv_prog_ac_ct_OBJDUMP
if test -n "$ac_ct_OBJDUMP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OBJDUMP" >&5
$as_echo "$ac_ct_OBJDUMP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_OBJDUMP" = x; then
    OBJDUMP="false"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    OBJDUMP=$ac_ct_OBJDUMP
  fi
else
  OBJDUMP="$ac_cv_prog_OBJDUMP"
fi

test -z "$OBJDUMP" && OBJDUMP=objdump

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to recognize
dependent libraries" >&5
$as_echo_n "checking how to recognize dependent libraries... " >&6; }
if ${lt_cv_deplibs_check_method+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_file_magic_cmd='$MAGIC_CMD'
lt_cv_file_magic_test_file=
lt_cv_deplibs_check_method='unknown'
# Need to set the preceding variable on all platforms that support

```

```

# interlibrary dependencies.
# 'none' -- dependencies not supported.
# `unknown' -- same as none, but documents that we really don't know.
# 'pass_all' -- all dependencies passed with no checks.
# 'test_compile' -- check by making test program.
# 'file_magic [[regex]]' -- check by looking for files in library path
# which responds to the $file_magic_cmd with a given extended regex.
# If you have `file' or equivalent on your system and you're not sure
# whether `pass_all' will *always* work, you probably want this one.

case $host_os in
aix[4-9]*)
    lt_cv_deplibs_check_method=pass_all
    ;;

beos*)
    lt_cv_deplibs_check_method=pass_all
    ;;

bsdi[45]*)
    lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(shared object|dynamic lib)'
    lt_cv_file_magic_cmd='/usr/bin/file -L'
    lt_cv_file_magic_test_file=/shlib/libc.so
    ;;

cygwin*)
    # func_win32_libid is a shell function defined in ltmain.sh
    lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86 DLL'
    lt_cv_file_magic_cmd='func_win32_libid'
    ;;

mingw* | pw32*)
    # Base MSYS/MinGW do not provide the 'file' command needed by
    # func_win32_libid shell function, so use a weaker test based on
    'objdump',
    # unless we find 'file', for example because we are cross-compiling.
    # func_win32_libid assumes BSD nm, so disallow it if using MS
    dumpbin.
    if ( test "$lt_cv_nm_interface" = "BSD nm" && file / ) >/dev/null
2>&1; then
        lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86
DLL'
        lt_cv_file_magic_cmd='func_win32_libid'
    else
        # Keep this pattern in sync with the one in func_win32_libid.
        lt_cv_deplibs_check_method='file_magic file format (pei*-
i386(.?architecture: i386)?|pe-arm-wince|pe-x86-64)'
        lt_cv_file_magic_cmd='$OBJDUMP -f'
    fi
    ;;

```

```

cegcc*)
  # use the weaker test based on 'objdump'. See mingw*.
  lt_cv_deplibs_check_method='file_magic file format pe-arm-
.*little(.*architecture: arm)?'
  lt_cv_file_magic_cmd='$OBJDUMP -f'
  ;;

darwin* | rhapsody*)
  lt_cv_deplibs_check_method=pass_all
  ;;

freebsd* | dragonfly*)
  if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
    case $host_cpu in
      i*86 )
        # Not sure whether the presence of OpenBSD here was a mistake.
        # Let's accept both of them until this is cleared up.
        lt_cv_deplibs_check_method='file_magic
(FreeBSD|OpenBSD|DragonFly)/i[3-9]86 (compact )?demand paged shared
library'
        lt_cv_file_magic_cmd=/usr/bin/file
        lt_cv_file_magic_test_file=`echo /usr/lib/libc.so.*`
        ;;
      esac
    else
      lt_cv_deplibs_check_method=pass_all
    fi
  ;;

gnu*)
  lt_cv_deplibs_check_method=pass_all
  ;;

haiku*)
  lt_cv_deplibs_check_method=pass_all
  ;;

hpux10.20* | hpux11*)
  lt_cv_file_magic_cmd=/usr/bin/file
  case $host_cpu in
    ia64*)
      lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|ELF-[0-
9][0-9]) shared object file - IA64'
      lt_cv_file_magic_test_file=/usr/lib/hpux32/libc.so
      ;;
    hppa*64*)
      lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|ELF[ -
][0-9][0-9]) (-bit)?( [LM]SB)? shared object( file)?[, -]* PA-RISC [0-
9]\.[0-9]'
      lt_cv_file_magic_test_file=/usr/lib/pa20_64/libc.sl
      ;;
  *)

```

```

    lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|PA-
RISC[0-9]\.[0-9]) shared library'
    lt_cv_file_magic_test_file=/usr/lib/libc.sl
    ;;
esac
;;

interix[3-9]*)
# PIC code is broken on Interix 3.x, that's why |\a not |_pic\a
here
lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so|\a)$'
;;

irix5* | irix6* | nonstopux*)
case $LD in
*-32|*" -32 ") libmagic=32-bit;;
*-n32|*" -n32 ") libmagic=N32;;
*-64|*" -64 ") libmagic=64-bit;;
*) libmagic=never-match;;
esac
lt_cv_deplibs_check_method=pass_all
;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu)
lt_cv_deplibs_check_method=pass_all
;;

netbsd*)
if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so\.[0-
9]+\.[0-9]+|_pic\a)$'
else
lt_cv_deplibs_check_method='match_pattern
/lib[^/]+(\.so|_pic\a)$'
fi
;;

newos6*)
lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(executable|dynamic lib)'
lt_cv_file_magic_cmd=/usr/bin/file
lt_cv_file_magic_test_file=/usr/lib/libnls.so
;;

*nto* | *qnx*)
lt_cv_deplibs_check_method=pass_all
;;

openbsd*)
if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then

```

```

    lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so\.[0-9]+|\.[0-9]+|\.so|_pic\.a)$'
else
    lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so\.[0-9]+|\.[0-9]+|_pic\.a)$'
fi
;;

osf3* | osf4* | osf5*)
    lt_cv_deplibs_check_method=pass_all
;;

rdos*)
    lt_cv_deplibs_check_method=pass_all
;;

solaris*)
    lt_cv_deplibs_check_method=pass_all
;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
    lt_cv_deplibs_check_method=pass_all
;;

sysv4 | sysv4.3*)
    case $host_vendor in
        motorola)
            lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(shared object|dynamic lib) M[0-9][0-9]* Version [0-9]'
            lt_cv_file_magic_test_file=`echo /usr/lib/libc.so*`
            ;;
        ncr)
            lt_cv_deplibs_check_method=pass_all
            ;;
        sequent)
            lt_cv_file_magic_cmd='/bin/file'
            lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [LM]SB
(shared object|dynamic lib )'
            ;;
        sni)
            lt_cv_file_magic_cmd='/bin/file'
            lt_cv_deplibs_check_method="file_magic ELF [0-9][0-9]*-bit [LM]SB
dynamic lib"
            lt_cv_file_magic_test_file=/lib/libc.so
            ;;
        siemens)
            lt_cv_deplibs_check_method=pass_all
            ;;
        pc)
            lt_cv_deplibs_check_method=pass_all
            ;;
    esac

```



```

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}dlltool", so it can be
  a program name with args.
  set dummy ${ac_tool_prefix}dlltool; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_DLLTOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$DLLTOOL"; then
      ac_cv_prog_DLLTOOL="$DLLTOOL" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_DLLTOOL="${ac_tool_prefix}dlltool"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      DLLTOOL=$ac_cv_prog_DLLTOOL
      if test -n "$DLLTOOL"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DLLTOOL" >&5
        $as_echo "$DLLTOOL" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi
    fi

    fi
    if test -z "$ac_cv_prog_DLLTOOL"; then
      ac_ct_DLLTOOL=$DLLTOOL
      # Extract the first word of "dlltool", so it can be a program name
      with args.
      set dummy dlltool; ac_word=$2
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $as_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_ac_ct_DLLTOOL+:} false; then :
        $as_echo_n "(cached) " >&6
      else

```



```

    if test -n "$ac_ct_DLLTOOL"; then
      ac_cv_prog_ac_ct_DLLTOOL="$ac_ct_DLLTOOL" # Let the user override
the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '$_ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_DLLTOOL="dlltool"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      ac_ct_DLLTOOL=$ac_cv_prog_ac_ct_DLLTOOL
      if test -n "$ac_ct_DLLTOOL"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DLLTOOL" >&5
$as_echo "$ac_ct_DLLTOOL" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
      fi

      if test "x$ac_ct_DLLTOOL" = x; then
        DLLTOOL="false"
      else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        DLLTOOL=$ac_ct_DLLTOOL
      fi
    else
      DLLTOOL="$ac_cv_prog_DLLTOOL"
    fi

test -z "$DLLTOOL" && DLLTOOL=dlltool

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to associate
runtime and link libraries" >&5
$as_echo_n "checking how to associate runtime and link libraries... "
>&6; }
if ${lt_cv_sharedlib_from_linklib_cmd+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_sharedlib_from_linklib_cmd='unknown'

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
  # two different shell functions defined in ltmain.sh
  # decide which to use based on capabilities of $DLLTOOL
  case ` $DLLTOOL --help 2>&1 ` in
*--identify-strict*)
  lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib
  ;;
*)
  lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib_fallback
  ;;
esac
  ;;
*)
  # fallback: assume linklib IS sharedlib
  lt_cv_sharedlib_from_linklib_cmd="$ECHO"
  ;;
esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_sharedlib_from_linklib_cmd" >&5
$as_echo "$lt_cv_sharedlib_from_linklib_cmd" >&6; }
sharedlib_from_linklib_cmd=$lt_cv_sharedlib_from_linklib_cmd
test -z "$sharedlib_from_linklib_cmd" &&
sharedlib_from_linklib_cmd=$ECHO

```

```

if test -n "$ac_tool_prefix"; then
  for ac_prog in ar

```

```

do
    # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
    be a program name with args.
    set dummy $ac_tool_prefix$ac_prog; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_AR+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$AR"; then
            ac_cv_prog_AR="$AR" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in ' ' $ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_AR="$ac_tool_prefix$ac_prog"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
                        $as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

            fi
            fi
            AR=$ac_cv_prog_AR
            if test -n "$AR"; then
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: $AR" >&5
                $as_echo "$AR" >&6; }
            else
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
                $as_echo "no" >&6; }
            fi

            test -n "$AR" && break
        done
    fi
    if test -z "$AR"; then
        ac_ct_AR=$AR
        for ac_prog in ar
        do
            # Extract the first word of "$ac_prog", so it can be a program name
            with args.
            set dummy $ac_prog; ac_word=$2
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
            $as_echo_n "checking for $ac_word... " >&6; }

```

```

if ${ac_cv_prog_ac_ct_AR+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_AR"; then
    ac_cv_prog_ac_ct_AR="$ac_ct_AR" # Let the user override the test.
  else
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
    for as_dir in $PATH
    do
      IFS=$as_save_IFS
      test -z "$as_dir" && as_dir=.
      for ac_exec_ext in '' $ac_executable_extensions; do
        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
          ac_cv_prog_ac_ct_AR="$ac_prog"
          $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
          break 2
        fi
      done
    done
    IFS=$as_save_IFS

    fi
    fi
    ac_ct_AR=$ac_cv_prog_ac_ct_AR
    if test -n "$ac_ct_AR"; then
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_AR" >&5
$as_echo "$ac_ct_AR" >&6; }
    else
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi

    test -n "$ac_ct_AR" && break
  done

  if test "x$ac_ct_AR" = x; then
    AR="false"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    AR=$ac_ct_AR
  fi
fi

```

```
: ${AR=ar}
: ${AR_FLAGS=cru}
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for archiver @FILE
support" >&5
$as_echo_n "checking for archiver @FILE support... " >&6; }
if ${lt_cv_ar_at_file+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_ar_at_file=no
  cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  echo conftest.$ac_objext > conftest.lst
  lt_ar_try='$AR $AR_FLAGS libconftest.a @conftest.lst >&5'
  { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$lt_ar_try\""; } >&5
  (eval $lt_ar_try) 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
  test $ac_status = 0; }
  if test "$ac_status" -eq 0; then
    # Ensure the archiver fails upon bogus file names.
    rm -f conftest.$ac_objext libconftest.a
    { { eval echo "\"\$as_me\":${as_lineno-$LINENO}: \"$lt_ar_try\"";
} >&5
  (eval $lt_ar_try) 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
  test $ac_status = 0; }
  if test "$ac_status" -ne 0; then
    lt_cv_ar_at_file=@
  fi
```

```

        fi
        rm -f conftest.* libconftest.a

fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_ar_at_file"
>&5
$as_echo "$lt_cv_ar_at_file" >&6; }

if test "x$lt_cv_ar_at_file" = xno; then
    archiver_list_spec=
else
    archiver_list_spec=$lt_cv_ar_at_file
fi
fi

if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}strip", so it can be a
    program name with args.
    set dummy ${ac_tool_prefix}strip; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_STRIP+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$STRIP"; then
            ac_cv_prog_STRIP="$STRIP" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '$ac_executable_extensions'; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_STRIP="${ac_tool_prefix}strip"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS
        fi
    fi
fi

```

```

fi
STRIP=$ac_cv_prog_STRIP
if test -n "$STRIP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $STRIP" >&5
$as_echo "$STRIP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_STRIP"; then
  ac_ct_STRIP=$STRIP
  # Extract the first word of "strip", so it can be a program name
  with args.
  set dummy strip; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_STRIP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_STRIP"; then
      ac_cv_prog_ac_ct_STRIP="$ac_ct_STRIP" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_STRIP="strip"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS
    fi
  fi
  ac_ct_STRIP=$ac_cv_prog_ac_ct_STRIP
  if test -n "$ac_ct_STRIP"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_STRIP" >&5
$as_echo "$ac_ct_STRIP" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  fi
fi

```

```

if test "x$ac_ct_STRIP" = x; then
  STRIP=":"
else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  STRIP=$ac_ct_STRIP
fi
else
  STRIP="$ac_cv_prog_STRIP"
fi

test -z "$STRIP" && STRIP=:

```

```

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}ranlib", so it can be
  a program name with args.
  set dummy ${ac_tool_prefix}ranlib; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_RANLIB+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$RANLIB"; then
      ac_cv_prog_RANLIB="$RANLIB" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_RANLIB="${ac_tool_prefix}ranlib"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS
    fi
  fi

```



```

fi
fi
RANLIB=$ac_cv_prog_RANLIB
if test -n "$RANLIB"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $RANLIB" >&5
$as_echo "$RANLIB" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_RANLIB"; then
  ac_ct_RANLIB=$RANLIB
  # Extract the first word of "ranlib", so it can be a program name
  with args.
  set dummy ranlib; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_RANLIB+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_RANLIB"; then
      ac_cv_prog_ac_ct_RANLIB="$ac_ct_RANLIB" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_RANLIB="ranlib"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

fi
fi
ac_ct_RANLIB=$ac_cv_prog_ac_ct_RANLIB
if test -n "$ac_ct_RANLIB"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_RANLIB" >&5
$as_echo "$ac_ct_RANLIB" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5

```

```

$as_echo "no" >&6; }
fi

if test "x$ac_ct_RANLIB" = x; then
  RANLIB=":"
else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  RANLIB=$ac_ct_RANLIB
fi
else
  RANLIB="$ac_cv_prog_RANLIB"
fi

test -z "$RANLIB" && RANLIB=:

```

```

# Determine commands to create old-style static archives.
old_archive_cmds='$AR $AR_FLAGS $oldlib$oldobjs'
old_postinstall_cmds='chmod 644 $oldlib'
old_postuninstall_cmds=

if test -n "$RANLIB"; then
  case $host_os in
openbsd*)
  old_postinstall_cmds="$old_postinstall_cmds~\`$RANLIB -t
\`$tool_oldlib"
  ;;
*)
  old_postinstall_cmds="$old_postinstall_cmds~\`$RANLIB
\`$tool_oldlib"
  ;;
esac
  old_archive_cmds="$old_archive_cmds~\`$RANLIB \`$tool_oldlib"
fi

case $host_os in
darwin*)
  lock_old_archive_extraction=yes ;;
*)
  lock_old_archive_extraction=no ;;
esac

```

```
# If no C compiler was specified, use CC.  
LTCC=${LTCC-"$CC"}
```

```
# If no C compiler flags were specified, use CFLAGS.  
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}
```

```
# Allow CC to be a program name with arguments.  
compiler=$CC
```

```
# Check for command to grab the raw symbol name followed by C symbol  
from nm.
```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking command to parse $NM
output from $compiler object" >&5
$as_echo_n "checking command to parse $NM output from $compiler
object... " >&6; }
if ${lt_cv_sys_global_symbol_pipe+:} false; then :
  $as_echo_n "(cached) " >&6
else

# These are sane defaults that work on at least a few old systems.
# [They come from Ultrix.  What could be older than Ultrix?!! ;)]

# Character class describing NM global symbol codes.
symcode=' [BCDEGRST] '

# Regexp to match symbols that can be accessed directly from C.
sympat='\([\_A-Za-z][\_A-Za-z0-9]*\) '

# Define system-specific variables.
case $host_os in
aix*)
  symcode=' [BCDT] '
  ;;
cygwin* | mingw* | pw32* | cegcc*)
  symcode=' [ABCDGISTW] '
  ;;
hpux*)
  if test "$host_cpu" = ia64; then
    symcode=' [ABCDEGRST] '
  fi
  ;;
irix* | nonstopux*)
  symcode=' [BCDEGRST] '
  ;;
osf*)
  symcode=' [BCDEGQRST] '
  ;;
solaris*)
  symcode=' [BDRT] '
  ;;
sco3.2v5*)
  symcode=' [DT] '
  ;;
sysv4.2uw2*)
  symcode=' [DT] '
  ;;
sysv5* | sco5v6* | unixware* | OpenUNIX*)
  symcode=' [ABDT] '
  ;;
sysv4)
  symcode=' [DFNSTU] '
  ;;
esac

```

```

# If we're using GNU nm, then use its standard symbol codes.
case `\$NM -V 2>&1` in
*GNU* | *'with BFD'*)
    symcode='[ABCDGIRSTW]' ;;
esac

# Transform an extracted symbol line into a proper C declaration.
# Some systems (esp. on ia64) link data and code symbols differently,
# so use this general approach.
lt_cv_sys_global_symbol_to_cdecl="sed -n -e 's/^T .* \(.*)$/extern
int \1();/p' -e 's/^\$symcode* .* \(.*)$/extern char \1;/p'"

# Transform an extracted symbol line into symbol name and symbol
address
lt_cv_sys_global_symbol_to_c_name_address="sed -n -e 's/^: \([^ ]*\)[
]*$/ {\\\"1\\\", (void *) 0},/p' -e 's/^\$symcode* \([^ ]*\) \([^
]*\)$/ {\"2\", (void *) \&2},/p'"
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix="sed -n -e 's/^:
\([^ ]*\)[ ]*$/ {\\\"1\\\", (void *) 0},/p' -e 's/^\$symcode* \([^
]*\) \(\lib\[^ ]*\)$/ {\"2\", (void *) \&2},/p' -e 's/^\$symcode* \([^
]*\) \([^ ]*\)$/ {\"lib2\", (void *) \&2},/p'"

# Handle CRLF in mingw tool chain
opt_cr=
case \$build_os in
mingw*)
    opt_cr='\$ECHO 'x\{0,1\}' | tr x '\015'` # option cr in regexp
    ;;
esac

# Try without a prefix underscore, then with it.
for ac_symprfx in "" "_"; do

    # Transform symcode, sympat, and symprfx into a raw symbol and a C
symbol.
    symxfrm="\\1 \$ac_symprfx\\2 \\2"

    # Write the raw and C identifiers.
    if test "\$lt_cv_nm_interface" = "MS dumpbin"; then
        # Fake it for dumpbin and say T for any non-static function
        # and D for any global variable.
        # Also find C++ and __fastcall symbols from MSVC++,
        # which start with @ or ?.
        lt_cv_sys_global_symbol_pipe="\$AWK '\
" {last_section=section; section=\$ 3};"\
" /^COFF SYMBOL TABLE/{for(i in hide) delete hide[i]};"\
" /Section length .*#relocs.*(pick any){hide[last_section]=1};"\
" \$ 0!~/External *|/{next};"\
" / 0+ UNDEF /{next}; / UNDEF \([^|]\)*()/ {next};"\
" {if(hide[section]) next};"\
" {f=0}; \$ 0~/\(\).*|/{f=1}; {printf f ? \"T \" : \"D \"};"\

```

```

"      {split(\$ 0, a, /\|\|\\r/); split(a[2], s)};"\
"      s[1]~/^[@?]/{print s[1], s[1]; next};"\
"      s[1]~prfx {split(s[1],t,\"@\""); print t[1],
substr(t[1],length(prfx))}"\
"      ' prfx=^$ac_symprfx"
    else
      lt_cv_sys_global_symbol_pipe="sed -n -e 's/^[^*[]\
](\$symcode$symcode*\)[ \t]*[ \t]*\
]*$ac_symprfx$sympat$opt_cr$/\$symxfrm/p'"
      fi
      lt_cv_sys_global_symbol_pipe="$lt_cv_sys_global_symbol_pipe | sed '/
__gnu_lto/d'"

# Check to see that the pipe works correctly.
pipe_works=no

rm -f conftest*
cat > conftest.$ac_ext <<_LT_EOF
#ifdef __cplusplus
extern "C" {
#endif
char nm_test_var;
void nm_test_func(void);
void nm_test_func(void){}
#ifdef __cplusplus
}
#endif
int main(){nm_test_var='a';nm_test_func();return(0);}
_LT_EOF

if { { eval echo "\$as_me\" : ${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; } then
  # Now try to grab the symbols.
  nlist=conftest.nm
  if { { eval echo "\$as_me\" : ${as_lineno-$LINENO}: \"$NM
conftest.$ac_objext \| \"$lt_cv_sys_global_symbol_pipe" \> $nlist\""; }
>&5
(eval $NM conftest.$ac_objext \| \"$lt_cv_sys_global_symbol_pipe" \>
$nlist) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; } && test -s "$nlist"; then
  # Try sorting and uniquifying the output.
  if sort "$nlist" | uniq > "$nlist.T"; then
    mv -f "$nlist.T" "$nlist"
  else
    rm -f "$nlist.T"
  fi
fi

```

```

    # Make sure that we snagged all the symbols we need.
    if $GREP ' nm_test_var$' "$nlist" >/dev/null; then
    if $GREP ' nm_test_func$' "$nlist" >/dev/null; then
        cat <<_LT_EOF > conftest.$ac_ext
/* Keep this code in sync between libtool.m4, ltmain, lt_system.h, and
tests. */
#if defined(_WIN32) || defined(__CYGWIN__) || defined(_WIN32_WCE)
/* DATA imports from DLLs on WIN32 can't be const, because runtime
relocations are performed -- see ld's documentation on pseudo-
relocs. */
# define LT_DLSYM_CONST
#elif defined(__osf__)
/* This system does not cope well with relocations in const data. */
# define LT_DLSYM_CONST
#else
# define LT_DLSYM_CONST const
#endif

#ifdef __cplusplus
extern "C" {
#endif

_LT_EOF
    # Now generate the symbol file.
    eval "$lt_cv_sys_global_symbol_to_cdecl" < "$nlist" | $GREP -v
main >> conftest.$ac_ext'

    cat <<_LT_EOF >> conftest.$ac_ext

/* The mapping between symbol names and symbols. */
LT_DLSYM_CONST struct {
    const char *name;
    void *address;
}
lt__PROGRAM__LTX_preloaded_symbols[] =
{
    { "@PROGRAM@", (void *) 0 },
_LT_EOF
    $SED "s/^\$symcode\$symcode* \\.*\) \\.*\)$/ {\\"2\", (void *)
\&2},/" < "$nlist" | $GREP -v main >> conftest.$ac_ext
    cat <<\_LT_EOF >> conftest.$ac_ext
    {0, (void *) 0}
};

/* This works around a problem in FreeBSD linker */
#ifdef FREEBSD_WORKAROUND
static const void *lt_preloaded_setup() {
    return lt__PROGRAM__LTX_preloaded_symbols;
}
#endif
#endif

```

```

#ifdef __cplusplus
}
#endif
_LT_EOF
    # Now try linking the two files.
    mv confptest.$ac_objext confftstm.$ac_objext
    lt_globsym_save_LIBS=$LIBS
    lt_globsym_save_CFLAGS=$CFLAGS
    LIBS="confftstm.$ac_objext"
    CFLAGS="$CFLAGS$lt_prog_compiler_no_built_in_flag"
    if { { eval echo "\`"$as_me`":${as_lineno-$LINENO}:
\"$ac_link\""; } >&5
    (eval $ac_link) 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
    test $ac_status = 0; } && test -s confptest${ac_exeext}; then
        pipe_works=yes
    fi
    LIBS=$lt_globsym_save_LIBS
    CFLAGS=$lt_globsym_save_CFLAGS
    else
        echo "cannot find nm_test_func in $nlist" >&5
    fi
    else
        echo "cannot find nm_test_var in $nlist" >&5
    fi
    else
        echo "cannot run $lt_cv_sys_global_symbol_pipe" >&5
    fi
    else
        echo "$progname: failed program was:" >&5
        cat confptest.$ac_ext >&5
    fi
    rm -rf confptest* confftst*

    # Do not use the global_symbol_pipe unless it works.
    if test "$pipe_works" = yes; then
        break
    else
        lt_cv_sys_global_symbol_pipe=
    fi
done

fi

if test -z "$lt_cv_sys_global_symbol_pipe"; then
    lt_cv_sys_global_symbol_to_cdecl=
fi
if test -z
"$lt_cv_sys_global_symbol_pipe$lt_cv_sys_global_symbol_to_cdecl"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: failed" >&5
    $as_echo "failed" >&6; }

```



```
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: ok" >&5
  $as_echo "ok" >&6; }
fi

# Response file support.
if test "$lt_cv_nm_interface" = "MS dumpbin"; then
  nm_file_list_spec='@'
elif $NM --help 2>/dev/null | grep '[@]FILE' >/dev/null; then
  nm_file_list_spec='@'
fi
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for sysroot" >&5
$as_echo_n "checking for sysroot... " >&6; }
```

```
# Check whether --with-libtool-sysroot was given.
if test "${with_libtool_sysroot+set}" = set; then :
  withval=$with_libtool_sysroot;
else
  with_libtool_sysroot=no
fi
```

```
lt_sysroot=
case ${with_libtool_sysroot} in #(
  yes)
```

```

    if test "$GCC" = yes; then
        lt_sysroot=`$CC --print-sysroot 2>/dev/null`
    fi
    ;; #(
/*)
    lt_sysroot=`echo "$with_libtool_sysroot" | sed -e
"$sed_quote_subst"`
    ;; #(
no|'')
    ;; #(
*)
    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
${with_libtool_sysroot}" >&5
$as_echo "${with_libtool_sysroot}" >&6; }
    as_fn_error $? "The sysroot must be an absolute path." "$LINENO" 5
    ;;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: ${lt_sysroot:-no}"
>&5
$as_echo "${lt_sysroot:-no}" >&6; }

# Check whether --enable-libtool-lock was given.
if test "${enable_libtool_lock+set}" = set; then :
    enableval=$enable_libtool_lock;
fi

test "x$enable_libtool_lock" != xno && enable_libtool_lock=yes

# Some flags need to be propagated to the compiler or linker for good
# libtool support.
case $host in
ia64-*-hpux*)
    # Find out which ABI we are using.
    echo 'int i;' > conftest.$ac_ext
    if { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}:"
\"$ac_compile\""; } >&5
    (eval $ac_compile) 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
        case `usr/bin/file conftest.$ac_objext` in
            *ELF-32*)
                HPUX_IA64_MODE="32"
                ;;
            *ELF-64*)
                HPUX_IA64_MODE="64"
                ;;
        esac
    fi

```

```

    esac
fi
rm -rf confptest*
;;
*-*-irix6*)
# Find out which ABI we are using.
echo '#line '$LINENO' "configure"' > confptest.$ac_ext
if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; }; then
  if test "$lt_cv_prog_gnu_ld" = yes; then
    case ` /usr/bin/file confptest.$ac_objext` in
      *32-bit*)
        LD="{LD-ld} -melf32bsmip"
        ;;
      *N32*)
        LD="{LD-ld} -melf32bmipn32"
        ;;
      *64-bit*)
        LD="{LD-ld} -melf64bmip"
        ;;
    esac
  else
    case ` /usr/bin/file confptest.$ac_objext` in
      *32-bit*)
        LD="{LD-ld} -32"
        ;;
      *N32*)
        LD="{LD-ld} -n32"
        ;;
      *64-bit*)
        LD="{LD-ld} -64"
        ;;
    esac
  fi
fi
rm -rf confptest*
;;

x86_64-*kfreebsd*-gnu|x86_64-*linux*|ppc*-*linux*|powerpc*-*linux*| \
s390*-*linux*|s390*-*tpf*|sparc*-*linux*)
# Find out which ABI we are using.
echo 'int i;' > confptest.$ac_ext
if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; }; then

```

```

case ` /usr/bin/file conftest.o ` in
  *32-bit*)
    case $host in
      x86_64-*kfreebsd*-gnu)
        LD="${LD-ld} -m elf_i386_fbsd"
        ;;
      x86_64-*linux*)
        LD="${LD-ld} -m elf_i386"
        ;;
      ppc64-*linux*|powerpc64-*linux*)
        LD="${LD-ld} -m elf32ppclinux"
        ;;
      s390x-*linux*)
        LD="${LD-ld} -m elf_s390"
        ;;
      sparc64-*linux*)
        LD="${LD-ld} -m elf32_sparc"
        ;;
    esac
  ;;
  *64-bit*)
    case $host in
      x86_64-*kfreebsd*-gnu)
        LD="${LD-ld} -m elf_x86_64_fbsd"
        ;;
      x86_64-*linux*)
        LD="${LD-ld} -m elf_x86_64"
        ;;
      ppc*-*linux*|powerpc*-*linux*)
        LD="${LD-ld} -m elf64ppc"
        ;;
      s390*-*linux*|s390*-*tpf*)
        LD="${LD-ld} -m elf64_s390"
        ;;
      sparc*-*linux*)
        LD="${LD-ld} -m elf64_sparc"
        ;;
    esac
  ;;
esac
fi
rm -rf conftest*
;;

*-*-sco3.2v5*)
  # On SCO OpenServer 5, we need -belf to get full-featured binaries.
  SAVE_CFLAGS="$CFLAGS"
  CFLAGS="$CFLAGS -belf"
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the C
  compiler needs -belf" >&5
  $as_echo_n "checking whether the C compiler needs -belf... " >&6; }
  if ${lt_cv_cc_needs_belf+:} false; then :

```

```

    $as_echo_n "(cached) " >&6
else
    ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    lt_cv_cc_needs_belf=yes
else
    lt_cv_cc_needs_belf=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
    ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_cc_needs_belf"
>&5
$as_echo "$lt_cv_cc_needs_belf" >&6; }
if test x"$lt_cv_cc_needs_belf" != x"yes"; then
    # this is probably gcc 2.8.0, egcs 1.0 or newer; no need for -belf
    CFLAGS="$SAVE_CFLAGS"
fi
;;
*-*solaris*)
    # Find out which ABI we are using.
    echo 'int i;' > conftest.$ac_ext
    if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
        (eval $ac_compile) 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
        test $ac_status = 0; }; then

```

```

case ` /usr/bin/file conftest.o ` in
*64-bit*)
  case $lt_cv_prog_gnu_ld in
  yes*)
    case $host in
    i?86-*-solaris*)
      LD="${LD-ld} -m elf_x86_64"
      ;;
    sparc*-*-solaris*)
      LD="${LD-ld} -m elf64_sparc"
      ;;
    esac
    # GNU ld 2.21 introduced _sol2 emulations. Use them if
available.
    if ${LD-ld} -V | grep _sol2 >/dev/null 2>&1; then
      LD="${LD-ld}_sol2"
    fi
    ;;
  *)
    if ${LD-ld} -64 -r -o conftest2.o conftest.o >/dev/null 2>&1;
then
      LD="${LD-ld} -64"
    fi
    ;;
  esac
  ;;
esac
fi
rm -rf conftest*
;;
esac

need_locks="$enable_libtool_lock"

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}mt", so it can be a
program name with args.
  set dummy ${ac_tool_prefix}mt; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_MANIFEST_TOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$MANIFEST_TOOL"; then
      ac_cv_prog_MANIFEST_TOOL="$MANIFEST_TOOL" # Let the user override
the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.

```

```

        for ac_exec_ext in ' ' $ac_executable_extensions; do
        if as_fn_executable_p "$sas_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_MANIFEST_TOOL="{ac_tool_prefix}mt"
            $sas_echo "$sas_me:${as_lineno-$LINENO}: found
$sas_dir/$ac_word$ac_exec_ext" >&5
            break 2
        fi
    done
    done
IFS=$sas_save_IFS

fi
fi
MANIFEST_TOOL=$ac_cv_prog_MANIFEST_TOOL
if test -n "$MANIFEST_TOOL"; then
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $MANIFEST_TOOL" >&5
    $sas_echo "$MANIFEST_TOOL" >&6; }
else
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
    $sas_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_MANIFEST_TOOL"; then
    ac_ct_MANIFEST_TOOL=$MANIFEST_TOOL
    # Extract the first word of "mt", so it can be a program name with
    args.
    set dummy mt; ac_word=$2
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $sas_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_MANIFEST_TOOL+:} false; then :
        $sas_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_MANIFEST_TOOL"; then
            ac_cv_prog_ac_ct_MANIFEST_TOOL="$ac_ct_MANIFEST_TOOL" # Let the user
            override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$sas_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in ' ' $ac_executable_extensions; do
                    if as_fn_executable_p "$sas_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_ac_ct_MANIFEST_TOOL="mt"
                        $sas_echo "$sas_me:${as_lineno-$LINENO}: found
                        $sas_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
        done
    done

```

```

IFS=$as_save_IFS

fi
fi
ac_ct_MANIFEST_TOOL=$ac_cv_prog_ac_ct_MANIFEST_TOOL
if test -n "$ac_ct_MANIFEST_TOOL"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_ct_MANIFEST_TOOL" >&5
$as_echo "$ac_ct_MANIFEST_TOOL" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_MANIFEST_TOOL" = x; then
    MANIFEST_TOOL=":"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    MANIFEST_TOOL=$ac_ct_MANIFEST_TOOL
  fi
else
  MANIFEST_TOOL="$ac_cv_prog_MANIFEST_TOOL"
fi

test -z "$MANIFEST_TOOL" && MANIFEST_TOOL=mt
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $MANIFEST_TOOL is
a manifest tool" >&5
$as_echo_n "checking if $MANIFEST_TOOL is a manifest tool... " >&6; }
if ${lt_cv_path_manifest_tool+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_path_manifest_tool=no
  echo "$as_me:$LINENO: $MANIFEST_TOOL '-?'" >&5
  $MANIFEST_TOOL '-?' 2>conftest.err > conftest.out
  cat conftest.err >&5
  if $GREP 'Manifest Tool' conftest.out > /dev/null; then
    lt_cv_path_manifest_tool=yes
  fi
  rm -f conftest*
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_path_manifest_tool" >&5
$as_echo "$lt_cv_path_manifest_tool" >&6; }
if test "x$lt_cv_path_manifest_tool" != xyes; then
  MANIFEST_TOOL=:

```


fi

```
case $host_os in
    rhapsody* | darwin*)
        if test -n "$ac_tool_prefix"; then
            # Extract the first word of "${ac_tool_prefix}dsymutil", so it can
            be a program name with args.
            set dummy ${ac_tool_prefix}dsymutil; ac_word=$2
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
            $as_echo_n "checking for $ac_word... " >&6; }
            if ${ac_cv_prog_DSYMUTIL+:} false; then :
                $as_echo_n "(cached) " >&6
            else
                if test -n "$DSYMUTIL"; then
                    ac_cv_prog_DSYMUTIL="$DSYMUTIL" # Let the user override the test.
                else
                    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
                    for as_dir in $PATH
                    do
                        IFS=$as_save_IFS
                        test -z "$as_dir" && as_dir=.
                        for ac_exec_ext in '$ac_executable_extensions; do
                            if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                                ac_cv_prog_DSYMUTIL="${ac_tool_prefix}dsymutil"
                                $as_echo "$as_me:${as_lineno-$LINENO}: found
                                $as_dir/$ac_word$ac_exec_ext" >&5
                                break 2
                            fi
                        done
                    done
                    IFS=$as_save_IFS

                    fi
                    fi
                    DSYMUTIL=$ac_cv_prog_DSYMUTIL
                    if test -n "$DSYMUTIL"; then
                        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DSYMUTIL" >&5
                        $as_echo "$DSYMUTIL" >&6; }
                    else
                        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
                        $as_echo "no" >&6; }
                    fi

                    fi
                    fi
                    if test -z "$ac_cv_prog_DSYMUTIL"; then
                        ac_ct_DSYMUTIL=$DSYMUTIL
```

```

# Extract the first word of "dsymutil", so it can be a program name
with args.
set dummy dsymutil; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_DSYMUTIL+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_DSYMUTIL"; then
    ac_cv_prog_ac_ct_DSYMUTIL="$ac_ct_DSYMUTIL" # Let the user override
the test.
  else
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
    for as_dir in $PATH
    do
      IFS=$as_save_IFS
      test -z "$as_dir" && as_dir=.
      for ac_exec_ext in ' $ac_executable_extensions; do
        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
          ac_cv_prog_ac_ct_DSYMUTIL="dsymutil"
          $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
          break 2
        fi
      done
    done
    IFS=$as_save_IFS

    fi
    fi
    ac_ct_DSYMUTIL=$ac_cv_prog_ac_ct_DSYMUTIL
    if test -n "$ac_ct_DSYMUTIL"; then
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DSYMUTIL"
>&5
$as_echo "$ac_ct_DSYMUTIL" >&6; }
    else
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi

    if test "x$ac_ct_DSYMUTIL" = x; then
      DSYMUTIL=":"
    else
      case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
      DSYMUTIL=$ac_ct_DSYMUTIL

```

```

    fi
else
    DSYMUTIL="$ac_cv_prog_DSYMUTIL"
fi

    if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}nmedit", so it can be
a program name with args.
set dummy ${ac_tool_prefix}nmedit; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_NMEDIT+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -n "$NMEDIT"; then
        ac_cv_prog_NMEDIT="$NMEDIT" # Let the user override the test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_NMEDIT="${ac_tool_prefix}nmedit"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
    done
IFS=$as_save_IFS

fi
fi
NMEDIT=$ac_cv_prog_NMEDIT
if test -n "$NMEDIT"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $NMEDIT" >&5
$as_echo "$NMEDIT" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_NMEDIT"; then
    ac_ct_NMEDIT=$NMEDIT
    # Extract the first word of "nmedit", so it can be a program name
with args.
set dummy nmedit; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5

```

```

$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_NMEDIT+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_NMEDIT"; then
    ac_cv_prog_ac_ct_NMEDIT="$ac_ct_NMEDIT" # Let the user override the
test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '$ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_ac_ct_NMEDIT="nmedit"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_NMEDIT=$ac_cv_prog_ac_ct_NMEDIT
if test -n "$ac_ct_NMEDIT"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_NMEDIT" >&5
$as_echo "$ac_ct_NMEDIT" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_NMEDIT" = x; then
    NMEDIT=":"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    NMEDIT=$ac_ct_NMEDIT
  fi
else
  NMEDIT="$ac_cv_prog_NMEDIT"
fi

```

```

    if test -n "$ac_tool_prefix"; then
      # Extract the first word of "${ac_tool_prefix}lipo", so it can be a
      program name with args.
      set dummy ${ac_tool_prefix}lipo; ac_word=$2
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $as_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_LIPO+:} false; then :
        $as_echo_n "(cached) " >&6
      else
        if test -n "$LIPO"; then
          ac_cv_prog_LIPO="$LIPO" # Let the user override the test.
        else
          as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
          for as_dir in $PATH
          do
            IFS=$as_save_IFS
            test -z "$as_dir" && as_dir=.
            for ac_exec_ext in ' $ac_executable_extensions; do
              if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                ac_cv_prog_LIPO="${ac_tool_prefix}lipo"
                $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                break 2
              fi
            done
          done
          IFS=$as_save_IFS

          fi
          fi
          LIPO=$ac_cv_prog_LIPO
          if test -n "$LIPO"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LIPO" >&5
            $as_echo "$LIPO" >&6; }
          else
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
            $as_echo "no" >&6; }
          fi

          fi
          if test -z "$ac_cv_prog_LIPO"; then
            ac_ct_LIPO=$LIPO
            # Extract the first word of "lipo", so it can be a program name with
            args.
            set dummy lipo; ac_word=$2
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
            $as_echo_n "checking for $ac_word... " >&6; }
            if ${ac_cv_prog_ac_ct_LIPO+:} false; then :
              $as_echo_n "(cached) " >&6
            else
              if test -n "$ac_ct_LIPO"; then

```

```

    ac_cv_prog_ac_ct_LIPO="$ac_ct_LIPO" # Let the user override the
test.
else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_ac_ct_LIPO="lipo"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_LIPO=$ac_cv_prog_ac_ct_LIPO
if test -n "$ac_ct_LIPO"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_LIPO" >&5
$as_echo "$ac_ct_LIPO" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    if test "x$ac_ct_LIPO" = x; then
        LIPO=":"
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        LIPO=$ac_ct_LIPO
    fi
else
    LIPO="$ac_cv_prog_LIPO"
fi

    if test -n "$ac_tool_prefix"; then
        # Extract the first word of "${ac_tool_prefix}otool", so it can be a
program name with args.
set dummy ${ac_tool_prefix}otool; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5

```

```

$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_OTOOL+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$OTOOL"; then
    ac_cv_prog_OTOOL="$OTOOL" # Let the user override the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_OTOOL="${ac_tool_prefix}otool"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
OTOOL=$ac_cv_prog_OTOOL
if test -n "$OTOOL"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OTOOL" >&5
$as_echo "$OTOOL" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_OTOOL"; then
  ac_ct_OTOOL=$OTOOL
  # Extract the first word of "otool", so it can be a program name
  with args.
  set dummy otool; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_OTOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_OTOOL"; then
      ac_cv_prog_ac_ct_OTOOL="$ac_ct_OTOOL" # Let the user override the
      test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH

```

```

do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_ac_ct_OTOOL="otool"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_OTOOL=$ac_cv_prog_ac_ct_OTOOL
if test -n "$ac_ct_OTOOL"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OTOOL" >&5
$as_echo "$ac_ct_OTOOL" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_OTOOL" = x; then
    OTOOL=":"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    OTOOL=$ac_ct_OTOOL
  fi
else
  OTOOL="$ac_cv_prog_OTOOL"
fi

  if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}otool64", so it can be
    a program name with args.
    set dummy ${ac_tool_prefix}otool64; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_OTOOL64+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      if test -n "$OTOOL64"; then

```



```

    ac_cv_prog_OTOOL64="$OTOOL64" # Let the user override the test.
else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_OTOOL64="{ac_tool_prefix}otool64"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
OTOOL64=$ac_cv_prog_OTOOL64
if test -n "$OTOOL64"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OTOOL64" >&5
$as_echo "$OTOOL64" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_OTOOL64"; then
    ac_ct_OTOOL64=$OTOOL64
    # Extract the first word of "otool64", so it can be a program name
    with args.
    set dummy otool64; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_OTOOL64+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_OTOOL64"; then
            ac_cv_prog_ac_ct_OTOOL64="$ac_ct_OTOOL64" # Let the user override
            the test.
        else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then

```

```

        ac_cv_prog_ac_ct_OTOOL64="otool64"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_OTOOL64=$ac_cv_prog_ac_ct_OTOOL64
if test -n "$ac_ct_OTOOL64"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OTOOL64" >&5
$as_echo "$ac_ct_OTOOL64" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    if test "x$ac_ct_OTOOL64" = x; then
        OTOOL64=":"
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        OTOOL64=$ac_ct_OTOOL64
    fi
else
    OTOOL64="$ac_cv_prog_OTOOL64"
fi

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for -
single_module linker flag" >&5
$as_echo_n "checking for -single_module linker flag... " >&6; }
if ${lt_cv_apple_cc_single_mod+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_apple_cc_single_mod=no
  if test -z "${LT_MULTI_MODULE}"; then
    # By default we will add the -single_module flag. You can
override
    # by either setting the environment variable LT_MULTI_MODULE
    # non-empty at configure time, or by adding -multi_module to the
    # link flags.
    rm -rf libconfptest.dylib*
    echo "int foo(void){return 1;}" > confptest.c
    echo "$LTCC $LTCFLAGS $LDFLAGS -o libconfptest.dylib \
-dynamiclib -Wl,-single_module confptest.c" >&5
    $LTCC $LTCFLAGS $LDFLAGS -o libconfptest.dylib \
    -dynamiclib -Wl,-single_module confptest.c 2>confptest.err
    _lt_result=$?
    # If there is a non-empty error log, and "single_module"
    # appears in it, assume the flag caused a linker warning
    if test -s confptest.err && $GREP single_module confptest.err;
then
      cat confptest.err >&5
      # Otherwise, if the output was created with a 0 exit code from
      # the compiler, it worked.
      elif test -f libconfptest.dylib && test $_lt_result -eq 0; then
        lt_cv_apple_cc_single_mod=yes
      else
        cat confptest.err >&5
      fi
      rm -rf libconfptest.dylib*
      rm -f confptest.*
    fi
  fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_apple_cc_single_mod" >&5
$as_echo "$lt_cv_apple_cc_single_mod" >&6; }

```

```

        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for -
exported_symbols_list linker flag" >&5
$as_echo_n "checking for -exported_symbols_list linker flag... " >&6;
}
if ${lt_cv_ld_exported_symbols_list+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_ld_exported_symbols_list=no
  save_LDFLAGS=$LDFLAGS
  echo "_main" > conftest.sym
  LDFLAGS="$LDFLAGS -Wl,-exported_symbols_list,conftest.sym"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  lt_cv_ld_exported_symbols_list=yes
else
  lt_cv_ld_exported_symbols_list=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
  LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_ld_exported_symbols_list" >&5
$as_echo "$lt_cv_ld_exported_symbols_list" >&6; }

        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for -force_load
linker flag" >&5
$as_echo_n "checking for -force_load linker flag... " >&6; }
if ${lt_cv_ld_force_load+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_ld_force_load=no
  cat > conftest.c << _LT_EOF
int forced_loaded() { return 2;}
_LT_EOF
  echo "$LTCC $LTCFLAGS -c -o conftest.o conftest.c" >&5
  $LTCC $LTCFLAGS -c -o conftest.o conftest.c 2>&5
  echo "$AR cru libconftest.a conftest.o" >&5
  $AR cru libconftest.a conftest.o 2>&5
  echo "$RANLIB libconftest.a" >&5
  $RANLIB libconftest.a 2>&5

```

```

        cat > confptest.c << _LT_EOF
int main() { return 0;}
_LT_EOF
        echo "$LTCC $LTCFLAGS $LDFLAGS -o confptest confptest.c -Wl,-
force_load,./libconfptest.a" >&5
        $LTCC $LTCFLAGS $LDFLAGS -o confptest confptest.c -Wl,-
force_load,./libconfptest.a 2>confptest.err
        _lt_result=$?
        if test -s confptest.err && $GREP force_load confptest.err; then
            cat confptest.err >&5
        elif test -f confptest && test $_lt_result -eq 0 && $GREP
forced_load confptest >/dev/null 2>&1 ; then
            lt_cv_ld_force_load=yes
        else
            cat confptest.err >&5
        fi
        rm -f confptest.err libconfptest.a confptest confptest.c
        rm -rf confptest.dSYM

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_ld_force_load"
>&5
$as_echo "$lt_cv_ld_force_load" >&6; }
    case $host_os in
        rhapsody* | darwin1.[012])
            _lt_dar_allow_undefined='${wl}-undefined ${wl}suppress' ;;
        darwin1.*)
            _lt_dar_allow_undefined='${wl}-flat_namespace ${wl}-undefined
${wl}suppress' ;;
        darwin*) # darwin 5.x on
            # if running on 10.5 or later, the deployment target defaults
            # to the OS version, if on x86, and 10.4, the deployment
            # target defaults to 10.4. Don't you love it?
            case ${MACOSX_DEPLOYMENT_TARGET-10.0},$host in
                10.0,*86*-darwin8*|10.0,*-darwin[91]*)
                    _lt_dar_allow_undefined='${wl}-undefined ${wl}dynamic_lookup'
;;
                10.[012]*)
                    _lt_dar_allow_undefined='${wl}-flat_namespace ${wl}-undefined
${wl}suppress' ;;
                10.*)
                    _lt_dar_allow_undefined='${wl}-undefined ${wl}dynamic_lookup'
;;
            esac
        ;;
    esac
    if test "$lt_cv_apple_cc_single_mod" = "yes"; then
        _lt_dar_single_mod='$single_module'
    fi
    if test "$lt_cv_ld_exported_symbols_list" = "yes"; then
        _lt_dar_export_syms=' ${wl}-
exported_symbols_list,$output_objdir/${libname}-symbols.expsym'

```

```

else
    _lt_dar_export_syms='~$NMEDIT -s $output_objdir/${libname}-
symbols.expsym ${lib}'
fi
if test "$DSYMUTIL" != ":" && test "$lt_cv_ld_force_load" = "no";
then
    _lt_dsymutil='~$DSYMUTIL $lib || :'
else
    _lt_dsymutil=
fi
;;
esac

# On IRIX 5.3, sys/types and inttypes.h are conflicting.
for ac_header in sys/types.h sys/stat.h stdlib.h string.h memory.h
strings.h \
    inttypes.h stdint.h unistd.h
do :
    as_ac_Header=`$as_echo "ac_cv_header_$ac_header" | $as_tr_sh`
    ac_fn_c_check_header_compile "$LINENO" "$ac_header" "$as_ac_Header"
"$ac_includes_default"
"
if eval test \"x\$$as_ac_Header\" = x"yes"; then :
    cat >>confdefs.h <<_ACEOF
#define ` $as_echo "HAVE_$ac_header" | $as_tr_cpp` 1
_ACEOF
fi
done

for ac_header in dlfcn.h
do :
    ac_fn_c_check_header_compile "$LINENO" "dlfcn.h"
"ac_cv_header_dlfcn_h" "$ac_includes_default"
"
if test "x$ac_cv_header_dlfcn_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_DLFCN_H 1
_ACEOF
fi
done

# Set options

```

```

        enable_dlopen=no

enable_win32_dll=no

        # Check whether --enable-shared was given.
if test "${enable_shared+set}" = set; then :
    enableval=$enable_shared; p=${PACKAGE-default}
    case $enableval in
    yes) enable_shared=yes ;;
    no) enable_shared=no ;;
    *)
        enable_shared=no
        # Look at the argument we got.  We use all the common list
separators.
        lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
        for pkg in $enableval; do
            IFS="$lt_save_ifs"
            if test "X$pkg" = "X$p"; then
                enable_shared=yes
            fi
        done
        IFS="$lt_save_ifs"
        ;;
    esac
else
    enable_shared=yes
fi

        # Check whether --enable-static was given.
if test "${enable_static+set}" = set; then :
    enableval=$enable_static; p=${PACKAGE-default}
    case $enableval in
    yes) enable_static=yes ;;
    no) enable_static=no ;;
    *)
        enable_static=no
        # Look at the argument we got.  We use all the common list
separators.
        lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
        for pkg in $enableval; do

```

```

IFS="$lt_save_ifs"
if test "X$pkg" = "X$p"; then
    enable_static=yes
fi
done
IFS="$lt_save_ifs"
;;
esac
else
    enable_static=yes
fi

```

```

# Check whether --with-pic was given.
if test "${with_pic+set}" = set; then :
    withval=$with_pic; lt_p=${PACKAGE-default}
    case $withval in
        yes|no) pic_mode=$withval ;;
        *)
            pic_mode=default
            # Look at the argument we got.  We use all the common list
            separators.
            lt_save_ifs="$IFS"; IFS="{IFS}$PATH_SEPARATOR,"
            for lt_pkg in $withval; do
                IFS="$lt_save_ifs"
                if test "X$lt_pkg" = "X$lt_p"; then
                    pic_mode=yes
                fi
            done
            IFS="$lt_save_ifs"
            ;;
    esac
else
    pic_mode=default
fi

```

```

test -z "$pic_mode" && pic_mode=default

```



```

# Check whether --enable-fast-install was given.
if test "${enable_fast_install+set}" = set; then :
  enableval=$enable_fast_install; p=${PACKAGE-default}
  case $enableval in
    yes) enable_fast_install=yes ;;
    no) enable_fast_install=no ;;
    *)
      enable_fast_install=no
      # Look at the argument we got.  We use all the common list
separators.
      lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
      for pkg in $enableval; do
        IFS="$lt_save_ifs"
        if test "X$pkg" = "X$p"; then
          enable_fast_install=yes
        fi
      done
      IFS="$lt_save_ifs"
      ;;
  esac
else
  enable_fast_install=yes
fi

```

```

# This can be used to rebuild libtool when needed
LIBTOOL_DEPS="$ltmain"

```

```

# Always use our own libtool.
LIBTOOL='$(top_builddir)'
LIBTOOL="$LIBTOOL/${host_alias}-libtool"

```

```
test -z "$LN_S" && LN_S="ln -s"
```

```
if test -n "${ZSH_VERSION+set}" ; then  
    setopt NO_GLOB_SUBST  
fi
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for objdir" >&5  
$as_echo_n "checking for objdir... " >&6; }  
if ${lt_cv_objdir+:} false; then :  
    $as_echo_n "(cached) " >&6  
else  
    rm -f .libs 2>/dev/null  
    mkdir .libs 2>/dev/null  
    if test -d .libs; then  
        lt_cv_objdir=.libs  
    else  
        # MS-DOS does not allow filenames that begin with a dot.  
        lt_cv_objdir=_libs  
    fi  
    rmdir .libs 2>/dev/null
```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_objdir" >&5
$as_echo "$lt_cv_objdir" >&6; }
objdir=$lt_cv_objdir

cat >>confdefs.h <<_ACEOF
#define LT_OBJDIR "$lt_cv_objdir/"
_ACEOF

case $host_os in
aix3*)
  # AIX sometimes has problems with the GCC collect2 program.  For
  some
  # reason, if we set the COLLECT_NAMES environment variable, the
  problems
  # vanish in a puff of smoke.
  if test "X${COLLECT_NAMES+set}" != Xset; then
    COLLECT_NAMES=
    export COLLECT_NAMES
  fi
  ;;
esac

# Global variables:
ofile=${host_alias}-libtool
can_build_shared=yes

# All known linkers require a `.a' archive for static linking (except
MSVC,
# which needs '.lib').
libext=a

with_gnu_ld="$lt_cv_prog_gnu_ld"

old_CC="$CC"
old_CFLAGS="$CFLAGS"

# Set sane defaults for various variables
test -z "$CC" && CC=cc
test -z "$LTCC" && LTCC=$CC
test -z "$LTCFLAGS" && LTCFLAGS=$CFLAGS
test -z "$LD" && LD=ld
test -z "$ac_objext" && ac_objext=o

for cc_temp in $compiler""; do

```

```

case $cc_temp in
  compile | *[\//]compile | ccache | *[\//]ccache ) ;;
  distcc | *[\//]distcc | purify | *[\//]purify ) ;;
  \-*) ;;
  *) break;;
esac
done
cc_basename=`$ECHO "$cc_temp" | $SED "s%.*/%%; s%^\$host_alias-%%"

# Only perform the check for file, if the check method requires it
test -z "$MAGIC_CMD" && MAGIC_CMD=file
case $deplibs_check_method in
file_magic*)
  if test "$file_magic_cmd" = '$MAGIC_CMD'; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
${ac_tool_prefix}file" >&5
$as_echo_n "checking for ${ac_tool_prefix}file... " >&6; }
if ${lt_cv_path_MAGIC_CMD+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $MAGIC_CMD in
[\\/*] | ?:[\\/*]*)
    lt_cv_path_MAGIC_CMD="$MAGIC_CMD" # Let the user override the test
with a path.
    ;;
  *)
    lt_save_MAGIC_CMD="$MAGIC_CMD"
    lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
    ac_dummy="/usr/bin$PATH_SEPARATOR$PATH"
    for ac_dir in $ac_dummy; do
      IFS="$lt_save_ifs"
      test -z "$ac_dir" && ac_dir=.
      if test -f $ac_dir/${ac_tool_prefix}file; then
        lt_cv_path_MAGIC_CMD="$ac_dir/${ac_tool_prefix}file"
        if test -n "$file_magic_test_file"; then
          case $deplibs_check_method in
"file_magic" *)
            file_magic_regex=`expr "$deplibs_check_method" : "file_magic
\(.*\)"`
            MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
            if eval $file_magic_cmd \$file_magic_test_file 2> /dev/null |
              $EGREP "$file_magic_regex" > /dev/null; then
              :
            else
              cat <<_LT_EOF 1>&2

*** Warning: the command libtool uses to detect shared libraries,
*** $file_magic_cmd, produces output that libtool cannot recognize.
*** The result is that libtool may fail to recognize shared libraries
*** as such. This will affect the creation of libtool libraries that
*** depend on shared libraries, but programs linked with such libtool

```

*** libraries will work regardless of this problem. Nevertheless, you
*** may want to report the problem to your system manager and/or to
*** bug-libtool@gnu.org

```
_LT_EOF
    fi ;;
  esac
  fi
  break
fi
done
IFS="$lt_save_ifs"
MAGIC_CMD="$lt_save_MAGIC_CMD"
;;
esac
fi

MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
if test -n "$MAGIC_CMD"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $MAGIC_CMD" >&5
$as_echo "$MAGIC_CMD" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

if test -z "$lt_cv_path_MAGIC_CMD"; then
  if test -n "$ac_tool_prefix"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for file" >&5
$as_echo_n "checking for file... " >&6; }
    if ${lt_cv_path_MAGIC_CMD+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      case $MAGIC_CMD in
[\ \/*] | ?:[\ \/*]*)
        lt_cv_path_MAGIC_CMD="$MAGIC_CMD" # Let the user override the test
with a path.
        ;;
*)
        lt_save_MAGIC_CMD="$MAGIC_CMD"
        lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
        ac_dummy="/usr/bin$PATH_SEPARATOR$PATH"
        for ac_dir in $ac_dummy; do
          IFS="$lt_save_ifs"
          test -z "$ac_dir" && ac_dir=.
          if test -f $ac_dir/file; then
            lt_cv_path_MAGIC_CMD="$ac_dir/file"
            if test -n "$file_magic_test_file"; then
```

```

    case $deplibs_check_method in
      "file_magic "*
        file_magic_regex=`expr "$deplibs_check_method" : "file_magic
\(.*\)"`
        MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
        if eval $file_magic_cmd \$file_magic_test_file 2> /dev/null |
          $EGREP "$file_magic_regex" > /dev/null; then
          :
        else
          cat <<_LT_EOF 1>&2

*** Warning: the command libtool uses to detect shared libraries,
*** $file_magic_cmd, produces output that libtool cannot recognize.
*** The result is that libtool may fail to recognize shared libraries
*** as such. This will affect the creation of libtool libraries that
*** depend on shared libraries, but programs linked with such libtool
*** libraries will work regardless of this problem. Nevertheless, you
*** may want to report the problem to your system manager and/or to
*** bug-libtool@gnu.org

_LT_EOF
        fi ;;
      esac
    fi
    break
  fi
done
IFS="$lt_save_ifs"
MAGIC_CMD="$lt_save_MAGIC_CMD"
;;
esac
fi

MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
if test -n "$MAGIC_CMD"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $MAGIC_CMD" >&5
$as_echo "$MAGIC_CMD" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

else
  MAGIC_CMD=:
fi
fi

fi
;;
esac

```

```

# Use C for the default configuration in the libtool script

lt_save_CC="$CC"
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

# Source file extension for C test sources.
ac_ext=c

# Object file extension for compiled C test sources.
objext=o
objext=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code="int some_variable = 0;"

# Code to be used in simple link tests
lt_simple_link_test_code='int main(){return(0);}'

# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC

# Save the default compiler, since it gets overwritten when the other
# tags are being tested, and _LT_TAGVAR(compiler, []) is a NOP.
compiler_DEFAULT=$CC

# save warnings/boilerplate of simple test code
ac_outfile=conftest.$ac_objext
echo "$lt_simple_compile_test_code" >conftest.$ac_ext
eval "$ac_compile" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d'
>conftest.err
_lt_compiler_boilerplate=`cat conftest.err`
$RM conftest*

ac_outfile=conftest.$ac_objext

```

```

echo "$lt_simple_link_test_code" >confptest.$ac_ext
eval "$ac_link" 2>&1 >/dev/null | $SED '/^$/d; / ^ *+ /d' >confptest.err
_lt_linker_boilerplate=`cat confptest.err`
$RM -r confptest*

```

```

if test -n "$compiler"; then

```

```

lt_prog_compiler_no_built_in_flag=

```

```

if test "$GCC" = yes; then
  case $cc_basename in
    nvcc*)
      lt_prog_compiler_no_built_in_flag=' -Xcompiler -fno-builtin' ;;
    *)
      lt_prog_compiler_no_built_in_flag=' -fno-builtin' ;;
  esac

```

```

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -fno-rtti -fno-exceptions" >&5
$as_echo_n "checking if $compiler supports -fno-rtti -fno-
exceptions... " >&6; }
if ${lt_cv_prog_compiler_rtti_exceptions+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_rtti_exceptions=no
  ac_outfile=confptest.$ac_objext
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext
  lt_compiler_flag="-fno-rtti -fno-exceptions"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\} :&$lt_compiler_flag :; t' \
-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "`echo \"$as_me:$LINENO: $lt_compile\"`" >&5)
  (eval "$lt_compile" 2>confptest.err)
  ac_status=$?
  cat confptest.err >&5
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  if (exit $ac_status) && test -s "$ac_outfile"; then
    # The compiler can only warn and ignore the option if not
recognized
    # So say no if there are warnings other than the usual output.
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >confptest.exp
    $SED '/^$/d; / ^ *+ /d' confptest.err >confptest.er2

```



```

        if test ! -s conftest.er2 || diff conftest.exp conftest.er2
>/dev/null; then
            lt_cv_prog_compiler_rtti_exceptions=yes
        fi
    fi
    $RM conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:"
$lt_cv_prog_compiler_rtti_exceptions" >&5
$as_echo "$lt_cv_prog_compiler_rtti_exceptions" >&6; }

if test x"$lt_cv_prog_compiler_rtti_exceptions" = xyes; then

lt_prog_compiler_no_builtin_flag="$lt_prog_compiler_no_builtin_flag -
fno-rtti -fno-exceptions"
else
    :
fi

fi

```

```

    lt_prog_compiler_wl=
    lt_prog_compiler_pic=
    lt_prog_compiler_static=

```

```

if test "$GCC" = yes; then
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_static='-static'

```

```

case $host_os in
    aix*)
        # All AIX code is PIC.
        if test "$host_cpu" = ia64; then
            # AIX 5 now supports IA64 processor
            lt_prog_compiler_static='-Bstatic'
        fi
        ;;

```

```

amigaos*)
    case $host_cpu in
        powerpc)
            # see comment about AmigaOS4 .so support
            lt_prog_compiler_pic='-fPIC'
            ;;
        m68k)

```

```

        # FIXME: we need at least 68020 code to build shared
libraries, but
        # adding the '-m68020' flag to GCC prevents building
anything better,
        # like '-m68040'.
        lt_prog_compiler_pic='-m68020 -resident32 -malways-
restore-a4'
        ;;
    esac
    ;;

beos* | irix5* | irix6* | nonstopux* | osf3* | osf4* | osf5*)
    # PIC is the default for these OSes.
    ;;

mingw* | cygwin* | pw32* | os2* | cegcc*)
    # This hack is so that the source file can tell whether it is
being
    # built for inclusion in a dll (and should export symbols for
example).
    # Although the cygwin gcc ignores -fPIC, still need this for
old-style
    # (--disable-auto-import) libraries
    lt_prog_compiler_pic='-DDLL_EXPORT'
    ;;

darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    lt_prog_compiler_pic='-fno-common'
    ;;

haiku*)
    # PIC is the default for Haiku.
    # The "-static" flag exists, but is broken.
    lt_prog_compiler_static=
    ;;

hpux*)
    # PIC is the default for 64-bit PA HP-UX, but not for 32-bit
    # PA HP-UX. On IA64 HP-UX, PIC is the default but the pic flag
    # sets the default TLS model and affects inlining.
    case $host_cpu in
        hppa*64*)
            # +Z the default
            ;;
        *)
            lt_prog_compiler_pic='-fPIC'
            ;;
    esac
    ;;

```

```

interix[3-9]*)
    # Interix 3.x gcc -fpic/-fPIC options generate broken code.
    # Instead, we relocate shared libraries at runtime.
    ;;

msdosdjgpp*)
    # Just because we use GCC doesn't mean we suddenly get shared
libraries
    # on systems that don't support them.
    lt_prog_compiler_can_build_shared=no
    enable_shared=no
    ;;

*nto* | *qnx*)
    # QNX uses GNU C++, but need to define -shared option too,
otherwise
    # it will coredump.
    lt_prog_compiler_pic='-fPIC -shared'
    ;;

sysv4*MP*)
    if test -d /usr/nec; then
        lt_prog_compiler_pic=-Kconform_pic
    fi
    ;;

*)
    lt_prog_compiler_pic='-fPIC'
    ;;
esac

case $cc_basename in
nvcc*) # Cuda Compiler Driver 2.2
    lt_prog_compiler_wl='-Xlinker '
    if test -n "$lt_prog_compiler_pic"; then
        lt_prog_compiler_pic="-Xcompiler $lt_prog_compiler_pic"
    fi
    ;;
esac
else
    # PORTME Check for flag to pass linker flags through the system
compiler.
    case $host_os in
aix*)
        lt_prog_compiler_wl='-Wl,'
        if test "$host_cpu" = ia64; then
            # AIX 5 now supports IA64 processor
            lt_prog_compiler_static='-Bstatic'
        else
            lt_prog_compiler_static='-bnso -bI:/lib/syscalls.exp'
        fi
        ;;

```

```

mingw* | cygwin* | pw32* | os2* | cegcc*)
    # This hack is so that the source file can tell whether it is
being
    # built for inclusion in a dll (and should export symbols for
example).
    lt_prog_compiler_pic='-DDLL_EXPORT'
    ;;

hpux9* | hpux10* | hpux11*)
    lt_prog_compiler_wl='-Wl,'
    # PIC is the default for IA64 HP-UX and 64-bit HP-UX, but
    # not for PA HP-UX.
    case $host_cpu in
    hppa*64*|ia64*)
    # +Z the default
    ;;
    *)
    lt_prog_compiler_pic='+Z'
    ;;
    esac
    # Is there a better lt_prog_compiler_static that works with the
bundled CC?
    lt_prog_compiler_static='${wl}-a ${wl}archive'
    ;;

irix5* | irix6* | nonstopux*)
    lt_prog_compiler_wl='-Wl,'
    # PIC (with -KPIC) is the default.
    lt_prog_compiler_static='-non_shared'
    ;;

linux* | k*bsd*-gnu | kopensolaris*-gnu)
    case $cc_basename in
    # old Intel for x86_64 which still supported -KPIC.
    ecc*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-static'
    ;;
    # icc used to be incompatible with GCC.
    # ICC 10 doesn't accept -KPIC any more.
    icc* | ifort*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-fPIC'
    lt_prog_compiler_static='-static'
    ;;
    # Lahey Fortran 8.1.
    lf95*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='--shared'
    lt_prog_compiler_static='--static'

```

```

;;
nagfor*)
# NAG Fortran compiler
lt_prog_compiler_wl='-Wl,-Wl,,'
lt_prog_compiler_pic='-PIC'
lt_prog_compiler_static='-Bstatic'
;;
pgcc* | pgf77* | pgf90* | pgf95* | pgfortran*)
# Portland Group compilers (*not* the Pentium gcc compiler,
# which looks to be a dead project)
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-fpic'
lt_prog_compiler_static='-Bstatic'
;;
ccc*)
lt_prog_compiler_wl='-Wl,'
# All Alpha code is PIC.
lt_prog_compiler_static='-non_shared'
;;
xl* | bgxl* | bgf* | mpixl*)
# IBM XL C 8.0/Fortran 10.1, 11.1 on PPC and BlueGene
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-qpik'
lt_prog_compiler_static='-qstaticlink'
;;
*)
case `\$CC -V 2>&1 | sed 5q` in
*Sun\ Ceres\ Fortran* | *Sun*Fortran*\ [1-7].* | *Sun*Fortran*\
8.[0-3]*)
# Sun Fortran 8.3 passes all unrecognized flags to the linker
lt_prog_compiler_pic='-KPIC'
lt_prog_compiler_static='-Bstatic'
lt_prog_compiler_wl=''
;;
*Sun\ F* | *Sun*Fortran*)
lt_prog_compiler_pic='-KPIC'
lt_prog_compiler_static='-Bstatic'
lt_prog_compiler_wl='-Qoption ld '
;;
*Sun\ C*)
# Sun C 5.9
lt_prog_compiler_pic='-KPIC'
lt_prog_compiler_static='-Bstatic'
lt_prog_compiler_wl='-Wl,'
;;
*Intel*\ [CF]*Compiler*)
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-fPIC'
lt_prog_compiler_static='-static'
;;
*Portland\ Group*)
lt_prog_compiler_wl='-Wl,'

```

```

        lt_prog_compiler_pic='-fpic'
        lt_prog_compiler_static='-Bstatic'
        ;;
    esac
    ;;
    esac
    ;;

newsos6)
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

*nto* | *qnx*)
    # QNX uses GNU C++, but need to define -shared option too,
otherwise
    # it will coredump.
    lt_prog_compiler_pic='-fPIC -shared'
    ;;

osf3* | osf4* | osf5*)
    lt_prog_compiler_wl='-Wl,'
    # All OSF/1 code is PIC.
    lt_prog_compiler_static='-non_shared'
    ;;

rdos*)
    lt_prog_compiler_static='-non_shared'
    ;;

solaris*)
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    case $cc_basename in
    f77* | f90* | f95* | sunf77* | sunf90* | sunf95*)
    lt_prog_compiler_wl='-Qoption ld ';;
    *)
    lt_prog_compiler_wl='-Wl, ';;
    esac
    ;;

sunos4*)
    lt_prog_compiler_wl='-Qoption ld '
    lt_prog_compiler_pic='-PIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

sysv4 | sysv4.2uw2* | sysv4.3*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

```

```

sysv4*MP*)
    if test -d /usr/nec ;then
    lt_prog_compiler_pic='-Kconform_pic'
    lt_prog_compiler_static='-Bstatic'
    fi
    ;;

sysv5* | unixware* | sco3.2v5* | sco5v6* | OpenUNIX*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

unicos*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_can_build_shared=no
    ;;

uts4*)
    lt_prog_compiler_pic='-pic'
    lt_prog_compiler_static='-Bstatic'
    ;;

*)
    lt_prog_compiler_can_build_shared=no
    ;;
esac
fi

case $host_os in
# For platforms which do not support PIC, -DPIC is meaningless:
*djgpp*)
    lt_prog_compiler_pic=
    ;;
*)
    lt_prog_compiler_pic="$lt_prog_compiler_pic -DPIC"
    ;;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $compiler option
to produce PIC" >&5
$as_echo_n "checking for $compiler option to produce PIC... " >&6; }
if ${lt_cv_prog_compiler_pic+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_prog_compiler_pic=$lt_prog_compiler_pic
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic" >&5
$as_echo "$lt_cv_prog_compiler_pic" >&6; }
lt_prog_compiler_pic=$lt_cv_prog_compiler_pic

```

```

#
# Check to make sure the PIC flag actually works.
#
if test -n "$lt_prog_compiler_pic"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler PIC
flag $lt_prog_compiler_pic works" >&5
$as_echo_n "checking if $compiler PIC flag $lt_prog_compiler_pic
works..." >&6; }
if ${lt_cv_prog_compiler_pic_works+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_pic_works=no
  ac_outfile=confptest.$ac_objext
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext
  lt_compiler_flag="$lt_prog_compiler_pic -DPIC"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "`"$as_me:$LINENO: $lt_compile`" >&5)
  (eval "$lt_compile" 2>confptest.err)
  ac_status=$?
  cat confptest.err >&5
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  if (exit $ac_status) && test -s "$ac_outfile"; then
    # The compiler can only warn and ignore the option if not
recognized
    # So say no if there are warnings other than the usual output.
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >confptest.exp
    $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2
    if test ! -s confptest.er2 || diff confptest.exp confptest.er2
>/dev/null; then
      lt_cv_prog_compiler_pic_works=yes
    fi
  fi
  $RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic_works" >&5
$as_echo "$lt_cv_prog_compiler_pic_works" >&6; }

if test x"$lt_cv_prog_compiler_pic_works" = xyes; then
  case $lt_prog_compiler_pic in

```



```

    "" | " ") ;;
    *) lt_prog_compiler_pic=" $lt_prog_compiler_pic" ;;
  esac
else
  lt_prog_compiler_pic=
  lt_prog_compiler_can_build_shared=no
fi

fi

#
# Check to make sure the static flag actually works.
#
wl=$lt_prog_compiler_wl eval
lt_tmp_static_flag="\$lt_prog_compiler_static\"
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler static
flag $lt_tmp_static_flag works" >&5
$as_echo_n "checking if $compiler static flag $lt_tmp_static_flag
works... " >&6; }
if ${lt_cv_prog_compiler_static_works+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_static_works=no
  save_LDFLAGS="$LDFLAGS"
  LDFLAGS="$LDFLAGS $lt_tmp_static_flag"
  echo "$lt_simple_link_test_code" > conftest.$ac_ext
  if (eval $ac_link 2>conftest.err) && test -s conftest.$ac_exeext;
then
  # The linker can only warn and ignore the option if not
  recognized
  # So say no if there are warnings
  if test -s conftest.err; then
    # Append any errors to the config.log.
    cat conftest.err 1>&5
    $ECHO "$_lt_linker_boilerplate" | $SED '/^$/d' > conftest.exp
    $SED '/^$/d; /^ *+/d' conftest.err >conftest.er2
    if diff conftest.exp conftest.er2 >/dev/null; then
      lt_cv_prog_compiler_static_works=yes
    fi
  else
    lt_cv_prog_compiler_static_works=yes
  fi
fi

```

```

fi
$RM -r conftest*
LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_static_works" >&5
$as_echo "$lt_cv_prog_compiler_static_works" >&6; }

if test x"$lt_cv_prog_compiler_static_works" = xyes; then
:
else
lt_prog_compiler_static=
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o+:} false; then :
$as_echo_n "(cached) " >&6
else
lt_cv_prog_compiler_c_o=no
$RM -r conftest 2>/dev/null
mkdir conftest
cd conftest
mkdir out
echo "$lt_simple_compile_test_code" > conftest.$ac_ext

lt_compiler_flag="-o out/conftest2.$ac_objext"
# Insert the option either (1) after the last *FLAGS variable, or
# (2) before a word containing "conftest.", or (3) at the end.
# Note that $ac_compile itself does not contain backslashes and
begins
# with a dollar sign (not a hyphen), so the echo should work
correctly.
lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\} :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*conftest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
(eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&5)
(eval "$lt_compile" 2>out/conftest.err)
ac_status=$?
cat out/conftest.err >&5
echo "$as_me:$LINENO: \$? = $ac_status" >&5
if (exit $ac_status) && test -s out/conftest2.$ac_objext

```

```

then
    # The compiler can only warn and ignore the option if not
    recognized
    # So say no if there are warnings
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/confptest.exp
    $SED '/^$/d; /^ *+/d' out/confptest.err >out/confptest.er2
    if test ! -s out/confptest.er2 || diff out/confptest.exp
out/confptest.er2 >/dev/null; then
        lt_cv_prog_compiler_c_o=yes
    fi
fi
chmod u+w . 2>&5
$RM confptest*
# SGI C++ compiler will create directory out/ii_files/ for
# template instantiation
test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
$RM out/* && rmdir out
cd ..
$RM -r confptest
$RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o" >&5
$as_echo "$lt_cv_prog_compiler_c_o" >&6; }

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_prog_compiler_c_o=no
    $RM -r confptest 2>/dev/null
    mkdir confptest
    cd confptest
    mkdir out
    echo "$lt_simple_compile_test_code" > confptest.$ac_ext

    lt_compiler_flag="-o out/confptest2.$ac_objext"
    # Insert the option either (1) after the last *FLAGS variable, or
    # (2) before a word containing "confptest.", or (3) at the end.
    # Note that $ac_compile itself does not contain backslashes and
    begins

```

```

# with a dollar sign (not a hyphen), so the echo should work
correctly.
lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS)\{0,1\} :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*conftest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
(eval echo "\\"$as_me:$LINENO: $lt_compile\"" >&5)
(eval "$lt_compile" 2>out/conftest.err)
ac_status=$?
cat out/conftest.err >&5
echo "$as_me:$LINENO: \ $? = $ac_status" >&5
if (exit $ac_status) && test -s out/conftest2.$ac_objext
then
# The compiler can only warn and ignore the option if not
recognized
# So say no if there are warnings
$ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/conftest.exp
$SED '/^$/d; /^ *+/d' out/conftest.err >out/conftest.er2
if test ! -s out/conftest.er2 || diff out/conftest.exp
out/conftest.er2 >/dev/null; then
lt_cv_prog_compiler_c_o=yes
fi
fi
chmod u+w . 2>&5
$RM conftest*
# SGI C++ compiler will create directory out/ii_files/ for
# template instantiation
test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
$RM out/* && rmdir out
cd ..
$RM -r conftest
$RM conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o" >&5
$as_echo "$lt_cv_prog_compiler_c_o" >&6; }

hard_links="nottested"
if test "$lt_cv_prog_compiler_c_o" = no && test "$need_locks" != no;
then
# do not overwrite the value of need_locks provided by the user
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if we can lock
with hard links" >&5
$as_echo_n "checking if we can lock with hard links... " >&6; }
hard_links=yes
$RM conftest*
ln conftest.a conftest.b 2>/dev/null && hard_links=no

```

```

touch confptest.a
ln confptest.a confptest.b 2>&5 || hard_links=no
ln confptest.a confptest.b 2>/dev/null && hard_links=no
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $hard_links" >&5
$sas_echo "$hard_links" >&6; }
if test "$hard_links" = no; then
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: WARNING: \`$CC' does not
support \`-c -o', so \`make -j' may be unsafe" >&5
$sas_echo "$sas_me: WARNING: \`$CC' does not support \`-c -o', so \`make
-j' may be unsafe" >&2;}
    need_locks=warn
fi
else
    need_locks=no
fi

```

```

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking whether the
$compiler linker ($LD) supports shared libraries" >&5
$sas_echo_n "checking whether the $compiler linker ($LD) supports
shared libraries... " >&6; }

```

```

runpath_var=
allow_undefined_flag=
always_export_symbols=no
archive_cmds=
archive_expsym_cmds=
compiler_needs_object=no
enable_shared_with_static_runtimes=no
export_dynamic_flag_spec=
export_symbols_cmds='$NM $libobjs $convenience | $global_symbol_pipe
| $SED '\''s/.* //'\'' | sort | uniq > $export_symbols'
hardcode_automatic=no
hardcode_direct=no
hardcode_direct_absolute=no
hardcode_libdir_flag_spec=
hardcode_libdir_separator=
hardcode_minus_L=no
hardcode_shlibpath_var=unsupported
inherit_rpath=no
link_all_deplibs=unknown
module_cmds=
module_expsym_cmds=
old_archive_from_new_cmds=
old_archive_from_expsyms_cmds=
thread_safe_flag_spec=
whole_archive_flag_spec=

```

```

# include_expsyms should be a list of space-separated symbols to be
*always*
# included in the symbol list
include_expsyms=
# exclude_expsyms can be an extended regexp of symbols to exclude
# it will be wrapped by ` (' and `)'$', so one must not match
beginning or
# end of line. Example: `a|bc|.*d.*' will exclude the symbols `a'
and `bc',
# as well as any symbol that contains `d'.
exclude_expsyms='_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*'
# Although _GLOBAL_OFFSET_TABLE_ is a valid symbol C name, most
a.out
# platforms (ab)use it in PIC code, but their linkers get confused
if
# the symbol is explicitly referenced. Since portable code cannot
# rely on this symbol name, it's probably fine to never include it
in
# preloaded symbol tables.
# Exclude shared library initialization/finalization symbols.
extract_expsyms_cmds=

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
# FIXME: the MSVC++ port hasn't been tested in a loooong time
# When not using gcc, we currently assume that we are using
# Microsoft Visual C++.
if test "$GCC" != yes; then
with_gnu_ld=no
fi
;;
interix*)
# we just hope/assume this is gcc and not c89 (= MSVC++)
with_gnu_ld=yes
;;
openbsd*)
with_gnu_ld=no
;;
esac

ld_shlibs=yes

# On some targets, GNU ld is compatible enough with the native
linker
# that we're better off using the native interface for both.
lt_use_gnu_ld_interface=no
if test "$with_gnu_ld" = yes; then
case $host_os in
aix*)
# The AIX port of GNU ld has always aspired to compatibility
# with the native linker. However, as the warning in the GNU ld

```

```

    # block says, versions before 2.19.5* couldn't really create
working
    # shared libraries, regardless of the interface used.
    case ` $LD -v 2>&1 ` in
        *\ \ (GNU\ Binutils\)\ 2.19.5* ) ;;
        *\ \ (GNU\ Binutils\)\ 2.[2-9]* ) ;;
        *\ \ (GNU\ Binutils\)\ [3-9]* ) ;;
        *)
            lt_use_gnu_ld_interface=yes
            ;;
    esac
    ;;
    *)
        lt_use_gnu_ld_interface=yes
        ;;
    esac
fi

if test "$lt_use_gnu_ld_interface" = yes; then
    # If archive_cmds runs LD, not CC, wlarc should be empty
    wlarc='${wl}'

    # Set some defaults for GNU ld with shared library support. These
    # are reset later if shared libraries are not supported. Putting
them
    # here allows them to be overridden if necessary.
    runpath_var=LD_RUN_PATH
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    export_dynamic_flag_spec='${wl}--export-dynamic'
    # ancient GNU ld didn't support --whole-archive et. al.
    if $LD --help 2>&1 | $GREP 'no-whole-archive' > /dev/null; then
        whole_archive_flag_spec="$wlarc"--whole-archive$convenience
        '$wlarc"--no-whole-archive'
    else
        whole_archive_flag_spec=
    fi
    supports_anon_versioning=no
    case ` $LD -v 2>&1 ` in
        *GNU\ gold*) supports_anon_versioning=yes ;;
        *\ [01].* | *\ 2.[0-9].* | *\ 2.10.* ) ;; # catch versions < 2.11
        *\ 2.11.93.0.2\ *) supports_anon_versioning=yes ;; # RH7.3 ...
        *\ 2.11.92.0.12\ *) supports_anon_versioning=yes ;; # Mandrake
8.2 ...
        *\ 2.11.* ) ;; # other 2.11 versions
        *) supports_anon_versioning=yes ;;
    esac

    # See if GNU ld supports shared libraries.
    case $host_os in
    aix[3-9]*)
        # On AIX/PPC, the GNU linker is very broken
        if test "$host_cpu" != ia64; then

```

```

ld_shlibs=no
cat <<_LT_EOF 1>&2

*** Warning: the GNU linker, at least up to release 2.19, is reported
*** to be unable to reliably create shared libraries on AIX.
*** Therefore, libtool is disabling shared libraries support.  If you
*** really care for shared libraries, you may want to install binutils
*** 2.20 or above, or modify your PATH so that a non-GNU linker is
found.
*** You will then need to restart the configuration process.

_LT_EOF
fi
;;

amigaos*)
case $host_cpu in
powerpc)
# see comment about AmigaOS4 .so support
archive_cmds='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
archive_expsym_cmds=''
;;
m68k)
archive_cmds='$RM $output_objdir/a2ixlibrary.data~$ECHO
"#define NAME $libname" > $output_objdir/a2ixlibrary.data~$ECHO
"#define LIBRARY_ID 1" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR
$AR_FLAGS $lib $libobjs~$RANLIB $lib~(cd $output_objdir && a2ixlibrary
-32)'
hardcode_libdir_flag_spec='-L$libdir'
hardcode_minus_L=yes
;;
esac
;;

beos*)
if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
allow_undefined_flag=unsupported
# Joseph Beckenbach <jrb3@best.com> says some releases of gcc
# support --undefined. This deserves some investigation.  FIXME
archive_cmds='$CC -nostart $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
else
ld_shlibs=no
fi
;;

cygwin* | mingw* | pw32* | cegcc*)

```



```

# _LT_TAGVAR(hardcode_libdir_flag_spec, ) is actually
meaningless,
# as there is no search path for DLLs.
hardcode_libdir_flag_spec='-L$libdir'
export_dynamic_flag_spec='${wl}--export-all-symbols'
allow_undefined_flag=unsupported
always_export_symbols=no
enable_shared_with_static_runtimes=yes
export_symbols_cmds='$NM $libobjs $convenience |
$global_symbol_pipe | $SED -e '\''/^([BCDGRS][ ]/s/.*[ ]\([^\ ]*\)/\1
DATA;/s/^\.*[ ]__nm__\([^\ ]*\)[ ]^\ ]*/\1 DATA;/^I[ ]/d;/^[AITW][
]/s/.* //'\' | sort | uniq > $export_symbols'

exclude_expsyms='[_]+GLOBAL_OFFSET_TABLE_|[_]+GLOBAL__[FID]_.*|[_]+hea
d_[A-Za-z0-9]+'_dll'|[_]+_dll_iname'

if $LD --help 2>&1 | $GREP 'auto-import' > /dev/null; then
  archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags -o
$output_objdir/$soname ${wl}--enable-auto-image-base -Xlinker --out-
implib -Xlinker $lib'
  # If the export-symbols file already is a .def file (1st line
  # is EXPORTS), use it as is; otherwise, prepend...
  archive_expsym_cmds='if test "x`$SED 1q $export_symbols`" =
xEXPORTS; then
  cp $export_symbols $output_objdir/$soname.def;
else
  echo EXPORTS > $output_objdir/$soname.def;
  cat $export_symbols >> $output_objdir/$soname.def;
fi~
$CC -shared $output_objdir/$soname.def $libobjs $deplibs
$compiler_flags -o $output_objdir/$soname ${wl}--enable-auto-image-
base -Xlinker --out-implib -Xlinker $lib'
else
  ld_shlibs=no
fi
;;

haiku*)
  archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
  link_all_deplibs=yes
  ;;

interix[3-9]*)
  hardcode_direct=no
  hardcode_shlibpath_var=no
  hardcode_libdir_flag_spec='${wl}-rpath,$libdir'
  export_dynamic_flag_spec='${wl}-E'
  # Hack: On Interix 3.x, we cannot compile PIC because of a
  broken gcc.
  # Instead, shared libraries are loaded at an image base
  (0x10000000 by

```

```

# default) and relocated if they conflict, which is a slow very
memory
# consuming and fragmenting process. To avoid this, we pick a
random,
# 256 KiB-aligned image base between 0x50000000 and 0x6FFC0000
at link
# time. Moving up from 0x10000000 also allows more sbrk(2)
space.
archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--image-base,\`expr ${RANDOM-$$} %
4096 / 2 \ $\backslash$ * 262144 + 1342177280\ $\backslash$  -o $lib'
archive_expsym_cmds='sed "s,^,_,\" $export_symbols
>$output_objdir/$soname.expsym~$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--retain-symbols-
file,$output_objdir/$soname.expsym ${wl}--image-base,\`expr ${RANDOM-
$$} % 4096 / 2 \ $\backslash$ * 262144 + 1342177280\ $\backslash$  -o $lib'
;;

gnu* | linux* | tpf* | k*bsd*-gnu | kopensolaris*-gnu)
tmp_diet=no
if test "$host_os" = linux-dietlibc; then
case $cc_basename in
diet\ *) tmp_diet=yes;; # linux-dietlibc with static linking
(!diet-dyn)
esac
fi
if $LD --help 2>&1 | $EGREP ': supported targets:.* elf' >
/dev/null \
&& test "$tmp_diet" = no
then
tmp_addflag=' $pic_flag'
tmp_sharedflag='-shared'
case $cc_basename,$host_cpu in
pgcc*) # Portland Group C compiler
whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`\"; do test -n \"$conv\" &&
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\`\" ` ${wl}--no-whole-archive'
tmp_addflag=' $pic_flag'
;;
pgf77* | pgf90* | pgf95* | pgfortran*)
# Portland Group f77 and f90 compilers
whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`\"; do test -n \"$conv\" &&
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\`\" ` ${wl}--no-whole-archive'
tmp_addflag=' $pic_flag -Mnomain' ;;
ecc*,ia64* | icc*,ia64*) # Intel C compiler on ia64
tmp_addflag=' -i_dynamic' ;;
efc*,ia64* | ifort*,ia64*) # Intel Fortran compiler on ia64
tmp_addflag=' -i_dynamic -nofor_main' ;;
ifc* | ifort*) # Intel Fortran compiler

```

```

    tmp_addflag=' -nofor_main' ;;
lf95*)
    # Lahey Fortran 8.1
    whole_archive_flag_spec=
    tmp_sharedflag='--shared' ;;
xl[cC]* | bgxl[cC]* | mpixl[cC]*) # IBM XL C 8.0 on PPC (deal
with xlf below)
    tmp_sharedflag='-qmkshrobj'
    tmp_addflag= ;;
nvcc*)
    # Cuda Compiler Driver 2.2
    whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`\`; do test -n \`${conv}\` &&
new_convenience=\`${new_convenience,$conv}\`; done; func_echo_all
\`${new_convenience}\` \`${wl}--no-whole-archive'
    compiler_needs_object=yes
    ;;
esac
case ` $CC -V 2>&1 | sed 5q` in
*Sun\ C*)
    # Sun C 5.9
    whole_archive_flag_spec='${wl}--whole-archive`new_convenience=;
for conv in $convenience\`\`; do test -z \`${conv}\` ||
new_convenience=\`${new_convenience,$conv}\`; done; func_echo_all
\`${new_convenience}\` \`${wl}--no-whole-archive'
    compiler_needs_object=yes
    tmp_sharedflag='-G' ;;
*Sun\ F*)
    # Sun Fortran 8.3
    tmp_sharedflag='-G' ;;
esac
archive_cmds='$CC "'$tmp_sharedflag"'$tmp_addflag"' $libobjs
$deplibs $compiler_flags ${wl}-soname $wl$soname -o $lib'

    if test "x$supports_anon_versioning" = xyes; then
        archive_expsym_cmds='echo "{ global:" >
$output_objdir/$libname.ver~
cat $export_symbols | sed -e "s/(.*)/\1;/ " >>
$output_objdir/$libname.ver~
echo "local: *; };" >> $output_objdir/$libname.ver~
$CC "'$tmp_sharedflag"'$tmp_addflag"' $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-version-script
${wl}$output_objdir/$libname.ver -o $lib'
    fi

case $cc_basename in
xlf* | bgf* | bgxlf* | mpixlf*)
    # IBM XL Fortran 10.1 on PPC cannot create shared libs itself
    whole_archive_flag_spec='--whole-archive$convenience --no-
whole-archive'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    archive_cmds='$LD -shared $libobjs $deplibs $linker_flags -
soname $soname -o $lib'
    if test "x$supports_anon_versioning" = xyes; then
        archive_expsym_cmds='echo "{ global:" >
$output_objdir/$libname.ver~

```

```

        cat $export_symbols | sed -e "s/\(.*\)\/\1;/" >>
$output_objdir/$libname.ver~
        echo "local: *; };" >> $output_objdir/$libname.ver~
        $LD -shared $libobjs $deplibs $linker_flags -soname $soname
-version-script $output_objdir/$libname.ver -o $lib'
    fi
    ;;
esac
else
    ld_shlibs=no
fi
;;

netbsd*)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
        archive_cmds='$LD -Bshareable $libobjs $deplibs $linker_flags -o
$lib'
        wlarc=
    else
        archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
    fi
    ;;

solaris*)
    if $LD -v 2>&1 | $GREP 'BFD 2\.8' > /dev/null; then
        ld_shlibs=no
        cat <<_LT_EOF 1>&2

*** Warning: The releases 2.8.* of the GNU linker cannot reliably
*** create shared libraries on Solaris systems.  Therefore, libtool
*** is disabling shared libraries support.  We urge you to upgrade GNU
*** binutils to release 2.9.1 or newer.  Another option is to modify
*** your PATH or compiler configuration so that the native linker is
*** used, and then restart.

_LT_EOF
        elif $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
            archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
            archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
        else
            ld_shlibs=no
        fi
    ;;

```

```

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX*)
  case ` $LD -v 2>&1 ` in
    *\ [01].* | *\ 2.[0-9].* | *\ 2.1[0-5].*)
      ld_shlibs=no
      cat <<_LT_EOF 1>&2

```

*** Warning: Releases of the GNU linker prior to 2.16.91.0.3 can not
 *** reliably create shared libraries on SCO systems. Therefore,
 libtool
 *** is disabling shared libraries support. We urge you to upgrade GNU
 *** binutils to release 2.16.91.0.3 or newer. Another option is to
 modify
 *** your PATH or compiler configuration so that the native linker is
 *** used, and then restart.

```

_LT_EOF
  ;;
  *)
    # For security reasons, it is highly recommended that you
    always
    # use absolute paths for naming shared libraries, and exclude
    the
    # DT_RUNPATH tag from executables and libraries. But doing so
    # requires that you compile everything twice, which is a pain.
    if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
    /dev/null; then
      hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
      archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags
      ${wl}-soname $wl$soname -o $lib'
      archive_expsym_cmds='$CC -shared $libobjs $deplibs
      $compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
      $wl$export_symbols -o $lib'
    else
      ld_shlibs=no
    fi
  ;;
esac
  ;;

sunos4*)
  archive_cmds='$LD -assert pure-text -Bshareable -o $lib $libobjs
  $deplibs $linker_flags'
  wlarc=
  hardcode_direct=yes
  hardcode_shlibpath_var=no
  ;;

  *)
    if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
    /dev/null; then
      archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
      $compiler_flags ${wl}-soname $wl$soname -o $lib'

```

```

        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
        else
        ld_shlibs=no
        fi
        ;;
    esac

    if test "$ld_shlibs" = no; then
        runpath_var=
        hardcode_libdir_flag_spec=
        export_dynamic_flag_spec=
        whole_archive_flag_spec=
    fi
else
    # PORTME fill in a description of your system's linker (not GNU
ld)
    case $host_os in
    aix3*)
        allow_undefined_flag=unsupported
        always_export_symbols=yes
        archive_expsym_cmds='$LD -o $output_objdir/$soname $libobjs
$deplibs $linker_flags -bE:$export_symbols -T512 -H512 -bM:SRE~$AR
$AR_FLAGS $lib $output_objdir/$soname'
        # Note: this linker hardcodes the directories in LIBPATH if
there
        # are no directories specified by -L.
        hardcode_minus_L=yes
        if test "$GCC" = yes && test -z "$lt_prog_compiler_static"; then
            # Neither direct hardcoding nor static linking is supported with
a
            # broken collect2.
            hardcode_direct=unsupported
            fi
        ;;
    aix[4-9]*)
        if test "$host_cpu" = ia64; then
            # On IA64, the linker does run time linking by default, so we
don't
            # have to do anything special.
            aix_use_runtimelinking=no
            exp_sym_flag='-Bexport'
            no_entry_flag=""
        else
            # If we're using GNU nm, then we don't want the "-C" option.
            # -C means demangle to AIX nm, but means don't demangle with GNU
nm
            # Also, AIX nm treats weak defined symbols like other global
            # defined symbols, whereas GNU nm marks them as "W".
            if $NM -V 2>&1 | $GREP 'GNU' > /dev/null; then

```

```

        export_symbols_cmds='$NM -Bpg $libobjs $convenience | awk '\''{
if ((\($ 2 == "T") || (\$ 2 == "D") || (\$ 2 == "B") || (\$ 2 == "W"))
&& (substr(\$ 3,1,1) != ".")) { print \$ 3 } }'\'' | sort -u >
$export_symbols'
    else
        export_symbols_cmds='$NM -BCpg $libobjs $convenience | awk
'\''{ if ((\($ 2 == "T") || (\$ 2 == "D") || (\$ 2 == "B")) &&
(substr(\$ 3,1,1) != ".")) { print \$ 3 } }'\'' | sort -u >
$export_symbols'
    fi
    aix_use_runtimelinking=no

# Test if we are trying to use run time linking or normal
# AIX style linking. If -brtl is somewhere in LDFLAGS, we
# need to do runtime linking.
case $host_os in aix4.[23]|aix4.[23].*|aix[5-9]*)
    for ld_flag in $LDFLAGS; do
        if (test $ld_flag = "-brtl" || test $ld_flag = "-Wl,-brtl");
then
            aix_use_runtimelinking=yes
            break
        fi
    done
    ;;
esac

exp_sym_flag='-bexport'
no_entry_flag='-bnoentry'
fi

# When large executables or shared objects are built, AIX ld can
# have problems creating the table of contents. If linking a
library
# or program results in "error TOC overflow" add -mminimal-toc
to
# CXXFLAGS/CFLAGS for g++/gcc. In the cases where that is not
# enough to fix the problem, add -Wl,-bbigtoc to LDFLAGS.

archive_cmds=''
hardcode_direct=yes
hardcode_direct_absolute=yes
hardcode_libdir_separator=':'
link_all_deplibs=yes
file_list_spec='${wl}-f,'

if test "$GCC" = yes; then
case $host_os in aix4.[012]|aix4.[012].*)
# We only want to do this on AIX 4.2 and lower, the check
# below for broken collect2 doesn't work under 4.3+
collect2name=`${CC} -print-prog-name=collect2`
if test -f "$collect2name" &&
strings "$collect2name" | $GREP resolve_lib_name >/dev/null

```

```

then
# We have reworked collect2
:
else
# We have old collect2
hardcode_direct=unsupported
# It fails to find uninstalled libraries when the uninstalled
# path is not listed in the libpath.  Setting hardcode_minus_L
# to unsupported forces relinking
hardcode_minus_L=yes
hardcode_libdir_flag_spec='-L$libdir'
hardcode_libdir_separator=
fi
;;
esac
shared_flag='-shared'
if test "$aix_use_runtimelinking" = yes; then
  shared_flag="$shared_flag "${wl}-G'
fi
else
# not using gcc
if test "$host_cpu" = ia64; then
# VisualAge C++, Version 5.5 for AIX 5L for IA-64, Beta 3 Release
# chokes on -Wl,-G. The following line is correct:
  shared_flag='-G'
else
  if test "$aix_use_runtimelinking" = yes; then
    shared_flag='${wl}-G'
  else
    shared_flag='${wl}-bM:SRE'
  fi
fi
fi

export_dynamic_flag_spec='${wl}-bexpall'
# It seems that -bexpall does not export symbols beginning with
# underscore (_), so it is better to generate a list of symbols
to export.
always_export_symbols=yes
if test "$aix_use_runtimelinking" = yes; then
# Warning - without using the other runtime loading flags (-
brtl),
# -berok will link without error, but may produce a broken
library.
  allow_undefined_flag='-berok'
  # Determine the default libpath from the value encoded in an
  # empty executable.
  if test "${lt_cv_aix_libpath+set}" = set; then
aix_libpath=$lt_cv_aix_libpath
else
if ${lt_cv_aix_libpath_+set} false; then :
$as_echo_n "(cached) " >&6

```



```

else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :

  lt_aix_libpath_sed='
    /Import File Strings/,/^$/ {
      /^0/ {
        s/^0  *\([^ ]*\) *$/\1/
        p
      }
    }'

  lt_cv_aix_libpath_=`dump -H conftest$ac_exeext 2>/dev/null | $SED -n
-e "$lt_aix_libpath_sed"`
  # Check for a 64-bit object if we didn't find anything.
  if test -z "$lt_cv_aix_libpath_"; then
    lt_cv_aix_libpath_=`dump -HX64 conftest$ac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
  fi
fi

rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
  if test -z "$lt_cv_aix_libpath_"; then
    lt_cv_aix_libpath_="/usr/lib:/lib"
  fi
fi

aix_libpath=$lt_cv_aix_libpath_
fi

  hardcode_libdir_flag_spec='${wl}-
blibpath:$libdir:'"$aix_libpath"
  archive_expsym_cmds='$CC -o $output_objdir/$soname $libobjs
$deplibs '"\${wl}$no_entry_flag"' $compiler_flags `if test
"x${allow_undefined_flag}" != "x"; then func_echo_all
"${wl}${allow_undefined_flag}"; else ;; fi`
'"\${wl}$exp_sym_flag:\$export_symbols $shared_flag"
  else
    if test "$host_cpu" = ia64; then
      hardcode_libdir_flag_spec='${wl}-R $libdir:/usr/lib:/lib'
      allow_undefined_flag="-z nodefs"

```

```

        archive_expsym_cmds="\$CC $shared_flag" -o
$output_objdir/$soname $libobjs $deplibs '"\${wl}$no_entry_flag"
$compiler_flags \${wl}\${allow_undefined_flag}
'\${wl}$exp_sym_flag:\$export_symbols"
    else
        # Determine the default libpath from the value encoded in an
        # empty executable.
        if test "\${lt_cv_aix_libpath+set}" = set; then
            aix_libpath=$lt_cv_aix_libpath
        else
            if \${lt_cv_aix_libpath+;} false; then :
                $as_echo_n "(cached) " >&6
            else
                cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :

    lt_aix_libpath_sed='
        /Import File Strings/,/^$/ {
            /^0/ {
                s/^0 *\[^\ ]*\) */\1/
                p
            }
        }'
    lt_cv_aix_libpath_=`dump -H conftest$ac_exeext 2>/dev/null | $SED -n
-e "$lt_aix_libpath_sed"`
    # Check for a 64-bit object if we didn't find anything.
    if test -z "$lt_cv_aix_libpath_"; then
        lt_cv_aix_libpath_=`dump -HX64 conftest$ac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
    fi
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
if test -z "$lt_cv_aix_libpath_"; then
    lt_cv_aix_libpath_="/usr/lib:/lib"
fi

fi

aix_libpath=$lt_cv_aix_libpath_
fi

```

```

        hardcode_libdir_flag_spec='${wl}-
bllibpath:$libdir:""$aix_libpath"
        # Warning - without using the other run time loading flags,
        # -berok will link without error, but may produce a broken
library.
        no_undefined_flag=' ${wl}-bernotok'
        allow_undefined_flag=' ${wl}-berok'
        if test "$with_gnu_ld" = yes; then
            # We only use this code for GNU lds that support --whole-
archive.
            whole_archive_flag_spec='${wl}--whole-archive$convenience
${wl}--no-whole-archive'
        else
            # Exported symbols can be pulled into shared objects from
archives
            whole_archive_flag_spec='$convenience'
        fi
        archive_cmds_need_lc=yes
        # This is similar to how AIX traditionally builds its shared
libraries.
        archive_expsym_cmds="\$CC $shared_flag" -o
$output_objdir/$soname $libobjs $deplibs ${wl}-bnoentry
$compiler_flags ${wl}-bE:$export_symbols${allow_undefined_flag}~$AR
$AR_FLAGS $output_objdir/$libname$release.a $output_objdir/$soname'
        fi
        fi
        ;;

amigaos*)
        case $host_cpu in
        powerpc)
            # see comment about AmigaOS4 .so support
            archive_cmds='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
            archive_expsym_cmds=''
            ;;
        m68k)
            archive_cmds='$RM $output_objdir/a2ixlibrary.data~$ECHO
"#define NAME $libname" > $output_objdir/a2ixlibrary.data~$ECHO
"#define LIBRARY_ID 1" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR
$AR_FLAGS $lib $libobjs~$RANLIB $lib~(cd $output_objdir && a2ixlibrary
-32)'
            hardcode_libdir_flag_spec='-L$libdir'
            hardcode_minus_L=yes
            ;;
        esac
        ;;
    ;;

bsd[45]*)
        export_dynamic_flag_spec=-rdynamic

```

```

;;

cygwin* | mingw* | pw32* | cegcc*)
# When not using gcc, we currently assume that we are using
# Microsoft Visual C++.
# hardcode_libdir_flag_spec is actually meaningless, as there is
# no search path for DLLs.
case $cc_basename in
cl*)
# Native MSVC
hardcode_libdir_flag_spec=' '
allow_undefined_flag=unsupported
always_export_symbols=yes
file_list_spec='@'
# Tell ltmain to make .lib files, not .a files.
libext=lib
# Tell ltmain to make .dll files, not .so files.
shrext_cmds=".dll"
# FIXME: Setting linknames here is a bad hack.
archive_cmds='$CC -o $output_objdir/$soname $libobjs
$compiler_flags $deplibs -Wl,-dll~linknames='
archive_expsym_cmds='if test "x`$SED 1q $export_symbols`" =
xEXPORTS; then
    sed -n -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -EXPORT:\\\\\\\\\\\\\\\\1/' -
e '1\\\\\\\\!p' < $export_symbols > $output_objdir/$soname.exp;
else
    sed -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -EXPORT:\\\\\\\\\\\\\\\\1/' <
$export_symbols > $output_objdir/$soname.exp;
fi~
$CC -o $tool_output_objdir$soname $libobjs $compiler_flags
$deplibs "@$tool_output_objdir$soname.exp" -Wl,-DLL,-
IMPLIB:"$tool_output_objdir$libname.dll.lib"~
linknames='
# The linker will not automatically build a static lib if we
build a DLL.
# _LT_TAGVAR(old_archive_from_new_cmds, )='true'
enable_shared_with_static_runtimes=yes
exclude_expsyms='_NULL_IMPORT_DESCRIPTOR|_IMPORT_DESCRIPTOR_*'
export_symbols_cmds='$NM $libobjs $convenience |
$global_symbol_pipe | $SED -e '\''/^([BCDGRS])[ ]/s/.*[ ]\([^\
]*\)/\1,DATA/'\'' | $SED -e '\''/^([AITW])[ ]/s/.*[ ]//'\'' | sort |
uniq > $export_symbols'
# Don't use ranlib
old_postinstall_cmds='chmod 644 $oldlib'
postlink_cmds='lt_outputfile="@OUTPUT@"~
lt_tool_outputfile="@TOOL_OUTPUT@"~
case $lt_outputfile in
*.exe|*.EXE) ;;
*)
    lt_outputfile="$lt_outputfile.exe"
    lt_tool_outputfile="$lt_tool_outputfile.exe"
;;

```

```

        esac~
        if test "$MANIFEST_TOOL" != ":" && test -f
"$lt_outputfile.manifest"; then
            $MANIFEST_TOOL -manifest "$lt_tool_outputfile.manifest" -
outputresource:"$lt_tool_outputfile" || exit 1;
            $RM "$lt_outputfile.manifest";
        fi'
;;
*)
# Assume MSVC wrapper
hardcode_libdir_flag_spec=' '
allow_undefined_flag=unsupported
# Tell ltmain to make .lib files, not .a files.
libext=lib
# Tell ltmain to make .dll files, not .so files.
shrext_cmds=".dll"
# FIXME: Setting linknames here is a bad hack.
archive_cmds='$CC -o $lib $libobjs $compiler_flags `func_echo_all
"$deplibs" | $SED \'\'s/ -lc$//\'\'` -link -dll~linknames='
# The linker will automatically build a .lib file if we build a
DLL.
old_archive_from_new_cmds='true'
# FIXME: Should let the user specify the lib program.
old_archive_cmds='lib -OUT:$oldlib$oldobjs$old_deplibs'
enable_shared_with_static_runtimes=yes
;;
esac
;;

darwin* | rhapsody*)

```

```

archive_cmds_need_lc=no
hardcode_direct=no
hardcode_automatic=yes
hardcode_shlibpath_var=unsupported
if test "$lt_cv_ld_force_load" = "yes"; then
    whole_archive_flag_spec='`for conv in $convenience\`; do test -
n \"$conv\" && new_convenience=\"$new_convenience ${wl}-
force_load,$conv\"; done; func_echo_all \"$new_convenience\`'
else
    whole_archive_flag_spec=' '
fi
link_all_deplibs=yes
allow_undefined_flag="$lt_dar_allow_undefined"
case $cc_basename in
    ifort*) _lt_dar_can_shared=yes ;;
    *) _lt_dar_can_shared=$GCC ;;
esac
if test "$lt_dar_can_shared" = "yes"; then
    output_verbose_link_cmd=func_echo_all

```

```

        archive_cmds="\$CC -dynamiclib \$allow_undefined_flag -o \$lib
\$libobjs \$deplibs \$compiler_flags -install_name \$rpath/\$soname
\$verstring \$_lt_dar_single_mod${_lt_dsymutil}"
        module_cmds="\$CC \$allow_undefined_flag -o \$lib -bundle
\$libobjs \$deplibs \$compiler_flags${_lt_dsymutil}"
        archive_expsym_cmds="sed 's,^,_, ' < \$export_symbols >
\$output_objdir/\${libname}-symbols.expsym~\$CC -dynamiclib
\$allow_undefined_flag -o \$lib \$libobjs \$deplibs \$compiler_flags -
install_name \$rpath/\$soname \$verstring
${_lt_dar_single_mod}${_lt_dar_export_syms}${_lt_dsymutil}"
        module_expsym_cmds="sed -e 's,^,_, ' < \$export_symbols >
\$output_objdir/\${libname}-symbols.expsym~\$CC \$allow_undefined_flag
-o \$lib -bundle \$libobjs \$deplibs
\$compiler_flags${_lt_dar_export_syms}${_lt_dsymutil}"

else
ld_shlibs=no
fi

;;

dgux*)
        archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
        hardcode_libdir_flag_spec='-L$libdir'
        hardcode_shlibpath_var=no
        ;;

# FreeBSD 2.2.[012] allows us to include c++rt0.o to get C++
constructor
# support.  Future versions do this automatically, but an explicit
c++rt0.o
# does not break anything, and helps significantly (at the cost of
a little
# extra space).
freebsd2.2*)
        archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags /usr/lib/c++rt0.o'
        hardcode_libdir_flag_spec='-R$libdir'
        hardcode_direct=yes
        hardcode_shlibpath_var=no
        ;;

# Unfortunately, older versions of FreeBSD 2 do not have this
feature.
freebsd2.*)
        archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags'
        hardcode_direct=yes
        hardcode_minus_L=yes
        hardcode_shlibpath_var=no
        ;;

```

```

# FreeBSD 3 and greater uses gcc -shared to do shared libraries.
freebsd* | dragonfly*)
    archive_cmds='$CC -shared $pic_flag -o $lib $libobjs $deplibs
$compiler_flags'
    hardcode_libdir_flag_spec='-R$libdir'
    hardcode_direct=yes
    hardcode_shlibpath_var=no
    ;;

hpux9*)
    if test "$GCC" = yes; then
        archive_cmds='$RM $output_objdir/$soname~$CC -shared $pic_flag
${wl}+b ${wl}$install_libdir -o $output_objdir/$soname $libobjs
$deplibs $compiler_flags~test $output_objdir/$soname = $lib || mv
$output_objdir/$soname $lib'
    else
        archive_cmds='$RM $output_objdir/$soname~$LD -b +b
$install_libdir -o $output_objdir/$soname $libobjs $deplibs
$linker_flags~test $output_objdir/$soname = $lib || mv
$output_objdir/$soname $lib'
    fi
    hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'
    hardcode_libdir_separator=:
    hardcode_direct=yes

    # hardcode_minus_L: Not really in the search PATH,
    # but as the default location of the library.
    hardcode_minus_L=yes
    export_dynamic_flag_spec='${wl}-E'
    ;;

hpux10*)
    if test "$GCC" = yes && test "$with_gnu_ld" = no; then
        archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname ${wl}+b
${wl}$install_libdir -o $lib $libobjs $deplibs $compiler_flags'
    else
        archive_cmds='$LD -b +h $soname +b $install_libdir -o $lib
$libobjs $deplibs $linker_flags'
    fi
    if test "$with_gnu_ld" = no; then
        hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'
        hardcode_libdir_separator=:
        hardcode_direct=yes
        hardcode_direct_absolute=yes
        export_dynamic_flag_spec='${wl}-E'
        # hardcode_minus_L: Not really in the search PATH,
        # but as the default location of the library.
        hardcode_minus_L=yes
    fi
    ;;

```

```

hpux11*)
  if test "$GCC" = yes && test "$with_gnu_ld" = no; then
  case $host_cpu in
  hppa*64*)
    archive_cmds='$CC -shared ${wl}+h ${wl}$soname -o $lib $libobjs
$deplibs $compiler_flags'
    ;;
  ia64*)
    archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname
${wl}+nodefaulttrpath -o $lib $libobjs $deplibs $compiler_flags'
    ;;
  *)
    archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname
${wl}+b ${wl}$install_libdir -o $lib $libobjs $deplibs
$compiler_flags'
    ;;
  esac
  else
  case $host_cpu in
  hppa*64*)
    archive_cmds='$CC -b ${wl}+h ${wl}$soname -o $lib $libobjs
$deplibs $compiler_flags'
    ;;
  ia64*)
    archive_cmds='$CC -b ${wl}+h ${wl}$soname ${wl}+nodefaulttrpath
-o $lib $libobjs $deplibs $compiler_flags'
    ;;
  *)
    # Older versions of the 11.00 compiler do not understand -b yet
    # (HP92453-01 A.11.01.20 doesn't, HP92453-01 B.11.X.35175-
35176.GP does)
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $CC
understands -b" >&5
$as_echo_n "checking if $CC understands -b... " >&6; }
if ${lt_cv_prog_compiler_b+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_b=no
  save_LDFLAGS="$LDFLAGS"
  LDFLAGS="$LDFLAGS -b"
  echo "$lt_simple_link_test_code" > conftest.$ac_ext
  if (eval $ac_link 2>conftest.err) && test -s conftest.$ac_exeext;
then
    # The linker can only warn and ignore the option if not
recognized
    # So say no if there are warnings
    if test -s conftest.err; then
      # Append any errors to the config.log.
      cat conftest.err 1>&5
      $ECHO "$_lt_linker_boilerplate" | $SED '/^$/d' > conftest.exp
      $SED '/^$/d; /^ *+/d' conftest.err >conftest.er2

```



```

        if diff conftest.exp conftest.er2 >/dev/null; then
            lt_cv_prog_compiler__b=yes
        fi
    else
        lt_cv_prog_compiler__b=yes
    fi
fi
$RM -r conftest*
LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:"
$lt_cv_prog_compiler__b" >&5
$as_echo "$lt_cv_prog_compiler__b" >&6; }

if test x"$lt_cv_prog_compiler__b" = xyes; then
    archive_cmds='$CC -b ${wl}+h ${wl}$soname ${wl}+b
${wl}$install_libdir -o $lib $libobjs $deplibs $compiler_flags'
else
    archive_cmds='$LD -b +h $soname +b $install_libdir -o $lib
$libobjs $deplibs $linker_flags'
fi

;;
esac
fi
if test "$with_gnu_ld" = no; then
hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'
hardcode_libdir_separator=:

case $host_cpu in
hppa*64*|ia64*)
    hardcode_direct=no
    hardcode_shlibpath_var=no
    ;;
*)
    hardcode_direct=yes
    hardcode_direct_absolute=yes
    export_dynamic_flag_spec='${wl}-E'

    # hardcode_minus_L: Not really in the search PATH,
    # but as the default location of the library.
    hardcode_minus_L=yes
    ;;
;;
esac
fi
;;

irix5* | irix6* | nonstopux*)
    if test "$GCC" = yes; then
        archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname ${wl}$soname `test -n "$verstring" &&

```

```

func_echo_all "${wl}-set_version ${wl}$verstring" ` ${wl}-
update_registry ${wl}${output_objdir}/so_locations -o $lib'
    # Try to use the -exported_symbol ld option, if it does not
    # work, assume that -exports_file does not work either and
    # implicitly export all symbols.
    # This should be the same for all languages, so no per-tag cache
variable.
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the
$host_os linker accepts -exported_symbol" >&5
$as_echo_n "checking whether the $host_os linker accepts -
exported_symbol... " >&6; }
if ${lt_cv_irix_exported_symbol+:} false; then :
    $as_echo_n "(cached) " >&6
else
    save_LDFLAGS="$LDFLAGS"
    LDFLAGS="$LDFLAGS -shared ${wl}-exported_symbol ${wl}foo
${wl}-update_registry ${wl}/dev/null"
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
int foo (void) { return 0; }
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    lt_cv_irix_exported_symbol=yes
else
    lt_cv_irix_exported_symbol=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
    LDFLAGS="$save_LDFLAGS"
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_irix_exported_symbol" >&5
$as_echo "$lt_cv_irix_exported_symbol" >&6; }
    if test "$lt_cv_irix_exported_symbol" = yes; then
        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname ${wl}$soname `test -n "$verstring" &&
func_echo_all "${wl}-set_version ${wl}$verstring" ` ${wl}-
update_registry ${wl}${output_objdir}/so_locations ${wl}-exports_file
${wl}$export_symbols -o $lib'
    fi
    else
        archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags -
soname $soname `test -n "$verstring" && func_echo_all "-set_version
$verstring" ` -update_registry ${output_objdir}/so_locations -o $lib'
        archive_expsym_cmds='$CC -shared $libobjs $deplibs
$compiler_flags -soname $soname `test -n "$verstring" && func_echo_all
"-set_version $verstring" ` -update_registry
${output_objdir}/so_locations -exports_file $export_symbols -o $lib'
    fi
    archive_cmds_need_lc='no'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    hardcode_libdir_separator=:

```

```

inherit_rpath=yes
link_all_deplibs=yes
;;

netbsd*)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
        archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags' # a.out
    else
        archive_cmds='$LD -shared -o $lib $libobjs $deplibs
$linker_flags' # ELF
    fi
    hardcode_libdir_flag_spec='-R$libdir'
    hardcode_direct=yes
    hardcode_shlibpath_var=no
    ;;

newsos6)
    archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_direct=yes
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    hardcode_libdir_separator=:
    hardcode_shlibpath_var=no
    ;;

*nto* | *qnx*)
    ;;

openbsd*)
    if test -f /usr/libexec/ld.so; then
        hardcode_direct=yes
        hardcode_shlibpath_var=no
        hardcode_direct_absolute=yes
        if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
            archive_cmds='$CC -shared $pic_flag -o $lib $libobjs $deplibs
$compiler_flags'
            archive_expsym_cmds='$CC -shared $pic_flag -o $lib $libobjs
$deplibs $compiler_flags ${wl}-retain-symbols-file,$export_symbols'
            hardcode_libdir_flag_spec='${wl}-rpath,$libdir'
            export_dynamic_flag_spec='${wl}-E'
        else
            case $host_os in
                openbsd[01].* | openbsd2.[0-7] | openbsd2.[0-7].*)
                    archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags'
                    hardcode_libdir_flag_spec='-R$libdir'
                ;;
            *)
                archive_cmds='$CC -shared $pic_flag -o $lib $libobjs
$deplibs $compiler_flags'
            ;;
        esac
    fi

```

```

        hardcode_libdir_flag_spec='${wl}-rpath,$libdir'
        ;;
    esac
fi
else
ld_shlibs=no
fi
;;

os2*)
    hardcode_libdir_flag_spec='-L$libdir'
    hardcode_minus_L=yes
    allow_undefined_flag=unsupported
    archive_cmds='$ECHO "LIBRARY $libname INITINSTANCE" >
$output_objdir/$libname.def~$ECHO "DESCRIPTION \"$libname\"" >>
$output_objdir/$libname.def~echo DATA >>
$output_objdir/$libname.def~echo " SINGLE NONSHARED" >>
$output_objdir/$libname.def~echo EXPORTS >>
$output_objdir/$libname.def~emxexp $libobjs >>
$output_objdir/$libname.def~$CC -Zdll -Zcrtdll -o $lib $libobjs
$deplibs $compiler_flags $output_objdir/$libname.def'
    old_archive_from_new_cmds='emximp -o $output_objdir/$libname.a
$output_objdir/$libname.def'
    ;;

osf3*)
    if test "$GCC" = yes; then
        allow_undefined_flag=' ${wl}-expect_unresolved ${wl}\*'
        archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags ${wl}-soname ${wl}$soname `test -n
"$sverstring" && func_echo_all "${wl}-set_version ${wl}$sverstring"`
${wl}-update_registry ${wl}${output_objdir}/so_locations -o $lib'
    else
        allow_undefined_flag=' -expect_unresolved \*'
        archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags -soname $soname `test -n "$sverstring" &&
func_echo_all "-set_version $sverstring"` -update_registry
${output_objdir}/so_locations -o $lib'
    fi
    archive_cmds_need_lc='no'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    hardcode_libdir_separator=:
    ;;

osf4* | osf5*)    # as osf3* with the addition of -msym flag
    if test "$GCC" = yes; then
        allow_undefined_flag=' ${wl}-expect_unresolved ${wl}\*'
        archive_cmds='$CC -shared${allow_undefined_flag} $pic_flag
$libobjs $deplibs $compiler_flags ${wl}-msym ${wl}-soname ${wl}$soname
`test -n "$sverstring" && func_echo_all "${wl}-set_version
${wl}$sverstring"` ${wl}-update_registry
${wl}${output_objdir}/so_locations -o $lib'

```

```

hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
else
  allow_undefined_flag=' -expect_unresolved \*'
  archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags -msym -soname $soname `test -n "$verstring"
&& func_echo_all "-set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib'
  archive_expsym_cmds='for i in `cat $export_symbols`; do printf
"%s %s\n" -exported_symbol "\$i" >> $lib.exp; done; printf "%s\n" "-
hidden">> $lib.exp~
  $CC -shared${allow_undefined_flag} ${wl}-input ${wl}$lib.exp
$compiler_flags $libobjs $deplibs -soname $soname `test -n
"$verstring" && $ECHO "-set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib~$RM $lib.exp'

# Both c and cxx compiler support -rpath directly
hardcode_libdir_flag_spec='-rpath $libdir'
fi
archive_cmds_need_lc='no'
hardcode_libdir_separator=:
;;

solaris*)
  no_undefined_flag=' -z defs'
  if test "$GCC" = yes; then
    wlarc='${wl}'
    archive_cmds='$CC -shared $pic_flag ${wl}-z ${wl}text ${wl}-h
${wl}$soname -o $lib $libobjs $deplibs $compiler_flags'
    archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
  $CC -shared $pic_flag ${wl}-z ${wl}text ${wl}-M ${wl}$lib.exp
${wl}-h ${wl}$soname -o $lib $libobjs $deplibs $compiler_flags~$RM
$lib.exp'
  else
    case ` $CC -V 2>&1 ` in
      *"Compilers 5.0"*)
        wlarc=''
        archive_cmds='$LD -G${allow_undefined_flag} -h $soname -o $lib
$libobjs $deplibs $linker_flags'
        archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
  $LD -G${allow_undefined_flag} -M $lib.exp -h $soname -o $lib
$libobjs $deplibs $linker_flags~$RM $lib.exp'
        ;;
      *)
        wlarc='${wl}'
        archive_cmds='$CC -G${allow_undefined_flag} -h $soname -o $lib
$libobjs $deplibs $compiler_flags'

```

```

        archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)\/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
        $CC -G${allow_undefined_flag} -M $lib.exp -h $soname -o $lib
$libobjs $deplibs $compiler_flags~$RM $lib.exp'
        ;;
    esac
    fi
    hardcode_libdir_flag_spec='-R$libdir'
    hardcode_shlibpath_var=no
    case $host_os in
    solaris2.[0-5] | solaris2.[0-5].*) ;;
    *)
        # The compiler driver will combine and reorder linker options,
        # but understands '-z linker_flag'. GCC discards it without
        ` $wl',
        # but is careful enough not to reorder.
        # Supported since Solaris 2.6 (maybe 2.5.1?)
        if test "$GCC" = yes; then
            whole_archive_flag_spec='${wl}-z ${wl}allextract$convenience
${wl}-z ${wl}defaultextract'
        else
            whole_archive_flag_spec='-z allextract$convenience -z
defaultextract'
        fi
        ;;
    esac
    link_all_deplibs=yes
    ;;

sunos4*)
    if test "x$host_vendor" = xsequent; then
        # Use $CC to link under sequent, because it throws in some extra
        .o
        # files that make .init and .fini sections work.
        archive_cmds='$CC -G ${wl}-h $soname -o $lib $libobjs $deplibs
$compiler_flags'
        else
            archive_cmds='$LD -assert pure-text -Bstatic -o $lib $libobjs
$deplibs $linker_flags'
        fi
        hardcode_libdir_flag_spec='-L$libdir'
        hardcode_direct=yes
        hardcode_minus_L=yes
        hardcode_shlibpath_var=no
        ;;
    sysv4)
        case $host_vendor in
        sni)
            archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'

```

```

        hardcode_direct=yes # is this really true???
;;
siemens)
    ## LD is ld it makes a PLAMLIB
    ## CC just makes a GrossModule.
    archive_cmds='$LD -G -o $lib $libobjs $deplibs $linker_flags'
    reload_cmds='$CC -r -o $output$reload_objs'
    hardcode_direct=no
    ;;
motorola)
    archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_direct=no #Motorola manual says yes, but my tests say
they lie
    ;;
esac
runpath_var='LD_RUN_PATH'
hardcode_shlibpath_var=no
;;

sysv4.3*)
    archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_shlibpath_var=no
    export_dynamic_flag_spec='-Bexport'
    ;;

sysv4*MP*)
    if test -d /usr/nec; then
        archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
        hardcode_shlibpath_var=no
        runpath_var=LD_RUN_PATH
        hardcode_runpath_var=yes
        ld_shlibs=yes
    fi
    ;;

sysv4*uw2* | sysv5OpenUNIX* | sysv5UnixWare7.[01].[10]* |
unixware7* | sco3.2v5.0.[024]*)
    no_undefined_flag='${wl}-z,text'
    archive_cmds_need_lc=no
    hardcode_shlibpath_var=no
    runpath_var='LD_RUN_PATH'

    if test "$GCC" = yes; then
        archive_cmds='$CC -shared ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
        archive_expsym_cmds='$CC -shared ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
    else

```

```

        archive_cmds='$CC -G ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
        archive_expsym_cmds='$CC -G ${wl}-Bexport:$export_symbols ${wl}-
h,$soname -o $lib $libobjs $deplibs $compiler_flags'
        fi
        ;;

sysv5* | sco3.2v5* | sco5v6*)
# Note: We can NOT use -z defs as we might desire, because we do
not
# link with -lc, and that would cause any symbols used from libc
to
# always be unresolved, which means just about no library would
# ever link correctly.  If we're not using GNU ld we use -z text
# though, which does catch some bad symbols but isn't as heavy-
handed
# as -z defs.
no_undefined_flag='${wl}-z,text'
allow_undefined_flag='${wl}-z,nodefs'
archive_cmds_need_lc=no
hardcode_shlibpath_var=no
hardcode_libdir_flag_spec='${wl}-R,$libdir'
hardcode_libdir_separator=':'
link_all_deplibs=yes
export_dynamic_flag_spec='${wl}-Bexport'
runpath_var='LD_RUN_PATH'

if test "$GCC" = yes; then
        archive_cmds='$CC -shared ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
        archive_expsym_cmds='$CC -shared ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
        else
        archive_cmds='$CC -G ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
        archive_expsym_cmds='$CC -G ${wl}-Bexport:$export_symbols ${wl}-
h,$soname -o $lib $libobjs $deplibs $compiler_flags'
        fi
        ;;

uts4*)
        archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
        hardcode_libdir_flag_spec='-L$libdir'
        hardcode_shlibpath_var=no
        ;;

*)
        ld_shlibs=no
        ;;
esac

```



```

    if test x$host_vendor = xsni; then
        case $host in
            sysv4 | sysv4.2uw2* | sysv4.3* | sysv5*)
                export_dynamic_flag_spec='${wl}-Blargedynsym'
                ;;
            esac
        fi
    fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ld_shlibs" >&5
$as_echo "$ld_shlibs" >&6; }
test "$ld_shlibs" = no && can_build_shared=no

with_gnu_ld=$with_gnu_ld

#
# Do we need to explicitly link libc?
#
case "x$archive_cmds_need_lc" in
x|xyes)
    # Assume -lc should be added
    archive_cmds_need_lc=yes

    if test "$enable_shared" = yes && test "$GCC" = yes; then
        case $archive_cmds in
            *'~'*)
                # FIXME: we may have to deal with multi-command sequences.
                ;;
            '$CC '* )
                # Test whether the compiler implicitly links with -lc since on
some
                # systems, -lgcc has to come before -lc. If gcc already passes -
lc
                # to ld, don't add -lc before -lgcc.
                { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether -lc
should be explicitly linked in" >&5
$as_echo_n "checking whether -lc should be explicitly linked in... "
>&6; }

```

```

if ${lt_cv_archive_cmds_need_lc+:} false; then :
  $as_echo_n "(cached) " >&6
else
  $RM conftest*
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext

  if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
    (eval $ac_compile) 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
    test $ac_status = 0; } 2>conftest.err; then
    soname=conftest
    lib=conftest
    libobjs=conftest.$ac_objext
    deplibs=
    wl=$lt_prog_compiler_wl
    pic_flag=$lt_prog_compiler_pic
    compiler_flags=-v
    linker_flags=-v
    verstring=
    output_objdir=.
    libname=conftest
    lt_save_allow_undefined_flag=$allow_undefined_flag
    allow_undefined_flag=
    if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$archive_cmds 2\>\&1 \\\ $GREP \" -lc \" \>/dev/null 2\>\&1\""; } >&5
      (eval $archive_cmds 2\>\&1 \\\ $GREP \" -lc \" \>/dev/null 2\>\&1)
    2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
    test $ac_status = 0; }
      then
        lt_cv_archive_cmds_need_lc=no
      else
        lt_cv_archive_cmds_need_lc=yes
      fi
    allow_undefined_flag=$lt_save_allow_undefined_flag
  else
    cat conftest.err 1>&5
  fi
  $RM conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_archive_cmds_need_lc" >&5
$as_echo "$lt_cv_archive_cmds_need_lc" >&6; }
  archive_cmds_need_lc=$lt_cv_archive_cmds_need_lc
  ;;
esac
fi
;;

```

esac


```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking dynamic linker
characteristics" >&5
$as_echo_n "checking dynamic linker characteristics... " >&6; }
```

```

if test "$GCC" = yes; then
  case $host_os in
    darwin*) lt_awk_arg="/^libraries:/,/LR/" ;;
    *) lt_awk_arg="/^libraries:/" ;;
  esac
  case $host_os in
    mingw* | cegcc*) lt_sed_strip_eq="s,=\([A-Za-z]:\),\1,g" ;;
    *) lt_sed_strip_eq="s,=/,/,g" ;;
  esac
  lt_search_path_spec=`$CC -print-search-dirs | awk $lt_awk_arg | $SED
-e "s/^libraries:/" -e $lt_sed_strip_eq`
  case $lt_search_path_spec in
    *\;*)
      # if the path contains ";" then we assume it to be the separator
      # otherwise default to the standard path separator (i.e. ":") - it
      is
      # assumed that no part of a normal pathname contains ";" but that
      should
      # okay in the real world where ";" in dirpaths is itself
      problematic.
      lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED 's;/;/
/g'`
      ;;
    *)
      lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED
"s/$PATH_SEPARATOR/ /g'`
      ;;
  esac
  # Ok, now we have the path, separated by spaces, we can step through
  it
  # and add multilib dir if necessary.
  lt_tmp_lt_search_path_spec=
  lt_multi_os_dir=`$CC $CPPFLAGS $CFLAGS $LDFLAGS -print-multi-os-
directory 2>/dev/null`
  for lt_sys_path in $lt_search_path_spec; do
    if test -d "$lt_sys_path/$lt_multi_os_dir"; then
      lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec
$lt_sys_path/$lt_multi_os_dir"
    else
      test -d "$lt_sys_path" && \
        lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec
$lt_sys_path"
    fi
  done
  lt_search_path_spec=`$ECHO "$lt_tmp_lt_search_path_spec" | awk '
BEGIN {RS=" "; FS="|\n";} {
  lt_foo="";
  lt_count=0;
  for (lt_i = NF; lt_i > 0; lt_i--) {
    if ($lt_i != "" && $lt_i != ".") {
      if ($lt_i == "..") {

```

```

        lt_count++;
    } else {
        if (lt_count == 0) {
            lt_foo="/" $lt_i lt_foo;
        } else {
            lt_count--;
        }
    }
}
}
if (lt_foo != "") { lt_freq[lt_foo]++; }
if (lt_freq[lt_foo] == 1) { print lt_foo; }
}'`
# AWK program above erroneously prepends '/' to C:/dos/paths
# for these hosts.
case $host_os in
    mingw* | cegcc*) lt_search_path_spec=`$ECHO "$lt_search_path_spec"
|\
        $SED 's,/\/\([A-Za-z]:\),\1,g'` ;;
    esac
    sys_lib_search_path_spec=`$ECHO "$lt_search_path_spec" | $lt_NL2SP`
else
    sys_lib_search_path_spec="/lib /usr/lib /usr/local/lib"
fi
library_names_spec=
libname_spec='lib$name'
soname_spec=
shrext_cmds=".so"
postinstall_cmds=
postuninstall_cmds=
finish_cmds=
finish_eval=
shlibpath_var=
shlibpath_overrides_runpath=unknown
version_type=none
dynamic_linker="$host_os ld.so"
sys_lib_dlsearch_path_spec="/lib /usr/lib"
need_lib_prefix=unknown
hardcode_into_libs=no

# when you set need_version to no, make sure it does not cause -
set_version
# flags to be left without arguments
need_version=unknown

case $host_os in
aix3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname.a'
    shlibpath_var=LIBPATH

```

```

# AIX 3 has no versioning support, so we append a major version to
the name.
soname_spec='${libname}${release}${shared_ext}$major'
;;

aix[4-9]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    hardcode_into_libs=yes
    if test "$host_cpu" = ia64; then
        # AIX 5 supports IA64
        library_names_spec='${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext}$versuffix $libname${shared_ext}'
        shlibpath_var=LD_LIBRARY_PATH
    else
        # With GCC up to 2.95.x, collect2 would create an import file
        # for dependence libraries. The import file would start with
        # the line `#! .'. This would cause the generated library to
        # depend on `.', always an invalid library. This was fixed in
        # development snapshots of GCC prior to 3.0.
        case $host_os in
            aix4 | aix4.[01] | aix4.[01].*)
                if { echo '#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__
>= 97)'
                    echo ' yes '
                    echo '#endif'; } | ${CC} -E - | $GREP yes > /dev/null; then
                    :
                else
                    can_build_shared=no
                fi
            ;;
        esac
        # AIX (on Power*) has no versioning support, so currently we can
not hardcode correct
        # soname into executable. Probably we can add versioning support
to
        # collect2, so additional links can be useful in future.
        if test "$aix_use_runtimelinking" = yes; then
            # If using run time linking (on AIX 4.2 or later) use
lib<name>.so
            # instead of lib<name>.a to let people know that these are not
            # typical AIX shared libraries.
            library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
        else
            # We preserve .a as extension for shared libraries through
AIX4.2
            # and later when we are not doing run time linking.
            library_names_spec='${libname}${release}.a $libname.a'

```



```

        soname_spec='${libname}${release}${shared_ext}$major'
    fi
    shlibpath_var=LIBPATH
fi
;;

amigaos*)
    case $host_cpu in
        powerpc)
            # Since July 2007 AmigaOS4 officially supports .so libraries.
            # When compiling the executable, add -use-dynld -Lsojbs: to the
            compileline.
            library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
            ;;
        m68k)
            library_names_spec='$libname.ixlibrary $libname.a'
            # Create ${libname}_ixlibrary.a entries in /sys/libs.
            finish_eval='for lib in `ls $libdir/*.ixlibrary 2>/dev/null`; do
libname=`func_echo_all "$lib" | $SED
\'\'s%^\./\([^/]*\)\.ixlibrary$%\1%\'\''; test $RM
/sys/libs/${libname}_ixlibrary.a; $show "cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a"; cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a || exit 1; done'
            ;;
    esac
;;

beos*)
    library_names_spec='${libname}${shared_ext}'
    dynamic_linker="$host_os ld.so"
    shlibpath_var=LIBRARY_PATH
;;

bsdi[45]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    sys_lib_search_path_spec="/shlib /usr/lib /usr/X11/lib
/usr/contrib/lib /lib /usr/local/lib"
    sys_lib_dlsearch_path_spec="/shlib /usr/lib /usr/local/lib"
    # the default ld.so.conf also contains /usr/contrib/lib and
    # /usr/X11R6/lib (/usr/X11 is a link to /usr/X11R6), but let us
allow
    # libtool to hard-code these into programs
;;

```

```

cygwin* | mingw* | pw32* | cegcc*)
    version_type=windows
    shrext_cmds=".dll"
    need_version=no
    need_lib_prefix=no

    case $GCC,$cc_basename in
    yes,*)
        # gcc
        library_names_spec='$libname.dll.a'
        # DLL is installed to $(libdir)/../bin by postinstall_cmds
        postinstall_cmds='base_file=`basename \${file}`~
            dlpath=`$SHELL 2>&1 -c '\''. $dir/\'''\${base_file}'\'''; echo
\${dlname}'\''`~
            dldir=$destdir/`dirname \${dlpath}`~
            test -d \${dldir} || mkdir -p \${dldir}~
            $install_prog $dir/\${dlname} \${dldir}/\${dlname}~
            chmod a+x \${dldir}/\${dlname}~
            if test -n '\''$striplib'\'' && test -n '\''$striplib'\''; then
                eval '\''$striplib \${dldir}/\${dlname}'\'' || exit \${?};
            fi'
        postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; echo
\${dlname}'\''`~
            dlpath=$dir/\${dldll}~
            $RM \${dlpath}'
        shlibpath_overrides_runpath=yes

    case $host_os in
    cygwin*)
        # Cygwin DLLs use 'cyg' prefix rather than 'lib'
        soname_spec=`echo \${libname} | sed -e 's/^lib/cyg/'``echo
\${release} | $SED -e 's/[.]/-/g'`\${versuffix}\${shared_ext}'

        sys_lib_search_path_spec="$sys_lib_search_path_spec
/usr/lib/w32api"
        ;;
    mingw* | cegcc*)
        # MinGW DLLs use traditional 'lib' prefix
        soname_spec='\${libname}`echo \${release} | $SED -e 's/[.]/-
/g'`\${versuffix}\${shared_ext}'
        ;;
    pw32*)
        # pw32 DLLs use 'pw' prefix rather than 'lib'
        library_names_spec=`echo \${libname} | sed -e 's/^lib/pw/'``echo
\${release} | $SED -e 's/[.]/-/g'`\${versuffix}\${shared_ext}'
        ;;
    esac
    dynamic_linker='Win32 ld.exe'
    ;;

*,cl*)
    # Native MSVC

```

```

libname_spec='$name'
soname_spec='${libname}`echo ${release} | $SED -e 's/[.]\/-
/g'`${versuffix}${shared_ext}'
library_names_spec='${libname}.dll.lib'

case $build_os in
mingw*)
  sys_lib_search_path_spec=
  lt_save_ifs=$IFS
  IFS=';'
  for lt_path in $LIB
  do
    IFS=$lt_save_ifs
    # Let DOS variable expansion print the short 8.3 style file
name.
    lt_path=`cd "$lt_path" 2>/dev/null && cmd //C "for %i in (".")
do @echo %~si"`
    sys_lib_search_path_spec="$sys_lib_search_path_spec $lt_path"
  done
  IFS=$lt_save_ifs
  # Convert to MSYS style.
  sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
sed -e 's|\\|/|g' -e 's| \\\([a-zA-Z]\\|):| /\\1|g' -e 's|^|/|'`
  ;;
cygwin*)
  # Convert to unix form, then to dos form, then back to unix form
  # but this time dos style (no spaces!) so that the unix form
looks
  # like /cygdrive/c/PROGRA~1:/cygdr...
  sys_lib_search_path_spec=`cygpath --path --unix "$LIB"`
  sys_lib_search_path_spec=`cygpath --path --dos
"$sys_lib_search_path_spec" 2>/dev/null`
  sys_lib_search_path_spec=`cygpath --path --unix
"$sys_lib_search_path_spec" | $SED -e "s/$PATH_SEPARATOR/ /g"`
  ;;
*)
  sys_lib_search_path_spec="$LIB"
  if $ECHO "$sys_lib_search_path_spec" | $GREP '[c-zA-Z]:/'
>/dev/null; then
    # It is most probably a Windows format PATH.
    sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
$SED -e 's/;/ /g'`
  else
    sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
$SED -e "s/$PATH_SEPARATOR/ /g"`
  fi
  # FIXME: find the short name or the path components, as spaces
are
  # common. (e.g. "Program Files" -> "PROGRA~1")
  ;;
esac

```

```

# DLL is installed to $(libdir)/../bin by postinstall_cmds
postinstall_cmds='base_file=`basename \${file}`~
  dlpath=`$SHELL 2>&1 -c '\``. $dir/\``'\${base_file}'\``i; echo
\${dlname}'\``~
  dldir=$destdir/`dirname \${dlpath}`~
  test -d \${dldir} || mkdir -p \${dldir}~
  $install_prog $dir/\${dlname} \${dldir}/\${dlname}'
postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\``. $file; echo
\${dlname}'\``~
  dlpath=$dir/\${dldll}~
  $RM \${dlpath}'
shlibpath_overrides_runpath=yes
dynamic_linker='Win32 link.exe'
;;

*)
# Assume MSVC wrapper
library_names_spec='${libname}`echo \${release} | $SED -e 's/[.]/-/g'`${versuffix}\${shared_ext} $libname.lib'
dynamic_linker='Win32 ld.exe'
;;
esac
# FIXME: first we should search . and the directory the executable
is in
shlibpath_var=PATH
;;

darwin* | rhapsody*)
dynamic_linker="$host_os dyld"
version_type=darwin
need_lib_prefix=no
need_version=no
library_names_spec='${libname}\${release}\${major}\${shared_ext}
\${libname}\${shared_ext}'
soname_spec='${libname}\${release}\${major}\${shared_ext}'
shlibpath_overrides_runpath=yes
shlibpath_var=DYLD_LIBRARY_PATH
shrext_cmds='`test .$module = .yes && echo .so || echo .dylib`'

sys_lib_search_path_spec="$sys_lib_search_path_spec /usr/local/lib"
sys_lib_dlsearch_path_spec='/usr/local/lib /lib /usr/lib'
;;

dgux*)
version_type=linux # correct to gnu/linux during the next big
refactor
need_lib_prefix=no
need_version=no
library_names_spec='${libname}\${release}\${shared_ext}\${versuffix}
\${libname}\${release}\${shared_ext}\${major} $libname\${shared_ext}'
soname_spec='${libname}\${release}\${shared_ext}\${major}'
shlibpath_var=LD_LIBRARY_PATH

```

```

;;

freebsd* | dragonfly*)
# DragonFly does not have aout.  When/if they implement a new
# versioning mechanism, adjust this.
if test -x /usr/bin/objformat; then
    objformat=`/usr/bin/objformat`
else
    case $host_os in
        freebsd[23].*) objformat=aout ;;
        *) objformat=elf ;;
    esac
fi
version_type=freebsd-$objformat
case $version_type in
    freebsd-elf*)
        library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext} $libname${shared_ext}'
        need_version=no
        need_lib_prefix=no
        ;;
    freebsd-*)
        library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname${shared_ext}$versuffix'
        need_version=yes
        ;;
    esac
shlibpath_var=LD_LIBRARY_PATH
case $host_os in
    freebsd2.*)
        shlibpath_overrides_runpath=yes
        ;;
    freebsd3.[01]* | freebsdelf3.[01]*)
        shlibpath_overrides_runpath=yes
        hardcode_into_libs=yes
        ;;
    freebsd3.[2-9]* | freebsdelf3.[2-9]* | \
    freebsd4.[0-5] | freebsdelf4.[0-5] | freebsd4.1.1 | freebsdelf4.1.1)
        shlibpath_overrides_runpath=no
        hardcode_into_libs=yes
        ;;
    *) # from 4.6 on, and DragonFly
        shlibpath_overrides_runpath=yes
        hardcode_into_libs=yes
        ;;
    esac
;;

gnu*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no

```

```

    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;

haiku*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    dynamic_linker="$host_os runtime_loader"
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    sys_lib_dlsearch_path_spec='/boot/home/config/lib /boot/common/lib
/boot/system/lib'
    hardcode_into_libs=yes
    ;;

hpux9* | hpux10* | hpux11*)
    # Give a soname corresponding to the major version so that dld.sl
refuses to
    # link against other versions.
    version_type=sunos
    need_lib_prefix=no
    need_version=no
    case $host_cpu in
    ia64*)
        shrext_cmds='.so'
        hardcode_into_libs=yes
        dynamic_linker="$host_os dld.so"
        shlibpath_var=LD_LIBRARY_PATH
        shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
        library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
        soname_spec='${libname}${release}${shared_ext}$major'
        if test "X$HPUX_IA64_MODE" = X32; then
            sys_lib_search_path_spec="/usr/lib/hpux32 /usr/local/lib/hpux32
/usr/local/lib"
        else
            sys_lib_search_path_spec="/usr/lib/hpux64 /usr/local/lib/hpux64"
        fi
        sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
    hppa*64*)
        shrext_cmds='.sl'

```

```

hardcode_into_libs=yes
dynamic_linker="$host_os dld.sl"
shlibpath_var=LD_LIBRARY_PATH # How should we handle SHLIB_PATH
shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
sys_lib_search_path_spec="/usr/lib/pa20_64 /usr/ccs/lib/pa20_64"
sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
;;
*)
shrext_cmds='.sl'
dynamic_linker="$host_os dld.sl"
shlibpath_var=SHLIB_PATH
shlibpath_overrides_runpath=no # +s is required to enable
SHLIB_PATH
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
;;
esac
# HP-UX runs *really* slowly unless shared libraries are mode 555,
...
postinstall_cmds='chmod 555 $lib'
# or fails outright, so override atomically:
install_override_mode=555
;;

interix[3-9]*)
version_type=linux # correct to gnu/linux during the next big
refactor
need_lib_prefix=no
need_version=no
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
dynamic_linker='Interix 3.x ld.so.1 (PE, like ELF)'
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=no
hardcode_into_libs=yes
;;

irix5* | irix6* | nonstopux*)
case $host_os in
nonstopux*) version_type=nonstopux ;;
*)
if test "$lt_cv_prog_gnu_ld" = yes; then
version_type=linux # correct to gnu/linux during the next
big refactor
else
version_type=irix
fi ;;

```

```

esac
need_lib_prefix=no
need_version=no
soname_spec='${libname}${release}${shared_ext}$major'
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext} $libname${shared_ext}'
case $host_os in
  irix5* | nonstopux*)
    libsuff= shlibsuff=
    ;;
*)
  case $LD in # libtool.m4 will add one of these switches to LD
    *-32|*" -32 " | *-melf32bsmip|*" -melf32bsmip ")
      libsuff= shlibsuff= libmagic=32-bit;;
    *-n32|*" -n32 " | *-melf32bmipn32|*" -melf32bmipn32 ")
      libsuff=32 shlibsuff=N32 libmagic=N32;;
    *-64|*" -64 " | *-melf64bmip|*" -melf64bmip ")
      libsuff=64 shlibsuff=64 libmagic=64-bit;;
    *) libsuff= shlibsuff= libmagic=never-match;;
  esac
  ;;
esac
shlibpath_var=LD_LIBRARY${shlibsuff}_PATH
shlibpath_overrides_runpath=no
sys_lib_search_path_spec="/usr/lib${libsuff} /lib${libsuff}
/usr/local/lib${libsuff}"
sys_lib_dldsearch_path_spec="/usr/lib${libsuff} /lib${libsuff}"
hardcode_into_libs=yes
;;

# No shared lib support for Linux oldld, aout, or coff.
linux*oldld* | linux*aout* | linux*coff*)
  dynamic_linker=no
  ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu)
  version_type=linux # correct to gnu/linux during the next big
  refactor
  need_lib_prefix=no
  need_version=no
  library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
  soname_spec='${libname}${release}${shared_ext}$major'
  finish_cmds='PATH="\$PATH:/sbin" ldconfig -n $libdir'
  shlibpath_var=LD_LIBRARY_PATH
  shlibpath_overrides_runpath=no

  # Some binutils ld are patched to set DT_RUNPATH
  if ${lt_cv_shlibpath_overrides_runpath+:} false; then :
    $as_echo_n "(cached) " >&6

```



```

else
  lt_cv_shlibpath_overrides_runpath=no
  save_LDFLAGS=$LDFLAGS
  save_libdir=$libdir
  eval "libdir=/foo; wl=\"\$lt_prog_compiler_wl\"; \
    LDFLAGS=\"\$LDFLAGS \$hardcode_libdir_flag_spec\""
  cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  if ($OBJDUMP -p conftest$ac_exeext) 2>/dev/null | grep
"RUNPATH.*$libdir" >/dev/null; then :
    lt_cv_shlibpath_overrides_runpath=yes
  fi
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
  LDFLAGS=$save_LDFLAGS
  libdir=$save_libdir

fi

shlibpath_overrides_runpath=$lt_cv_shlibpath_overrides_runpath

# This implies no fast_install, which is unacceptable.
# Some rework will be needed to allow for fast_install
# before this can be enabled.
hardcode_into_libs=yes

# Append ld.so.conf contents to the search path
if test -f /etc/ld.so.conf; then
  lt_ld_extra=`awk '/^include / { system(sprintf("cd /etc; cat %s
2>/dev/null", \$2)); skip = 1; } { if (!skip) print \$0; skip = 0; }'
< /etc/ld.so.conf | $SED -e 's/#.*//;/^[ ]*hwcap[ ]/d;s/[: , ]/
/g;s/[^=]*$//;s/[^= ]* / /g;s/"//g;/^$/d' | tr '\n' ' '`
  sys_lib_dlsearch_path_spec="/lib /usr/lib $lt_ld_extra"
fi

# We used to test for /lib/ld.so.1 and disable shared libraries on
# powerpc, because MkLinux only supported shared libraries with the
# GNU dynamic linker. Since this was broken with cross compilers,
# most powerpc-linux boxes support dynamic linking these days and
# people can always --disable-shared, the test was removed, and we
# assume the GNU/Linux dynamic linker is in use.

```

```

dynamic_linker='GNU/Linux ld.so'
;;

netbsd*)
version_type=sunos
need_lib_prefix=no
need_version=no
if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
    dynamic_linker='NetBSD (a.out) ld.so'
else
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    dynamic_linker='NetBSD ld.elf_so'
fi
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=yes
hardcode_into_libs=yes
;;

newsos6)
version_type=linux # correct to gnu/linux during the next big
refactor
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=yes
;;

*nto* | *qnx*)
version_type=qnx
need_lib_prefix=no
need_version=no
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=no
hardcode_into_libs=yes
dynamic_linker='ldqnx.so'
;;

openbsd*)
version_type=sunos
sys_lib_dlsearch_path_spec="/usr/lib"
need_lib_prefix=no
# Some older versions of OpenBSD (3.3 at least) *do* need versioned
libs.
case $host_os in

```

```

        openbsd3.3 | openbsd3.3.*)    need_version=yes ;;
    *)                                need_version=no  ;;
esac
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
finish_cmds='PATH="$PATH:/sbin" ldconfig -m $libdir'
shlibpath_var=LD_LIBRARY_PATH
if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
    case $host_os in
        openbsd2.[89] | openbsd2.[89].*)
            shlibpath_overrides_runpath=no
            ;;
        *)
            shlibpath_overrides_runpath=yes
            ;;
    esac
else
    shlibpath_overrides_runpath=yes
fi
;;

os2*)
    libname_spec='$name'
    shrext_cmds=".dll"
    need_lib_prefix=no
    library_names_spec='$libname${shared_ext} $libname.a'
    dynamic_linker='OS/2 ld.exe'
    shlibpath_var=LIBPATH
    ;;

osf3* | osf4* | osf5*)
    version_type=osf
    need_lib_prefix=no
    need_version=no
    soname_spec='${libname}${release}${shared_ext}$major'
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH
    sys_lib_search_path_spec="/usr/shlib /usr/ccs/lib /usr/lib/cmplrs/cc
/usr/lib /usr/local/lib /var/shlib"
    sys_lib_dlsearch_path_spec="$sys_lib_search_path_spec"
    ;;

rdos*)
    dynamic_linker=no
    ;;

solaris*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no

```

```

    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    # ldd complains unless libraries are executable
    postinstall_cmds='chmod +x $lib'
    ;;

sunos4*)
    version_type=sunos
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/usr/etc" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    if test "$with_gnu_ld" = yes; then
        need_lib_prefix=no
    fi
    need_version=yes
    ;;

sysv4 | sysv4.3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    case $host_vendor in
        sni)
            shlibpath_overrides_runpath=no
            need_lib_prefix=no
            runpath_var=LD_RUN_PATH
            ;;
        siemens)
            need_lib_prefix=no
            ;;
        motorola)
            need_lib_prefix=no
            need_version=no
            shlibpath_overrides_runpath=no
            sys_lib_search_path_spec='/lib /usr/lib /usr/ccs/lib'
            ;;
    esac
    ;;

sysv4*MP*)
    if test -d /usr/nec ;then

```

```

    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='$libname${shared_ext}.$versuffix
$libname${shared_ext}.$major $libname${shared_ext}'
    soname_spec='$libname${shared_ext}.$major'
    shlibpath_var=LD_LIBRARY_PATH
fi
;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
    version_type=freebsd-elf
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext} $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    if test "$with_gnu_ld" = yes; then
        sys_lib_search_path_spec='/usr/local/lib /usr/gnu/lib /usr/ccs/lib
/usr/lib /lib'
    else
        sys_lib_search_path_spec='/usr/ccs/lib /usr/lib'
        case $host_os in
            sco3.2v5*)
                sys_lib_search_path_spec="$sys_lib_search_path_spec /lib"
            ;;
        esac
    fi
    sys_lib_dlsearch_path_spec='/usr/lib'
;;

tpf*)
    # TPF is a cross-target only. Preferred cross-host = GNU/Linux.
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
;;

uts4*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'

```

```
shlibpath_var=LD_LIBRARY_PATH
;;

*)
  dynamic_linker=no
  ;;
esac
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dynamic_linker" >&5
$as_echo "$dynamic_linker" >&6; }
test "$dynamic_linker" = no && can_build_shared=no

variables_saved_for_relink="PATH $shlibpath_var $runpath_var"
if test "$GCC" = yes; then
  variables_saved_for_relink="$variables_saved_for_relink
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH"
fi

if test "${lt_cv_sys_lib_search_path_spec+set}" = set; then
  sys_lib_search_path_spec="$lt_cv_sys_lib_search_path_spec"
fi
if test "${lt_cv_sys_lib_dlsearch_path_spec+set}" = set; then
  sys_lib_dlsearch_path_spec="$lt_cv_sys_lib_dlsearch_path_spec"
fi
```



```

    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking how to hardcode
library paths into programs" >&5
$sas_echo_n "checking how to hardcode library paths into programs... "
>&6; }
hardcode_action=
if test -n "$hardcode_libdir_flag_spec" ||
    test -n "$runpath_var" ||
    test "X$hardcode_automatic" = "Xyes" ; then

    # We can hardcode non-existent directories.
    if test "$hardcode_direct" != no &&
        # If the only mechanism to avoid hardcoding is shlibpath_var, we
        # have to relink, otherwise we might link with an installed
library
        # when we should be linking with a yet-to-be-installed one
        ## test "$_LT_TAGVAR(hardcode_shlibpath_var, )" != no &&
        test "$hardcode_minus_L" != no; then
        # Linking always hardcodes the temporary library directory.
        hardcode_action=relink
    else
        # We can link without hardcoding, and we can hardcode nonexisting
dirs.
        hardcode_action=immediate
    fi
else
    # We cannot hardcode anything, or else we can only hardcode existing
# directories.
    hardcode_action=unsupported
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $hardcode_action" >&5
$sas_echo "$hardcode_action" >&6; }

if test "$hardcode_action" = relink ||
    test "$inherit_rpath" = yes; then
    # Fast installation is not supported
    enable_fast_install=no
elif test "$shlibpath_overrides_runpath" = yes ||
    test "$enable_shared" = no; then
    # Fast installation is not necessary
    enable_fast_install=needless
fi

```



```

    if test "x$enable_dlopen" != xyes; then
    enable_dlopen=unknown
    enable_dlopen_self=unknown
    enable_dlopen_self_static=unknown
else
    lt_cv_dlopen=no
    lt_cv_dlopen_libs=

    case $host_os in
    beos*)
        lt_cv_dlopen="load_add_on"
        lt_cv_dlopen_libs=
        lt_cv_dlopen_self=yes
        ;;

    mingw* | pw32* | cegcc*)
        lt_cv_dlopen="LoadLibrary"
        lt_cv_dlopen_libs=
        ;;

    cygwin*)
        lt_cv_dlopen="dlopen"
        lt_cv_dlopen_libs=
        ;;

    darwin*)
        # if libdl is installed we need to link against it
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -
ldl" >&5
$as_echo_n "checking for dlopen in -ldl... " >&6; }
if ${ac_cv_lib_dl_dlopen+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-ldl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int

```

```

main ()
{
return dlopen ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_dl_dlopen=yes
else
ac_cv_lib_dl_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_lib_dl_dlopen"
>&5
$as_echo "$ac_cv_lib_dl_dlopen" >&6; }
if test "x$ac_cv_lib_dl_dlopen" = xyes; then :
lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-ldl"
else

lt_cv_dlopen="dyld"
lt_cv_dlopen_libs=
lt_cv_dlopen_self=yes

fi

;;

*)
ac_fn_c_check_func "$LINENO" "shl_load" "ac_cv_func_shl_load"
if test "x$ac_cv_func_shl_load" = xyes; then :
lt_cv_dlopen="shl_load"
else
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for shl_load in -
ldld" >&5
$as_echo_n "checking for shl_load in -ldld... " >&6; }
if ${ac_cv_lib_dld_shl_load+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-ldld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"

```

```

#endif
char shl_load ();
int
main ()
{
return shl_load ();
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_dld_shl_load=yes
else
    ac_cv_lib_dld_shl_load=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dld_shl_load" >&5
$as_echo "$ac_cv_lib_dld_shl_load" >&6; }
if test "x$ac_cv_lib_dld_shl_load" = xyes; then :
    lt_cv_dlopen="shl_load" lt_cv_dlopen_libs="-ldld"
else
    ac_fn_c_check_func "$LINENO" "dlopen" "ac_cv_func_dlopen"
if test "x$ac_cv_func_dlopen" = xyes; then :
    lt_cv_dlopen="dlopen"
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -
ldl" >&5
$as_echo_n "checking for dlopen in -ldl... " >&6; }
if ${ac_cv_lib_dl_dlopen+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-ldl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int
main ()
{
return dlopen ();
    ;
}

```

```

    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_dl_dlopen=yes
else
    ac_cv_lib_dl_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_lib_dl_dlopen"
>&5
$as_echo "$ac_cv_lib_dl_dlopen" >&6; }
if test "x$ac_cv_lib_dl_dlopen" = xyes; then :
    lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-ldl"
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -
lsvld" >&5
$as_echo_n "checking for dlopen in -lsvld... " >&6; }
if ${ac_cv_lib_svld_dlopen+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lsvld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int
main ()
{
return dlopen ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_svld_dlopen=yes
else
    ac_cv_lib_svld_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS

```

```

fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_svld_dlopen" >&5
$sas_echo "$ac_cv_lib_svld_dlopen" >&6; }
if test "x$ac_cv_lib_svld_dlopen" = xyes; then :
  lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-lsvld"
else
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for dld_link in -
ldld" >&5
$sas_echo_n "checking for dld_link in -ldld... " >&6; }
if ${ac_cv_lib_dld_dld_link+:} false; then :
  $sas_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-ldld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char dld_link ();
int
main ()
{
return dld_link ();
  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_dld_dld_link=yes
else
  ac_cv_lib_dld_dld_link=no
fi
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dld_dld_link" >&5
$sas_echo "$ac_cv_lib_dld_dld_link" >&6; }
if test "x$ac_cv_lib_dld_dld_link" = xyes; then :
  lt_cv_dlopen="dld_link" lt_cv_dlopen_libs="-ldld"
fi
fi

```

```

fi

fi

fi

fi

    ;;
esac

if test "x$lt_cv_dlopen" != xno; then
  enable_dlopen=yes
else
  enable_dlopen=no
fi

case $lt_cv_dlopen in
dlopen)
  save_CPPFLAGS="$CPPFLAGS"
  test "x$sac_cv_header_dlfcn_h" = xyes && CPPFLAGS="$CPPFLAGS -
DHAVE_DLFCN_H"

  save_LDFLAGS="$LDFLAGS"
  wl=$lt_prog_compiler_wl eval LDFLAGS="\`$LDFLAGS
$export_dynamic_flag_spec\`"

  save_LIBS="$LIBS"
  LIBS="$lt_cv_dlopen_libs $LIBS"

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether a
program can dlopen itself" >&5
$as_echo_n "checking whether a program can dlopen itself... " >&6; }
if ${lt_cv_dlopen_self+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :
    lt_cv_dlopen_self=cross
  else
    lt_dlunknown=0; lt_dlno_uscore=1; lt_dlneed_uscore=2
    lt_status=$lt_dlunknown
    cat > conftest.$sac_ext <<_LT_EOF
#line $LINENO "configure"
#include "confdefs.h"

#if HAVE_DLFCN_H
#include <dlfcn.h>
#endif

```

```

#include <stdio.h>

#ifdef RTLD_GLOBAL
# define LT_DLGLOBAL      RTLD_GLOBAL
#else
# ifdef DL_GLOBAL
#   define LT_DLGLOBAL    DL_GLOBAL
# else
#   define LT_DLGLOBAL    0
# endif
#endif

/* We may have to define LT_DLLAZY_OR_NOW in the command line if we
   find out it does not work in some platform. */
#ifndef LT_DLLAZY_OR_NOW
# ifdef RTLD_LAZY
#   define LT_DLLAZY_OR_NOW      RTLD_LAZY
# else
#   ifdef DL_LAZY
#     define LT_DLLAZY_OR_NOW    DL_LAZY
#   else
#     ifdef RTLD_NOW
#       define LT_DLLAZY_OR_NOW  RTLD_NOW
#     else
#       ifdef DL_NOW
#         define LT_DLLAZY_OR_NOW  DL_NOW
#       else
#         define LT_DLLAZY_OR_NOW  0
#       endif
#     endif
#   endif
# endif
#endif

/* When -fvisibility=hidden is used, assume the code has been annotated
   correspondingly for the symbols needed. */
#ifdef __GNUC__ && (((__GNUC__ == 3) && (__GNUC_MINOR__ >= 3))
|| (__GNUC__ > 3))
int fnord () __attribute__((visibility("default")));
#endif

int fnord () { return 42; }
int main ()
{
  void *self = dlopen (0, LT_DLGLOBAL|LT_DLLAZY_OR_NOW);
  int status = $lt_dlunknown;

  if (self)
    {
      if (dlsym (self,"fnord"))      status = $lt_dlno_uscore;
      else

```

```

        {
            if (dlsym( self, "_fnord")) status = $lt_dlneed_uscore;
            else puts (dlerror ());
        }
        /* dlclose (self); */
    }
else
    puts (dlerror ());

return status;
}
_LT_EOF
if { { eval echo "\$as_me\":"${as_lineno-$LINENO}: \"$ac_link\""; }
>&5
(eval $ac_link) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; } && test -s conftest${ac_exeext} 2>/dev/null;
then
    (./conftest; exit; ) >&5 2>/dev/null
    lt_status=$?
    case x$lt_status in
        x$lt_dlno_uscore) lt_cv_dlopen_self=yes ;;
        x$lt_dlneed_uscore) lt_cv_dlopen_self=yes ;;
        x$lt_dlunknown|x*) lt_cv_dlopen_self=no ;;
    esac
else :
    # compilation failed
    lt_cv_dlopen_self=no
fi
fi
rm -fr conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_dlopen_self"
>&5
$as_echo "$lt_cv_dlopen_self" >&6; }

if test "x$lt_cv_dlopen_self" = xyes; then
    wl=$lt_prog_compiler_wl eval LDFLAGS="\$LDFLAGS
$lt_prog_compiler_static\"
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether a
statically linked program can dlopen itself" >&5
$as_echo_n "checking whether a statically linked program can dlopen
itself... " >&6; }
if ${lt_cv_dlopen_self_static+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test "$cross_compiling" = yes; then :
        lt_cv_dlopen_self_static=cross
    else

```



```

    lt_dlunknown=0; lt_dlno_uscore=1; lt_dlneed_uscore=2
    lt_status=$lt_dlunknown
    cat > conftest.$ac_ext <<_LT_EOF
#line $LINENO "configure"
#include "confdefs.h"

#if HAVE_DLFCN_H
#include <dlfcn.h>
#endif

#include <stdio.h>

#ifdef RTLD_GLOBAL
# define LT_DLGLOBAL      RTLD_GLOBAL
#else
# ifdef DL_GLOBAL
#   define LT_DLGLOBAL    DL_GLOBAL
# else
#   define LT_DLGLOBAL    0
# endif
#endif

/* We may have to define LT_DLLAZY_OR_NOW in the command line if we
   find out it does not work in some platform. */
#ifndef LT_DLLAZY_OR_NOW
# ifdef RTLD_LAZY
#   define LT_DLLAZY_OR_NOW      RTLD_LAZY
# else
#   ifdef DL_LAZY
#     define LT_DLLAZY_OR_NOW    DL_LAZY
#   else
#     ifdef RTLD_NOW
#       define LT_DLLAZY_OR_NOW RTLD_NOW
#     else
#       ifdef DL_NOW
#         define LT_DLLAZY_OR_NOW    DL_NOW
#       else
#         define LT_DLLAZY_OR_NOW    0
#       endif
#     endif
#   endif
# endif
#endif

/* When -fvisibility=hidden is used, assume the code has been annotated
   correspondingly for the symbols needed. */
#if defined(__GNUC__) && (((__GNUC__ == 3) && (__GNUC_MINOR__ >= 3))
|| (__GNUC__ > 3))
int fnord () __attribute__((visibility("default")));
#endif

int fnord () { return 42; }

```

```

int main ()
{
  void *self = dlopen (0, LT_DLGLOBAL|LT_DLLAZY_OR_NOW);
  int status = $lt_dlunknown;

  if (self)
  {
    if (dlsym (self,"fnord"))      status = $lt_dlno_uscore;
    else
    {
      if (dlsym( self,"_fnord"))  status = $lt_dlneed_uscore;
      else puts (dlerror ());
    }
    /* dlclose (self); */
  }
  else
    puts (dlerror ());

  return status;
}
_LT_EOF
if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}: \"$ac_link\""; }
>&5
(eval $ac_link) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
test $ac_status = 0; } && test -s conftest${ac_exeext} 2>/dev/null;
then
  (./conftest; exit; ) >&5 2>/dev/null
  lt_status=$?
  case x$lt_status in
    x$lt_dlno_uscore) lt_cv_dlopen_self_static=yes ;;
    x$lt_dlneed_uscore) lt_cv_dlopen_self_static=yes ;;
    x$lt_dlunknown|x*) lt_cv_dlopen_self_static=no ;;
  esac
  else :
    # compilation failed
    lt_cv_dlopen_self_static=no
  fi
fi
rm -fr conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_dlopen_self_static" >&5
$as_echo "$lt_cv_dlopen_self_static" >&6; }
fi

CPPFLAGS="$save_CPPFLAGS"
LDFLAGS="$save_LDFLAGS"
LIBS="$save_LIBS"

```

```

    ;;
esac

case $lt_cv_dlopen_self in
yes|no) enable_dlopen_self=$lt_cv_dlopen_self ;;
*) enable_dlopen_self=unknown ;;
esac

case $lt_cv_dlopen_self_static in
yes|no) enable_dlopen_self_static=$lt_cv_dlopen_self_static ;;
*) enable_dlopen_self_static=unknown ;;
esac
fi

striplib=
old_striplib=
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether stripping
libraries is possible" >&5
$as_echo_n "checking whether stripping libraries is possible... " >&6;
}
if test -n "$STRIP" && $STRIP -V 2>&1 | $GREP "GNU strip" >/dev/null;
then
  test -z "$old_striplib" && old_striplib="$STRIP --strip-debug"
  test -z "$striplib" && striplib="$STRIP --strip-unneeded"
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
# FIXME - insert some real tests, host_os isn't really good enough
case $host_os in
darwin*)
  if test -n "$STRIP" ; then
    striplib="$STRIP -x"
    old_striplib="$STRIP -S"
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
  else

```

```

        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi
    ;;
*)
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    ;;
esac
fi

```

```

# Report which library types will actually be built
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if libtool
supports shared libraries" >&5
$as_echo_n "checking if libtool supports shared libraries... " >&6; }
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $scan_build_shared"
>&5
$as_echo "$scan_build_shared" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
shared libraries" >&5
$as_echo_n "checking whether to build shared libraries... " >&6; }
test "$scan_build_shared" = "no" && enable_shared=no

# On AIX, shared libraries and static libraries use the same
namespace, and
# are all built from PIC.
case $host_os in
aix3*)
    test "$enable_shared" = yes && enable_static=no
    if test -n "$RANLIB"; then
        archive_cmds="$archive_cmds~\${RANLIB} \$lib"
        postinstall_cmds='${RANLIB} $lib'
    fi
    ;;

aix[4-9]*)
    if test "$host_cpu" != ia64 && test "$aix_use_runtimelinking" = no
; then
        test "$enable_shared" = yes && enable_static=no
    fi

```

```

    ;;
  esac
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_shared" >&5
$as_echo "$enable_shared" >&6; }

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
static libraries" >&5
$as_echo_n "checking whether to build static libraries... " >&6; }
  # Make sure either enable_shared or enable_static is yes.
  test "$enable_shared" = yes || enable_static=yes
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_static" >&5
$as_echo "$enable_static" >&6; }

```

```

fi
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

CC="$lt_save_CC"

```

```

ac_config_commands="$ac_config_commands libtool"

```

```

# Only expand once:

```

```

# compress spaces in flags
CFLAGS=`echo "$CFLAGS" | sed -e 's/ +/ /g'`
CPPFLAGS=`echo "$CPPFLAGS" | sed -e 's/ +/ /g'`

```

```

if test x$enable_gcov = xyes; then
    # so that config.h changes when you toggle gcov support

cat >>confdefs.h <<_ACEOF
#define DBUS_GCOV_ENABLED __GNUC__ * 10000 + __GNUC_MINOR__ * 100 +
__GNUC_PATCHLEVEL__
_ACEOF

fi

#### Various functions
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for library
containing socket" >&5
$as_echo_n "checking for library containing socket... " >&6; }
if ${ac_cv_search_socket+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_func_search_save_LIBS=$LIBS
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char socket ();
int
main ()
{
return socket ();
;
return 0;
}
_ACEOF
for ac_lib in '' socket; do
if test -z "$ac_lib"; then
ac_res="none required"
else
ac_res=-l$ac_lib
LIBS="-l$ac_lib $ac_func_search_save_LIBS"
fi
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_search_socket=$ac_res
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext
if ${ac_cv_search_socket+:} false; then :
break
fi

```

```

done
if ${ac_cv_search_socket+:} false; then :

else
    ac_cv_search_socket=no
fi
rm conftest.$ac_ext
LIBS=$ac_func_search_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_search_socket"
>&5
$as_echo "$ac_cv_search_socket" >&6; }
ac_res=$ac_cv_search_socket
if test "$ac_res" != no; then :
    test "$ac_res" = "none required" || LIBS="$ac_res $LIBS"

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether socklen_t is
defined" >&5
$as_echo_n "checking whether socklen_t is defined... " >&6; }
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>

int
main ()
{

socklen_t foo;
foo = 1;

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    dbus_have_socklen_t=yes
else
    dbus_have_socklen_t=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_have_socklen_t"
>&5
$as_echo "$dbus_have_socklen_t" >&6; }

if test "x$dbus_have_socklen_t" = "xyes"; then

```

```

$as_echo "#define HAVE_SOCKLEN_T 1" >>confdefs.h

fi

#### Abstract sockets

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking abstract socket
namespace" >&5
$as_echo_n "checking abstract socket namespace... " >&6; }
if ${ac_cv_have_abstract_sockets+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`$ac_pwd':"
>&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error $? "cannot run test program while cross compiling
See \`config.log' for more details" "$LINENO" 5; }
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <errno.h>

int
main ()
{

  int listen_fd;
  struct sockaddr_un addr;

  listen_fd = socket (PF_UNIX, SOCK_STREAM, 0);

  if (listen_fd < 0)
    {
      fprintf (stderr, "socket() failed: %s\n", strerror (errno));
      exit (1);
    }
}

```



```

memset (&addr, '\0', sizeof (addr));
addr.sun_family = AF_UNIX;
strcpy (addr.sun_path, "X/tmp/dbus-fake-socket-path-used-in-
configure-test");
addr.sun_path[0] = '\0'; /* this is what makes it abstract */

if (bind (listen_fd, (struct sockaddr*) &addr, SUN_LEN (&addr)) < 0)
{
    fprintf (stderr, "Abstract socket namespace bind() failed:
%s\n",
            strerror (errno));
    exit (1);
}
else
    exit (0);

;
return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    ac_cv_have_abstract_sockets=yes
else
    ac_cv_have_abstract_sockets=no

fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$ac_exeext \
    conftest.$ac_objext conftest.beam conftest.$ac_ext
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_have_abstract_sockets" >&5
$as_echo "$ac_cv_have_abstract_sockets" >&6; }
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

if test x$enable_abstract_sockets = xyes; then
    if test x$ac_cv_have_abstract_sockets = xno; then
        as_fn_error $? "Abstract sockets explicitly required, and support
not detected." "$LINENO" 5
    fi
fi

if test x$enable_abstract_sockets = xno; then
    ac_cv_have_abstract_sockets=no;
fi

```

```

if test x$ac_cv_have_abstract_sockets = xyes ; then
    DBUS_PATH_OR_ABSTRACT=abstract

$as_echo "#define HAVE_ABSTRACT_SOCKETS 1" >>confdefs.h

else
    DBUS_PATH_OR_ABSTRACT=path
fi

# this is used in addresses to prefer abstract, e.g.
# unix:path=/foo or unix:abstract=/foo

#### Sort out XML library

# see what we have
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for
XML_ParserCreate_MM in -lexpat" >&5
$as_echo_n "checking for XML_ParserCreate_MM in -lexpat... " >&6; }
if ${ac_cv_lib_expat_XML_ParserCreate_MM+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lexpat $LIBS"
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char XML_ParserCreate_MM ();
int
main ()
{
return XML_ParserCreate_MM ();
    ;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_expat_XML_ParserCreate_MM=yes
else
    ac_cv_lib_expat_XML_ParserCreate_MM=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_expat_XML_ParserCreate_MM" >&5
$as_echo "$ac_cv_lib_expat_XML_ParserCreate_MM" >&6; }
if test "x$ac_cv_lib_expat_XML_ParserCreate_MM" = xyes; then :
  for ac_header in expat.h
do :
  ac_fn_c_check_header_mongrel "$LINENO" "expat.h"
"ac_cv_header_expat_h" "$ac_includes_default"
if test "x$ac_cv_header_expat_h" = xyes; then :
  cat >>confdefs.h <<_ACEOF
#define HAVE_EXPAT_H 1
_ACEOF
  have_expat=true
else
  have_expat=false
fi

done

else
  have_expat=false
fi

if ! $have_expat ; then
  as_fn_error $? "expat library not found, check config.log for failed
attempts" "$LINENO" 5
fi

XML_LIBS=-lexpat
XML_CFLAGS=

#### Set up final flags

if test "x$ac_cv_env_PKG_CONFIG_set" != "xset"; then
  if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}pkg-config", so it can
    be a program name with args.
    set dummy ${ac_tool_prefix}pkg-config; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_PKG_CONFIG+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $PKG_CONFIG in

```

```

    [\\/* | ?:[\\/*]*)
    ac_cv_path_PKG_CONFIG="$PKG_CONFIG" # Let the user override the test
with a path.
    ;;
*)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_path_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
    done
IFS=$as_save_IFS

    ;;
esac
fi
PKG_CONFIG=$ac_cv_path_PKG_CONFIG
if test -n "$PKG_CONFIG"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $PKG_CONFIG" >&5
$as_echo "$PKG_CONFIG" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_path_PKG_CONFIG"; then
    ac_pt_PKG_CONFIG=$PKG_CONFIG
    # Extract the first word of "pkg-config", so it can be a program
name with args.
    set dummy pkg-config; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_path_ac_pt_PKG_CONFIG+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        case $ac_pt_PKG_CONFIG in
        [\\/* | ?:[\\/*]*)
            ac_cv_path_ac_pt_PKG_CONFIG="$ac_pt_PKG_CONFIG" # Let the user
override the test with a path.
            ;;
*)
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR

```

```

for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_ac_pt_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

;;
esac
fi
ac_pt_PKG_CONFIG=$ac_cv_path_ac_pt_PKG_CONFIG
if test -n "$ac_pt_PKG_CONFIG"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_pt_PKG_CONFIG"
>&5
$as_echo "$ac_pt_PKG_CONFIG" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

if test "x$ac_pt_PKG_CONFIG" = x; then
  PKG_CONFIG=""
else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  PKG_CONFIG=$ac_pt_PKG_CONFIG
fi
else
  PKG_CONFIG="$ac_cv_path_PKG_CONFIG"
fi

fi
if test -n "$PKG_CONFIG"; then
  _pkg_min_version=0.9.0
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking pkg-config is
at least version $_pkg_min_version" >&5
$as_echo_n "checking pkg-config is at least version
$_pkg_min_version... " >&6; }

```

```

        if $PKG_CONFIG --atleast-pkgconfig-version $_pkg_min_version;
then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
        else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        PKG_CONFIG=""
        fi
fi

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for DBUS" >&5
$as_echo_n "checking for DBUS... " >&6; }

if test -n "$DBUS_CFLAGS"; then
    pkg_cv_DBUS_CFLAGS="$DBUS_CFLAGS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \ $PKG_CONFIG --exists -
-print-errors \"dbus-1 >= 1.2.16\""; } >&5
        ($PKG_CONFIG --exists --print-errors "dbus-1 >= 1.2.16") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_DBUS_CFLAGS=`$PKG_CONFIG --cflags "dbus-1 >= 1.2.16"
2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi
if test -n "$DBUS_LIBS"; then
    pkg_cv_DBUS_LIBS="$DBUS_LIBS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \ $PKG_CONFIG --exists -
-print-errors \"dbus-1 >= 1.2.16\""; } >&5
        ($PKG_CONFIG --exists --print-errors "dbus-1 >= 1.2.16") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_DBUS_LIBS=`$PKG_CONFIG --libs "dbus-1 >= 1.2.16" 2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi

```

```

if test $pkg_failed = yes; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        DBUS_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "dbus-1 >= 1.2.16" 2>&1`
    else
        DBUS_PKG_ERRORS=`$PKG_CONFIG --print-errors "dbus-1 >=
1.2.16" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$DBUS_PKG_ERRORS" >&5

    as_fn_error $? "Package requirements (dbus-1 >= 1.2.16) were not
met:

$DBUS_PKG_ERRORS

```

Consider adjusting the PKG_CONFIG_PATH environment variable if you installed software in a non-standard prefix.

Alternatively, you may set the environment variables DBUS_CFLAGS and DBUS_LIBS to avoid the need to call pkg-config.

See the pkg-config man page for more details." "\$LINENO" 5

```

elif test $pkg_failed = untried; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
    as_fn_error $? "The pkg-config script could not be found or is too
old. Make sure it
is in your PATH or set the PKG_CONFIG environment variable to the full
path to pkg-config.

```

Alternatively, you may set the environment variables DBUS_CFLAGS and DBUS_LIBS to avoid the need to call pkg-config.

See the pkg-config man page for more details.

To get pkg-config, see <<http://pkg-config.freedesktop.org/>>.

See `config.log' for more details" "\$LINENO" 5; }

```

else
    DBUS_CFLAGS=$pkg_cv_DBUS_CFLAGS
    DBUS_LIBS=$pkg_cv_DBUS_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5

```

```

$as_echo "yes" >&6; }

fi

# Glib detection

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for DBUS_GLIB" >&5
$as_echo_n "checking for DBUS_GLIB... " >&6; }

if test -n "$DBUS_GLIB_CFLAGS"; then
  pkg_cv_DBUS_GLIB_CFLAGS="$DBUS_GLIB_CFLAGS"
  elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
      { { $as_echo "$as_me:${as_lineno-$LINENO}: \$PKG_CONFIG --exists -
--print-errors \"gobject-2.0 >= 2.26, gio-2.0 >= 2.26\""; } >&5
      ($PKG_CONFIG --exists --print-errors "gobject-2.0 >= 2.26, gio-2.0
>= 2.26") 2>&5
      ac_status=$?
      $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
      test $ac_status = 0; }; then
      pkg_cv_DBUS_GLIB_CFLAGS=`$PKG_CONFIG --cflags "gobject-2.0 >= 2.26,
gio-2.0 >= 2.26" 2>/dev/null`
    else
      pkg_failed=yes
    fi
  else
    pkg_failed=untried
  fi
if test -n "$DBUS_GLIB_LIBS"; then
  pkg_cv_DBUS_GLIB_LIBS="$DBUS_GLIB_LIBS"
  elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
      { { $as_echo "$as_me:${as_lineno-$LINENO}: \$PKG_CONFIG --exists -
--print-errors \"gobject-2.0 >= 2.26, gio-2.0 >= 2.26\""; } >&5
      ($PKG_CONFIG --exists --print-errors "gobject-2.0 >= 2.26, gio-2.0
>= 2.26") 2>&5
      ac_status=$?
      $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
      test $ac_status = 0; }; then
      pkg_cv_DBUS_GLIB_LIBS=`$PKG_CONFIG --libs "gobject-2.0 >= 2.26, gio-
2.0 >= 2.26" 2>/dev/null`
    else
      pkg_failed=yes
    fi
  else
    pkg_failed=untried
  fi
fi

```



```

if test $pkg_failed = yes; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        DBUS_GLIB_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "gobject-2.0 >= 2.26, gio-2.0 >= 2.26" 2>&1`
    else
        DBUS_GLIB_PKG_ERRORS=`$PKG_CONFIG --print-errors
"gobject-2.0 >= 2.26, gio-2.0 >= 2.26" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$DBUS_GLIB_PKG_ERRORS" >&5

    as_fn_error $? "Package requirements (gobject-2.0 >= 2.26, gio-
2.0 >= 2.26) were not met:

$DBUS_GLIB_PKG_ERRORS

```

Consider adjusting the PKG_CONFIG_PATH environment variable if you installed software in a non-standard prefix.

Alternatively, you may set the environment variables DBUS_GLIB_CFLAGS and DBUS_GLIB_LIBS to avoid the need to call pkg-config.

See the pkg-config man page for more details." "\$LINENO" 5

```

elif test $pkg_failed = untried; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error $? "The pkg-config script could not be found or is too
old. Make sure it
is in your PATH or set the PKG_CONFIG environment variable to the full
path to pkg-config.

```

Alternatively, you may set the environment variables DBUS_GLIB_CFLAGS and DBUS_GLIB_LIBS to avoid the need to call pkg-config.

See the pkg-config man page for more details.

To get pkg-config, see <<http://pkg-config.freedesktop.org/>>.

See `config.log' for more details" "\$LINENO" 5; }

```

else
    DBUS_GLIB_CFLAGS=$pkg_cv_DBUS_GLIB_CFLAGS
    DBUS_GLIB_LIBS=$pkg_cv_DBUS_GLIB_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5

```

```

$as_echo "yes" >&6; }

fi

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for
DBUS_GLIB_THREADS" >&5
$as_echo_n "checking for DBUS_GLIB_THREADS... " >&6; }

if test -n "$DBUS_GLIB_THREADS_CFLAGS"; then
  pkg_cv_DBUS_GLIB_THREADS_CFLAGS="$DBUS_GLIB_THREADS_CFLAGS"
  elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
      { { $as_echo "$as_me:${as_lineno-$LINENO}: \$PKG_CONFIG --exists -
-print-errors \"gthread-2.0 >= 2.6\""; } >&5
      ($PKG_CONFIG --exists --print-errors "gthread-2.0 >= 2.6") 2>&5
      ac_status=$?
      $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
      test $ac_status = 0; }; then
      pkg_cv_DBUS_GLIB_THREADS_CFLAGS=`$PKG_CONFIG --cflags "gthread-2.0
>= 2.6" 2>/dev/null`
    else
      pkg_failed=yes
    fi
  else
    pkg_failed=untried
  fi
fi
if test -n "$DBUS_GLIB_THREADS_LIBS"; then
  pkg_cv_DBUS_GLIB_THREADS_LIBS="$DBUS_GLIB_THREADS_LIBS"
  elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
      { { $as_echo "$as_me:${as_lineno-$LINENO}: \$PKG_CONFIG --exists -
-print-errors \"gthread-2.0 >= 2.6\""; } >&5
      ($PKG_CONFIG --exists --print-errors "gthread-2.0 >= 2.6") 2>&5
      ac_status=$?
      $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
      test $ac_status = 0; }; then
      pkg_cv_DBUS_GLIB_THREADS_LIBS=`$PKG_CONFIG --libs "gthread-2.0 >=
2.6" 2>/dev/null`
    else
      pkg_failed=yes
    fi
  else
    pkg_failed=untried
  fi
fi

if test $pkg_failed = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

```

```

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        DBUS_GLIB_THREADS_PKG_ERRORS=`$PKG_CONFIG --short-errors
--print-errors "gthread-2.0 >= 2.6" 2>&1`
    else
        DBUS_GLIB_THREADS_PKG_ERRORS=`$PKG_CONFIG --print-errors
"gthread-2.0 >= 2.6" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$DBUS_GLIB_THREADS_PKG_ERRORS" >&5

    have_glib_threads=no
elif test $pkg_failed = untried; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    have_glib_threads=no
else
    DBUS_GLIB_THREADS_CFLAGS=$pkg_cv_DBUS_GLIB_THREADS_CFLAGS
    DBUS_GLIB_THREADS_LIBS=$pkg_cv_DBUS_GLIB_THREADS_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    have_glib_threads=yes
fi

if test x$have_glib_threads = xyes; then
    HAVE_GLIB_THREADS_TRUE=
    HAVE_GLIB_THREADS_FALSE='#'
else
    HAVE_GLIB_THREADS_TRUE='#'
    HAVE_GLIB_THREADS_FALSE=
fi

GLIB_GENMARSHAL=`$PKG_CONFIG --variable=glib_genmarshal glib-2.0`

DBUS_GLIB_TOOL_CFLAGS=$XML_CFLAGS
DBUS_GLIB_TOOL_LIBS="$XML_LIBS"

### gtk-doc Documentation

```

```

    # Extract the first word of "gtkdoc-check", so it can be a program
name with args.
set dummy gtkdoc-check; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_GTKDOC_CHECK+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $GTKDOC_CHECK in
    [\\/* | ?:[\\/*]*)
      ac_cv_path_GTKDOC_CHECK="$GTKDOC_CHECK" # Let the user override the
test with a path.
      ;;
    *)
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_GTKDOC_CHECK="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

    ;;
  esac
fi
GTKDOC_CHECK=$ac_cv_path_GTKDOC_CHECK
if test -n "$GTKDOC_CHECK"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $GTKDOC_CHECK" >&5
$as_echo "$GTKDOC_CHECK" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

for ac_prog in gtkdoc-rebase
do
  # Extract the first word of "$ac_prog", so it can be a program name
with args.
set dummy $ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_GTKDOC_REBASE+:} false; then :

```

```

    $as_echo_n "(cached) " >&6
else
  case $GTKDOC_REBASE in
    [\\/* | ?:[\\/*]*)
      ac_cv_path GTKDOC_REBASE="$GTKDOC_REBASE" # Let the user override
the test with a path.
      ;;
    *)
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path GTKDOC_REBASE="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
  done
IFS=$as_save_IFS

  ;;
esac
fi
GTKDOC_REBASE=$ac_cv_path GTKDOC_REBASE
if test -n "$GTKDOC_REBASE"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $GTKDOC_REBASE" >&5
$as_echo "$GTKDOC_REBASE" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  test -n "$GTKDOC_REBASE" && break
done
test -n "$GTKDOC_REBASE" || GTKDOC_REBASE="true"

# Extract the first word of "gtkdoc-mkpdf", so it can be a program
name with args.
set dummy gtkdoc-mkpdf; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path GTKDOC_MKPDF+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $GTKDOC_MKPDF in
    [\\/* | ?:[\\/*]*)

```

```
    ac_cv_path_GTKDOC_MKPDF="$GTKDOC_MKPDF" # Let the user override the
test with a path.
```

```
;;
*)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_path_GTKDOC_MKPDF="$as_dir/$ac_word$ac_exec_ext"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

;;
esac
fi
GTKDOC_MKPDF=$ac_cv_path_GTKDOC_MKPDF
if test -n "$GTKDOC_MKPDF"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $GTKDOC_MKPDF" >&5
$as_echo "$GTKDOC_MKPDF" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi
```

```
# Check whether --with-html-dir was given.
if test "${with_html_dir+set}" = set; then :
    withval=$with_html_dir;
else
    with_html_dir='${datadir}/gtk-doc/html'
fi
```

```
HTML_DIR="$with_html_dir"
```

```
# Check whether --enable-gtk-doc was given.
if test "${enable_gtk_doc+set}" = set; then :
    enableval=$enable_gtk_doc;
else
    enable_gtk_doc=no
fi
```

```

if test x$enable_gtk_doc = xyes; then
  if test -n "$PKG_CONFIG" && \
    { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"gtk-doc >= 1.4\""; } >&5
    ($PKG_CONFIG --exists --print-errors "gtk-doc >= 1.4") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    :
else
  as_fn_error $? "You need to have gtk-doc >= 1.4 installed to build
$PACKAGE_NAME" "$LINENO" 5
fi
    if test "x$PACKAGE_NAME" != "xglib"; then

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for GTKDOC_DEPS" >&5
$as_echo_n "checking for GTKDOC_DEPS... " >&6; }

if test -n "$GTKDOC_DEPS_CFLAGS"; then
  pkg_cv_GTKDOC_DEPS_CFLAGS="$GTKDOC_DEPS_CFLAGS"
elif test -n "$PKG_CONFIG"; then
  if test -n "$PKG_CONFIG" && \
    { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"glib-2.0 >= 2.10.0 gobject-2.0 >= 2.10.0\""; } >&5
    ($PKG_CONFIG --exists --print-errors "glib-2.0 >= 2.10.0 gobject-2.0
>= 2.10.0") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    pkg_cv_GTKDOC_DEPS_CFLAGS=`$PKG_CONFIG --cflags "glib-2.0 >= 2.10.0
gobject-2.0 >= 2.10.0" 2>/dev/null`
  else
    pkg_failed=yes
  fi
else
  pkg_failed=untried
fi
if test -n "$GTKDOC_DEPS_LIBS"; then
  pkg_cv_GTKDOC_DEPS_LIBS="$GTKDOC_DEPS_LIBS"
elif test -n "$PKG_CONFIG"; then
  if test -n "$PKG_CONFIG" && \
    { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"glib-2.0 >= 2.10.0 gobject-2.0 >= 2.10.0\""; } >&5
    ($PKG_CONFIG --exists --print-errors "glib-2.0 >= 2.10.0 gobject-2.0
>= 2.10.0") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    pkg_cv_GTKDOC_DEPS_LIBS=`$PKG_CONFIG --libs "glib-2.0 >= 2.10.0
gobject-2.0 >= 2.10.0" 2>/dev/null`

```

```

else
  pkg_failed=yes
fi
else
  pkg_failed=untried
fi

if test $pkg_failed = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
  _pkg_short_errors_supported=yes
else
  _pkg_short_errors_supported=no
fi
  if test $_pkg_short_errors_supported = yes; then
    GTKDOC_DEPS_PKG_ERRORS=`$PKG_CONFIG --short-errors --
print-errors "glib-2.0 >= 2.10.0 gobject-2.0 >= 2.10.0" 2>&1`
  else
    GTKDOC_DEPS_PKG_ERRORS=`$PKG_CONFIG --print-errors "glib-
2.0 >= 2.10.0 gobject-2.0 >= 2.10.0" 2>&1`
  fi
  # Put the nasty error message in config.log where it belongs
  echo "$GTKDOC_DEPS_PKG_ERRORS" >&5

  as_fn_error $? "Package requirements (glib-2.0 >= 2.10.0 gobject-
2.0 >= 2.10.0) were not met:

$GTKDOC_DEPS_PKG_ERRORS

Consider adjusting the PKG_CONFIG_PATH environment variable if you
installed software in a non-standard prefix.

Alternatively, you may set the environment variables
GTKDOC_DEPS_CFLAGS
and GTKDOC_DEPS_LIBS to avoid the need to call pkg-config.
See the pkg-config man page for more details." "$LINENO" 5
elif test $pkg_failed = untried; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error $? "The pkg-config script could not be found or is too
old. Make sure it
is in your PATH or set the PKG_CONFIG environment variable to the full
path to pkg-config.

```


Alternatively, you may set the environment variables
GTKDOC_DEPS_CFLAGS
and GTKDOC_DEPS_LIBS to avoid the need to call pkg-config.
See the pkg-config man page for more details.

To get pkg-config, see <<http://pkg-config.freedesktop.org/>>.

```
See `config.log' for more details" "$LINENO" 5; }
else
    GTKDOC_DEPS_CFLAGS=$pkg_cv GTKDOC_DEPS_CFLAGS
    GTKDOC_DEPS_LIBS=$pkg_cv GTKDOC_DEPS_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }

fi
    fi
    fi

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
gtk-doc documentation" >&5
$as_echo_n "checking whether to build gtk-doc documentation... " >&6;
}
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_gtk_doc"
>&5
$as_echo "$enable_gtk_doc" >&6; }

    # Check whether --enable-gtk-doc-html was given.
if test "${enable_gtk_doc_html+set}" = set; then :
    enableval=$enable_gtk_doc_html;
else
    enable_gtk_doc_html=yes
fi

    # Check whether --enable-gtk-doc-pdf was given.
if test "${enable_gtk_doc_pdf+set}" = set; then :
    enableval=$enable_gtk_doc_pdf;
else
    enable_gtk_doc_pdf=no
fi

if test -z "$GTKDOC_MKPDF"; then
    enable_gtk_doc_pdf=no
fi

    if test x$enable_gtk_doc = xyes; then
ENABLE_GTK_DOC_TRUE=
ENABLE_GTK_DOC_FALSE='#'
else
ENABLE_GTK_DOC_TRUE='#'
ENABLE_GTK_DOC_FALSE=
fi
```

```

    if test x$enable_gtk_doc_html = xyes; then
GTK_DOC_BUILD_HTML_TRUE=
GTK_DOC_BUILD_HTML_FALSE='#'
else
GTK_DOC_BUILD_HTML_TRUE='#'
GTK_DOC_BUILD_HTML_FALSE=
fi

    if test x$enable_gtk_doc_pdf = xyes; then
GTK_DOC_BUILD_PDF_TRUE=
GTK_DOC_BUILD_PDF_FALSE='#'
else
GTK_DOC_BUILD_PDF_TRUE='#'
GTK_DOC_BUILD_PDF_FALSE=
fi

    if test -n "$LIBTOOL"; then
GTK_DOC_USE_LIBTOOL_TRUE=
GTK_DOC_USE_LIBTOOL_FALSE='#'
else
GTK_DOC_USE_LIBTOOL_TRUE='#'
GTK_DOC_USE_LIBTOOL_FALSE=
fi

    if test -n "$GTKDOC_REBASE"; then
GTK_DOC_USE_REBASE_TRUE=
GTK_DOC_USE_REBASE_FALSE='#'
else
GTK_DOC_USE_REBASE_TRUE='#'
GTK_DOC_USE_REBASE_FALSE=
fi

#### Have to go $localstatedir->$prefix/var->/usr/local/var
#### someone please fix this a better way...

#### find the actual value for $prefix that we'll end up with
## (I know this is broken and should be done in the Makefile, but
## that's a major pain and almost nobody actually seems to care)
REAL_PREFIX=
if test "x$prefix" = "xNONE"; then
REAL_PREFIX=$ac_default_prefix
else
REAL_PREFIX=$prefix
fi

## temporarily change prefix and exec_prefix
old_prefix=$prefix
prefix=$REAL_PREFIX

```

```
if test "x$exec_prefix" = xNONE ; then
    REAL_EXEC_PREFIX=$REAL_PREFIX
else
    REAL_EXEC_PREFIX=$exec_prefix
fi
old_exec_prefix=$exec_prefix
exec_prefix=$REAL_EXEC_PREFIX

## eval everything
LOCALSTATEDIR_TMP="$localstatedir"
EXPANDED_LOCALSTATEDIR=`eval echo $LOCALSTATEDIR_TMP`

SYSCONFDIR_TMP="$sysconfdir"
EXPANDED_SYSCONFDIR=`eval echo $SYSCONFDIR_TMP`

BINDIR_TMP="$bindir"
EXPANDED_BINDIR=`eval echo $BINDIR_TMP`

LIBDIR_TMP="$libdir"
EXPANDED_LIBDIR=`eval echo $LIBDIR_TMP`

DATADIR_TMP="$datadir"
EXPANDED_DATADIR=`eval echo $DATADIR_TMP`

## put prefix and exec_prefix back
prefix=$old_prefix
exec_prefix=$old_exec_prefix

#### Tell tests where to find certain stuff in builddir
ABSOLUTE_TOP_BUILDDIR=`cd ${ac_top_builddir}. && pwd`

TEST_SERVICE_DIR=${ABSOLUTE_TOP_BUILDDIR}/test/data/valid-service-
files

cat >>confdefs.h <<_ACEOF
#define TEST_SERVICE_DIR "$TEST_SERVICE_DIR"
_ACEOF

TEST_SERVICE_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/test-service

cat >>confdefs.h <<_ACEOF
```

```
#define TEST_SERVICE_BINARY "$TEST_SERVICE_BINARY"  
_ACEOF
```

```
TEST_SHELL_SERVICE_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/test-shell-  
service
```

```
cat >>confdefs.h <<_ACEOF  
#define TEST_SHELL_SERVICE_BINARY "$TEST_SHELL_SERVICE_BINARY"  
_ACEOF
```

```
TEST_CORE_SERVICE_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/core/test-  
service-glib
```

```
cat >>confdefs.h <<_ACEOF  
#define TEST_CORE_SERVICE_BINARY "$TEST_CORE_SERVICE_BINARY"  
_ACEOF
```

```
TEST_INTERFACES_SERVICE_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/interface  
s/test-service
```

```
cat >>confdefs.h <<_ACEOF  
#define TEST_INTERFACES_SERVICE_BINARY  
"$TEST_INTERFACES_SERVICE_BINARY"  
_ACEOF
```

```
TEST_EXIT_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/test-exit
```

```
cat >>confdefs.h <<_ACEOF  
#define TEST_EXIT_BINARY "$TEST_EXIT_BINARY"  
_ACEOF
```

```
TEST_SEGFAULT_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/test-segfault
```

```
cat >>confdefs.h <<_ACEOF  
#define TEST_SEGFAULT_BINARY "$TEST_SEGFAULT_BINARY"  
_ACEOF
```

```
TEST_SLEEP_FOREVER_BINARY=${ABSOLUTE_TOP_BUILDDIR}/test/test-sleep-forever
```

```
cat >>confdefs.h <<_ACEOF
#define TEST_SLEEP_FOREVER_BINARY "$TEST_SLEEP_FOREVER_BINARY"
_ACEOF
```

```
if ! test -z "$with_test_socket_dir" ; then
    TEST_SOCKET_DIR="$with_test_socket_dir"
else
    TEST_SOCKET_DIR=$DEFAULT_SOCKET_DIR
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_TEST_SOCKET_DIR "$TEST_SOCKET_DIR"
_ACEOF
```

```
ac_config_files="$ac_config_files Makefile m4/Makefile doc/Makefile
doc/reference/Makefile doc/reference/version.xml dbus/Makefile
dbus/examples/Makefile dbus/examples/statemachine/Makefile
test/Makefile test/core/Makefile test/interfaces/Makefile
test/data/valid-service-files/debug-glib.service test/data/valid-
service-files/debug-echo.service test/data/valid-service-
files/interfaces-test.service test/lib/Makefile test/manual/Makefile
tools/Makefile dbus-glib-1.pc dbus-glib-1-uninstalled.pc"
```

```
cat >confcache <<\_ACEOF
# This file is a shell script that caches the results of configure
# tests run on this system so they can be shared between configure
# scripts and configure runs, see configure's option --config-cache.
# It is not useful on other systems.  If it contains results you don't
# want to keep, you may remove or edit it.
#
# config.status only pays attention to the cache file if you give it
# the --recheck option to rerun configure.
#
# `ac_cv_env_foo' variables (set or unset) will be overridden when
# loading this file, other *unset* `ac_cv_foo' will be assigned the
# following values.
```

```
_ACEOF
```

```

# The following way of writing the cache mishandles newlines in
values,
# but we know of no workaround that is simple, portable, and
efficient.
# So, we kill variables containing newlines.
# Ultrix sh set writes to stderr and can't be redirected directly,
# and sets the high bit in the cache file unless we assign to the
vars.
(
  for ac_var in `(set) 2>&1 | sed -n 's/^\([a-zA-Z_][a-zA-Z0-
9_]*\)=.*\/\1/p'`; do
    eval ac_val=\$$ac_var
    case $ac_val in #(
      *${as_nl}*)
        case $ac_var in #(
          *_cv_*) { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: cache
variable $ac_var contains a newline" >&5
$as_echo "$as_me: WARNING: cache variable $ac_var contains a newline"
>&2;} ;;
          esac
          case $ac_var in #(
            _ | IFS | as_nl) ;; #(
            BASH_ARGV | BASH_SOURCE) eval $ac_var= ;; #(
            *) { eval $ac_var=; unset $ac_var;} ;;
          esac ;;
        esac
      done

      (set) 2>&1 |
        case $as_nl `(ac_space=' '; set) 2>&1` in #(
          *${as_nl}ac_space=\ *)
            # `set' does not quote correctly, so add quotes: double-quote
            # substitution turns \\ \\ into \, and sed turns \ into \.
            sed -n \
              "s/'/'\\\\"'/g;

s/^\(\([_$_as_cr_alnum]*_cv_[_$_as_cr_alnum]*\)\)=\(\.*\)\/\1='\'2'/p"
            ;; #(
          *)
            # `set' quotes correctly as required by POSIX, so do not add
            quotes.
            sed -n "/^\([_$_as_cr_alnum]*_cv_[_$_as_cr_alnum]*\)=/p"
              ;;
          esac |
          sort
        ) |
        sed '
/^ac_cv_env_/b end
t clear
:clear
s/^\([^\=]*\)=\(\.*\{\}\.*\)$/test "${\1+set}" = set || &/
t end

```

```

s/^\([^=]*\)=(.*)\$/\1=${\1=\2}/
:end' >>confcache
if diff "$cache_file" confcache >/dev/null 2>&1; then ;; else
  if test -w "$cache_file"; then
    if test "x$cache_file" != "x/dev/null"; then
      { $as_echo "$as_me:${as_lineno-$LINENO}: updating cache
$cache_file" >&5
$as_echo "$as_me: updating cache $cache_file" >&6;}
      if test ! -f "$cache_file" || test -h "$cache_file"; then
        cat confcache >"$cache_file"
      else
        case $cache_file in #(
          */* | ?:* )
            mv -f confcache "$cache_file"$$ &&
            mv -f "$cache_file"$$ "$cache_file" ;; #(
          *)
            mv -f confcache "$cache_file" ;;
        esac
      fi
    fi
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: not updating unwritable
cache $cache_file" >&5
$as_echo "$as_me: not updating unwritable cache $cache_file" >&6;}
  fi
fi
rm -f confcache

test "x$prefix" = xNONE && prefix=$ac_default_prefix
# Let make expand exec_prefix.
test "x$exec_prefix" = xNONE && exec_prefix='${prefix}'

DEFS=-DHAVE_CONFIG_H

ac_libobjs=
ac_ltlibobjs=
U=
for ac_i in : $LIBOBJS; do test "x$ac_i" = x: && continue
# 1. Remove the extension, and $U if already installed.
ac_script='s/\$U\././;s/\.o$//;s/\.obj$//'
ac_i=`$as_echo "$ac_i" | sed "$ac_script"`
# 2. Prepend LIBOBJDIR. When used with automake>=1.10 LIBOBJDIR
# will be set to the directory where LIBOBJS objects are built.
as_fn_append ac_libobjs " \${LIBOBJDIR}$ac_i\$U.$ac_objext"
as_fn_append ac_ltlibobjs " \${LIBOBJDIR}$ac_i"'\$U.lo'
done
LIBOBJS=$ac_libobjs

LTLIBOBJS=$ac_ltlibobjs

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking that generated files
are newer than configure" >&5
$as_echo_n "checking that generated files are newer than configure...
" >&6; }
  if test -n "$am_sleep_pid"; then
    # Hide warnings about reused PIDs.
    wait $am_sleep_pid 2>/dev/null
  fi
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: done" >&5
$as_echo "done" >&6; }
  if test -n "$EXEEXT"; then
    am__EXEEXT_TRUE=
    am__EXEEXT_FALSE='#'
  else
    am__EXEEXT_TRUE='#'
    am__EXEEXT_FALSE=
  fi

if test -z "${MAINTAINER_MODE_TRUE}" && test -z
"${MAINTAINER_MODE_FALSE}"; then
  as_fn_error $? "conditional \"MAINTAINER_MODE\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${AMDEP_TRUE}" && test -z "${AMDEP_FALSE}"; then
  as_fn_error $? "conditional \"AMDEP\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${am__fastdepCC_TRUE}" && test -z
"${am__fastdepCC_FALSE}"; then
  as_fn_error $? "conditional \"am__fastdepCC\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BASH_COMPLETION_TRUE}" && test -z
"${DBUS_BASH_COMPLETION_FALSE}"; then
  as_fn_error $? "conditional \"DBUS_BASH_COMPLETION\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BUILD_TESTS_TRUE}" && test -z
"${DBUS_BUILD_TESTS_FALSE}"; then
  as_fn_error $? "conditional \"DBUS_BUILD_TESTS\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${HAVE_GLIB_THREADS_TRUE}" && test -z
"${HAVE_GLIB_THREADS_FALSE}"; then
  as_fn_error $? "conditional \"HAVE_GLIB_THREADS\" was never defined.

```



```

Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${ENABLE_GTK_DOC_TRUE}" && test -z
"${ENABLE_GTK_DOC_FALSE}"; then
    as_fn_error $? "conditional \"ENABLE_GTK_DOC\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${GTK_DOC_BUILD_HTML_TRUE}" && test -z
"${GTK_DOC_BUILD_HTML_FALSE}"; then
    as_fn_error $? "conditional \"GTK_DOC_BUILD_HTML\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${GTK_DOC_BUILD_PDF_TRUE}" && test -z
"${GTK_DOC_BUILD_PDF_FALSE}"; then
    as_fn_error $? "conditional \"GTK_DOC_BUILD_PDF\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${GTK_DOC_USE_LIBTOOL_TRUE}" && test -z
"${GTK_DOC_USE_LIBTOOL_FALSE}"; then
    as_fn_error $? "conditional \"GTK_DOC_USE_LIBTOOL\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${GTK_DOC_USE_REBASE_TRUE}" && test -z
"${GTK_DOC_USE_REBASE_FALSE}"; then
    as_fn_error $? "conditional \"GTK_DOC_USE_REBASE\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi

: "${CONFIG_STATUS=./config.status}"
ac_write_fail=0
ac_clean_files_save=$ac_clean_files
ac_clean_files="$ac_clean_files $CONFIG_STATUS"
{ $as_echo "$as_me:${as_lineno-$LINENO}: creating $CONFIG_STATUS" >&5
$as_echo "$as_me: creating $CONFIG_STATUS" >&6;}
as_write_fail=0
cat >$CONFIG_STATUS <<_ASEOF || as_write_fail=1
#! $SHELL
# Generated by $as_me.
# Run this file to recreate the current configuration.
# Compiler output produced by configure, useful for debugging
# configure, is in config.log if it exists.

debug=false

```



```

    if test "X`(/usr/ucb/echo -n -n $as_echo) 2>/dev/null`" = "X-n
$as_echo"; then
        as_echo_body='eval /usr/ucb/echo -n "$1$as_nl"'
        as_echo_n='/usr/ucb/echo -n'
    else
        as_echo_body='eval expr "X$1" : "X\\(.*)\\)"'
        as_echo_n_body='eval
arg=$1;
case $arg in #(
*$as_nl"*)
expr "X$arg" : "X\\(.*)\\)$as_nl";
arg=`expr "X$arg" : ".*$as_nl\\(.*)\\)"`;
esac;
expr "X$arg" : "X\\(.*)\\)" | tr -d "$as_nl"
,
export as_echo_n_body
as_echo_n='sh -c $as_echo_n_body as_echo'
fi
export as_echo_body
as_echo='sh -c $as_echo_body as_echo'
fi

# The user is always right.
if test "${PATH_SEPARATOR+set}" != set; then
    PATH_SEPARATOR=:
    (PATH='/bin;/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 && {
        (PATH='/bin:/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 ||
        PATH_SEPARATOR=';'
    }
fi

# IFS
# We need space, tab and new line, in precisely that order. Quoting
is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " $as_nl

# Find who we are. Look in the path if we contain no directory
separator.
as_myself=
case $0 in #(
*[\ \/]* ) as_myself=$0 ;;
*) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
IFS=$as_save_IFS
test -z "$as_dir" && as_dir=.
test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done

```

```

IFS=$as_save_IFS

;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
    as_myself=$0
fi
if test ! -f "$as_myself"; then
    $as_echo "$as_myself: error: cannot find myself; rerun with an
absolute file name" >&2
    exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
# 1"
# suppresses any "Segmentation fault" message there. ' ((' could
# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH
do eval test x\${$as_var+set} = xset \
    && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

# NLS nuisances.
LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
# are
# provided, also output the error to LOG_FD, referencing LINENO. Then
# exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
    as_status=$1; test $as_status -eq 0 && as_status=1
    if test "$4"; then
        as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack

```

```

    $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
fi
$as_echo "$as_me: error: $2" >&2
as_fn_exit $as_status
} # as_fn_error

# as_fn_set_status STATUS
# -----
# Set $? to STATUS, without forking.
as_fn_set_status ()
{
    return $1
} # as_fn_set_status

# as_fn_exit STATUS
# -----
# Exit the shell with STATUS, even in a "trap 0" or "set -e" context.
as_fn_exit ()
{
    set +e
    as_fn_set_status $1
    exit $1
} # as_fn_exit

# as_fn_unset VAR
# -----
# Portably unset VAR.
as_fn_unset ()
{
    { eval $1=; unset $1;}
}
as_unset=as_fn_unset
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
VAR. Take
# advantage of any shell optimizations that allow amortized linear
growth over
# repeated appends, instead of the typical quadratic growth present in
naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\$as_var = x12") 2>/dev/null;
then :
    eval 'as_fn_append ()
    {
        eval $1+=\$2
    }'
else
    as_fn_append ()
    {
        eval $1=\$1\$2
    }
fi

```

```

    }
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
the
# global $as_val. Take advantage of shells that can avoid forks. The
arguments
# must be portable across $(( )) and expr.
if (eval "test \${( ( 1 + 1 ) )} = 2") 2>/dev/null; then :
    eval 'as_fn_arith ()
        {
            as_val=${( $* )}
        }'
else
    as_fn_arith ()
    {
        as_val=`expr "$@" || test $? -eq 1`
    }
fi # as_fn_arith

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
X/"; then
    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\([^/][^/]*\)/*$' \| \| \
X"$0" : 'X\(/\)\$' \| \| \
X"$0" : 'X\(/\)\' \| \| . 2>/dev/null ||
$as_echo X/"$0" |
sed '/^.*\([^/][^/]*\)\/*$/ {
    s//\1/
    q

```

```

}
/^X\\(\\|\\|\\)$/{
  s//\\1/
  q
}
/^X\\(\\|\\|\\).*/{
  s//\\1/
  q
}
s/.*/.;/ q'`

```

```

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyz'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

```

```

ECHO_C= ECHO_N= ECHO_T=
case `echo -n x` in #((((
-n*)
  case `echo 'xy\c'` in
  *c*) ECHO_T=' ';; # ECHO_T is single tab character.
  xy) ECHO_C='\c';;
  *) echo `echo ksh88 bug on AIX 6.1` > /dev/null
     ECHO_T=' ';;
  esac;;
*)
  ECHO_N='-n';;
esac

```

```

rm -f conf$$ conf$$exe conf$$file
if test -d conf$$dir; then
  rm -f conf$$dir/conf$$file
else
  rm -f conf$$dir
  mkdir conf$$dir 2>/dev/null
fi
if (echo >conf$$file) 2>/dev/null; then
  if ln -s conf$$file conf$$ 2>/dev/null; then
    as_ln_s='ln -s'
    # ... but there are two gotchas:
    # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
    # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
    executable.
    # In both cases, we have to default to `cp -pR'.
    ln -s conf$$file conf$$dir 2>/dev/null && test ! -f conf$$exe
  ||
    as_ln_s='cp -pR'
  elif ln conf$$file conf$$ 2>/dev/null; then
    as_ln_s=ln
  else

```

```

        as_ln_s='cp -pR'
    fi
else
    as_ln_s='cp -pR'
fi
rm -f conf$$ conf$$.$exe conf$$.$dir/conf$$.$file conf$$.$file
rmdir conf$$.$dir 2>/dev/null

# as_fn_mkdir_p
# -----
# Create "$as_dir" as a directory, including parents if necessary.
as_fn_mkdir_p ()
{
    case $as_dir in #(
    -*) as_dir=./$as_dir;;
    esac
    test -d "$as_dir" || eval $as_mkdir_p || {
        as_dirs=
        while ;; do
            case $as_dir in #(
            *'*) as_qdir=`$as_echo "$as_dir" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\/g"`;;
            #'(
            *) as_qdir=$as_dir;;
            esac
            as_dirs="'$as_qdir' $as_dirs"
            as_dir=`$as_dirname -- "$as_dir" ||
$as_expr X"$as_dir" : 'X\(.*[^\/]\)\([^\/][^\/]*/*$' \| \
    X"$as_dir" : 'X\(//\) [^/]' \| \
    X"$as_dir" : 'X\(//\) $' \| \
    X"$as_dir" : 'X\(//\) ' \| . 2>/dev/null ||
$as_echo X"$as_dir" |
    sed '/^X\(.*[^\/]\)\([^\/][^\/]*\/*$/{
        s//\1/
        q
    }
/^X\(\\\//\) [^/].*/{
    s//\1/
    q
}
/^X\(\\\//\) $/{
    s//\1/
    q
}
/^X\(\\\//\) .*/{
    s//\1/
    q
}
s/.*/.;/ q'`
            test -d "$as_dir" && break
        done
    }
}

```



```

    test -z "$sas_dirs" || eval "mkdir $sas_dirs"
  } || test -d "$sas_dir" || as_fn_error $? "cannot create directory
  $sas_dir"

} # as_fn_mkdir_p
if mkdir -p . 2>/dev/null; then
  as_mkdir_p='mkdir -p "$sas_dir"'
else
  test -d ./-p && rmdir ./-p
  as_mkdir_p=false
fi

# as_fn_executable_p FILE
# -----
# Test if FILE is an executable regular file.
as_fn_executable_p ()
{
  test -f "$1" && test -x "$1"
} # as_fn_executable_p
as_test_x='test -x'
as_executable_p=as_fn_executable_p

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed
'y%*$sas_cr_letters%P$sas_cr_LETTERS%;s%[^_ $sas_cr_alnum]%%_g'"

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_ $sas_cr_alnum]%%_g'"

exec 6>&1
## ----- ##
## Main body of $CONFIG_STATUS script. ##
## ----- ##
_ACEOF
test $sas_write_fail = 0 && chmod +x $CONFIG_STATUS || ac_write_fail=1

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# Save the log message, to keep $0 and so on meaningful, and to
# report actual input values of CONFIG_FILES etc. instead of their
# values after options handling.
ac_log="
This file was extended by dbus-glib $sas_me 0.100.2, which was
generated by GNU Autoconf 2.69.  Invocation command line was

CONFIG_FILES      = $CONFIG_FILES
CONFIG_HEADERS    = $CONFIG_HEADERS
CONFIG_LINKS      = $CONFIG_LINKS
CONFIG_COMMANDS   = $CONFIG_COMMANDS
$ $0 $@"

```

```
on `(hostname || uname -n) 2>/dev/null | sed 1q`  
"
```

```
_ACEOF
```

```
case $ac_config_files in *)  
  *) set x $ac_config_files; shift; ac_config_files=$*;;  
esac
```

```
case $ac_config_headers in *)  
  *) set x $ac_config_headers; shift; ac_config_headers=$*;;  
esac
```

```
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1  
# Files that config.status was made for.  
config_files="$ac_config_files"  
config_headers="$ac_config_headers"  
config_commands="$ac_config_commands"
```

```
_ACEOF
```

```
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1  
ac_cs_usage="\n  
`$as_me' instantiates files and other configuration actions  
from templates according to the current configuration. Unless the  
files  
and actions are specified as TAGs, all are instantiated by default.
```

```
Usage: $0 [OPTION]... [TAG]...
```

```
  -h, --help          print this help, then exit  
  -V, --version      print version number and configuration settings,  
then exit  
  --config          print configuration, then exit  
  -q, --quiet, --silent  
do not print progress messages  
  -d, --debug       don't remove temporary files  
  --recheck        update $as_me by reconfiguring in the same  
conditions  
  --file=FILE[:TEMPLATE]  
instantiate the configuration file FILE  
  --header=FILE[:TEMPLATE]  
instantiate the configuration header FILE
```

```
Configuration files:  
$config_files
```

```
Configuration headers:  
$config_headers
```

Configuration commands:
\$config_commands

Report bugs to
<https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLib>."

```
_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
ac_cs_config="\`$as_echo "$ac_configure_args" | sed 's/^ //;
s/[\\"'"`\\$]/\\\\&/g'\`"
ac_cs_version="\`
dbus-glib config.status 0.100.2
configured by $0, generated by GNU Autoconf 2.69,
with options \\"`$ac_cs_config`\`"
```

Copyright (C) 2012 Free Software Foundation, Inc.
This config.status script is free software; the Free Software
Foundation
gives unlimited permission to copy, distribute and modify it."

```
ac_pwd='$ac_pwd'
srcdir='$srcdir'
INSTALL='$INSTALL'
MKDIR_P='$MKDIR_P'
AWK='$AWK'
test -n "\`$AWK" || AWK=awk
_ACEOF
```

```
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# The default lists apply if the user does not specify any file.
ac_need_defaults=:
while test $# != 0
do
  case $1 in
    --*=?)
      ac_option=`expr "X$1" : 'X\[^\]=*\)`
      ac_optarg=`expr "X$1" : 'X\[^\]=*\(.*\)`'
      ac_shift=:
      ;;
    --*=)
      ac_option=`expr "X$1" : 'X\[^\]=*\)`
      ac_optarg=
      ac_shift=:
      ;;
    *)
      ac_option=$1
      ac_optarg=$2
      ac_shift=shift
      ;;
  esac
```

```

case $ac_option in
# Handling of the options.
-recheck | --recheck | --rehec | --reche | --rech | --rec | --re |
--r)
    ac_cs_recheck=: ;;
--version | --versio | --versi | --vers | --ver | --ve | --v | -V )
    $as_echo "$ac_cs_version"; exit ;;
--config | --confi | --conf | --con | --co | --c )
    $as_echo "$ac_cs_config"; exit ;;
--debug | --debu | --deb | --de | --d | -d )
    debug=: ;;
--file | --fil | --fi | --f )
    $ac_shift
    case $ac_optarg in
    *\'*) ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\'/g"`
;;
    ') as_fn_error $? "missing file argument" ;;
    esac
    as_fn_append CONFIG_FILES " '$ac_optarg'"
    ac_need_defaults=false;;
--header | --heade | --head | --hea )
    $ac_shift
    case $ac_optarg in
    *\'*) ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\\\\\\\\\\\\\\'/g"`
;;
    esac
    as_fn_append CONFIG_HEADERS " '$ac_optarg'"
    ac_need_defaults=false;;
--he | --h)
    # Conflict between --help and --header
    as_fn_error $? "ambiguous option: \`$1'
Try \`$0 --help' for more information.";;
--help | --hel | -h )
    $as_echo "$ac_cs_usage"; exit ;;
-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil | --si | --s)
    ac_cs_silent=: ;;

# This is an error.
-*) as_fn_error $? "unrecognized option: \`$1'
Try \`$0 --help' for more information." ;;

*) as_fn_append ac_config_targets " $1"
    ac_need_defaults=false ;;

    esac
    shift
done

ac_configure_extra_args=

if $ac_cs_silent; then

```

```

    exec 6>/dev/null
    ac_configure_extra_args="$ac_configure_extra_args --silent"
fi

_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
if \${ac_cs_recheck}; then
    set X $SHELL '$0' $ac_configure_args \${ac_configure_extra_args --no-
create --no-recursion
    shift
    \${as_echo} "running CONFIG_SHELL=$SHELL \${*}" >&6
    CONFIG_SHELL='$SHELL'
    export CONFIG_SHELL
    exec "\${@}"
fi

_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
exec 5>>config.log
{
    echo
    sed 'h;s/./-/g;s/^\.../## /;s/...$/ ##/;p;x;p;x' <<_ASBOX
## Running $as_me. ##
_ACEOF
    \${as_echo} "$ac_log"
} >&5

_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
#
# INIT-COMMANDS
#
AMDEP_TRUE="$AMDEP_TRUE" ac_aux_dir="$ac_aux_dir"

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

sed_quote_subst='${sed_quote_subst}'
double_quote_subst='${double_quote_subst}'
delay_variable_subst='${delay_variable_subst}'
macro_version='`$ECHO "$macro_version" | $SED
"$delay_single_quote_subst"`'
macro_revision='`$ECHO "$macro_revision" | $SED
"$delay_single_quote_subst"`'
enable_shared='`$ECHO "$enable_shared" | $SED
"$delay_single_quote_subst"`'
enable_static='`$ECHO "$enable_static" | $SED
"$delay_single_quote_subst"`'
pic_mode='`$ECHO "$pic_mode" | $SED "$delay_single_quote_subst`'`'

```

```
enable_fast_install='`$ECHO "$enable_fast_install" | $SED
"$delay_single_quote_subst"``'
SHELL='`$ECHO "$SHELL" | $SED "$delay_single_quote_subst"``'
ECHO='`$ECHO "$ECHO" | $SED "$delay_single_quote_subst"``'
PATH_SEPARATOR='`$ECHO "$PATH_SEPARATOR" | $SED
"$delay_single_quote_subst"``'
host_alias='`$ECHO "$host_alias" | $SED "$delay_single_quote_subst"``'
host='`$ECHO "$host" | $SED "$delay_single_quote_subst"``'
host_os='`$ECHO "$host_os" | $SED "$delay_single_quote_subst"``'
build_alias='`$ECHO "$build_alias" | $SED
"$delay_single_quote_subst"``'
build='`$ECHO "$build" | $SED "$delay_single_quote_subst"``'
build_os='`$ECHO "$build_os" | $SED "$delay_single_quote_subst"``'
SED='`$ECHO "$SED" | $SED "$delay_single_quote_subst"``'
Xsed='`$ECHO "$Xsed" | $SED "$delay_single_quote_subst"``'
GREP='`$ECHO "$GREP" | $SED "$delay_single_quote_subst"``'
EGREP='`$ECHO "$EGREP" | $SED "$delay_single_quote_subst"``'
FGREP='`$ECHO "$FGREP" | $SED "$delay_single_quote_subst"``'
LD='`$ECHO "$LD" | $SED "$delay_single_quote_subst"``'
NM='`$ECHO "$NM" | $SED "$delay_single_quote_subst"``'
LN_S='`$ECHO "$LN_S" | $SED "$delay_single_quote_subst"``'
max_cmd_len='`$ECHO "$max_cmd_len" | $SED
"$delay_single_quote_subst"``'
ac_objext='`$ECHO "$ac_objext" | $SED "$delay_single_quote_subst"``'
exeext='`$ECHO "$exeext" | $SED "$delay_single_quote_subst"``'
lt_unset='`$ECHO "$lt_unset" | $SED "$delay_single_quote_subst"``'
lt_SP2NL='`$ECHO "$lt_SP2NL" | $SED "$delay_single_quote_subst"``'
lt_NL2SP='`$ECHO "$lt_NL2SP" | $SED "$delay_single_quote_subst"``'
lt_cv_to_host_file_cmd='`$ECHO "$lt_cv_to_host_file_cmd" | $SED
"$delay_single_quote_subst"``'
lt_cv_to_tool_file_cmd='`$ECHO "$lt_cv_to_tool_file_cmd" | $SED
"$delay_single_quote_subst"``'
reload_flag='`$ECHO "$reload_flag" | $SED
"$delay_single_quote_subst"``'
reload_cmds='`$ECHO "$reload_cmds" | $SED
"$delay_single_quote_subst"``'
OBJDUMP='`$ECHO "$OBJDUMP" | $SED "$delay_single_quote_subst"``'
deplibs_check_method='`$ECHO "$deplibs_check_method" | $SED
"$delay_single_quote_subst"``'
file_magic_cmd='`$ECHO "$file_magic_cmd" | $SED
"$delay_single_quote_subst"``'
file_magic_glob='`$ECHO "$file_magic_glob" | $SED
"$delay_single_quote_subst"``'
want_nocaseglob='`$ECHO "$want_nocaseglob" | $SED
"$delay_single_quote_subst"``'
DLLTOOL='`$ECHO "$DLLTOOL" | $SED "$delay_single_quote_subst"``'
sharedlib_from_linklib_cmd='`$ECHO "$sharedlib_from_linklib_cmd" |
$SED "$delay_single_quote_subst"``'
AR='`$ECHO "$AR" | $SED "$delay_single_quote_subst"``'
AR_FLAGS='`$ECHO "$AR_FLAGS" | $SED "$delay_single_quote_subst"``'
archiver_list_spec='`$ECHO "$archiver_list_spec" | $SED
"$delay_single_quote_subst"``'
```

```
STRIP='`$ECHO "$STRIP" | $SED "$delay_single_quote_subst"``'
RANLIB='`$ECHO "$RANLIB" | $SED "$delay_single_quote_subst"``'
old_postinstall_cmds='`$ECHO "$old_postinstall_cmds" | $SED
"$delay_single_quote_subst"``'
old_postuninstall_cmds='`$ECHO "$old_postuninstall_cmds" | $SED
"$delay_single_quote_subst"``'
old_archive_cmds='`$ECHO "$old_archive_cmds" | $SED
"$delay_single_quote_subst"``'
lock_old_archive_extraction='`$ECHO "$lock_old_archive_extraction" |
$SED "$delay_single_quote_subst"``'
CC='`$ECHO "$CC" | $SED "$delay_single_quote_subst"``'
CFLAGS='`$ECHO "$CFLAGS" | $SED "$delay_single_quote_subst"``'
compiler='`$ECHO "$compiler" | $SED "$delay_single_quote_subst"``'
GCC='`$ECHO "$GCC" | $SED "$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_pipe='`$ECHO "$lt_cv_sys_global_symbol_pipe" |
$SED "$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_to_cdecl='`$ECHO
"$lt_cv_sys_global_symbol_to_cdecl" | $SED
"$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_to_c_name_address='`$ECHO
"$lt_cv_sys_global_symbol_to_c_name_address" | $SED
"$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix='`$ECHO
"$lt_cv_sys_global_symbol_to_c_name_address_lib_prefix" | $SED
"$delay_single_quote_subst"``'
nm_file_list_spec='`$ECHO "$nm_file_list_spec" | $SED
"$delay_single_quote_subst"``'
lt_sysroot='`$ECHO "$lt_sysroot" | $SED "$delay_single_quote_subst"``'
objdir='`$ECHO "$objdir" | $SED "$delay_single_quote_subst"``'
MAGIC_CMD='`$ECHO "$MAGIC_CMD" | $SED "$delay_single_quote_subst"``'
lt_prog_compiler_no_builtin_flag='`$ECHO
"$lt_prog_compiler_no_builtin_flag" | $SED
"$delay_single_quote_subst"``'
lt_prog_compiler_pic='`$ECHO "$lt_prog_compiler_pic" | $SED
"$delay_single_quote_subst"``'
lt_prog_compiler_wl='`$ECHO "$lt_prog_compiler_wl" | $SED
"$delay_single_quote_subst"``'
lt_prog_compiler_static='`$ECHO "$lt_prog_compiler_static" | $SED
"$delay_single_quote_subst"``'
lt_cv_prog_compiler_c_o='`$ECHO "$lt_cv_prog_compiler_c_o" | $SED
"$delay_single_quote_subst"``'
need_locks='`$ECHO "$need_locks" | $SED "$delay_single_quote_subst"``'
MANIFEST_TOOL='`$ECHO "$MANIFEST_TOOL" | $SED
"$delay_single_quote_subst"``'
DSYMUTIL='`$ECHO "$DSYMUTIL" | $SED "$delay_single_quote_subst"``'
NMEDIT='`$ECHO "$NMEDIT" | $SED "$delay_single_quote_subst"``'
LIPO='`$ECHO "$LIPO" | $SED "$delay_single_quote_subst"``'
OTOOL='`$ECHO "$OTOOL" | $SED "$delay_single_quote_subst"``'
OTOOL64='`$ECHO "$OTOOL64" | $SED "$delay_single_quote_subst"``'
libext='`$ECHO "$libext" | $SED "$delay_single_quote_subst"``'
shrext_cmds='`$ECHO "$shrext_cmds" | $SED
"$delay_single_quote_subst"``'
```

```
extract_expsyms_cmds='`$ECHO "$extract_expsyms_cmds" | $SED
"$delay_single_quote_subst"`'
archive_cmds_need_lc='`$ECHO "$archive_cmds_need_lc" | $SED
"$delay_single_quote_subst"`'
enable_shared_with_static_runtimes='`$ECHO
"$enable_shared_with_static_runtimes" | $SED
"$delay_single_quote_subst"`'
export_dynamic_flag_spec='`$ECHO "$export_dynamic_flag_spec" | $SED
"$delay_single_quote_subst"`'
whole_archive_flag_spec='`$ECHO "$whole_archive_flag_spec" | $SED
"$delay_single_quote_subst"`'
compiler_needs_object='`$ECHO "$compiler_needs_object" | $SED
"$delay_single_quote_subst"`'
old_archive_from_new_cmds='`$ECHO "$old_archive_from_new_cmds" | $SED
"$delay_single_quote_subst"`'
old_archive_from_expsyms_cmds='`$ECHO "$old_archive_from_expsyms_cmds"
| $SED "$delay_single_quote_subst"`'
archive_cmds='`$ECHO "$archive_cmds" | $SED
"$delay_single_quote_subst"`'
archive_expsym_cmds='`$ECHO "$archive_expsym_cmds" | $SED
"$delay_single_quote_subst"`'
module_cmds='`$ECHO "$module_cmds" | $SED
"$delay_single_quote_subst"`'
module_expsym_cmds='`$ECHO "$module_expsym_cmds" | $SED
"$delay_single_quote_subst"`'
with_gnu_ld='`$ECHO "$with_gnu_ld" | $SED
"$delay_single_quote_subst"`'
allow_undefined_flag='`$ECHO "$allow_undefined_flag" | $SED
"$delay_single_quote_subst"`'
no_undefined_flag='`$ECHO "$no_undefined_flag" | $SED
"$delay_single_quote_subst"`'
hardcode_libdir_flag_spec='`$ECHO "$hardcode_libdir_flag_spec" | $SED
"$delay_single_quote_subst"`'
hardcode_libdir_separator='`$ECHO "$hardcode_libdir_separator" | $SED
"$delay_single_quote_subst"`'
hardcode_direct='`$ECHO "$hardcode_direct" | $SED
"$delay_single_quote_subst"`'
hardcode_direct_absolute='`$ECHO "$hardcode_direct_absolute" | $SED
"$delay_single_quote_subst"`'
hardcode_minus_L='`$ECHO "$hardcode_minus_L" | $SED
"$delay_single_quote_subst"`'
hardcode_shlibpath_var='`$ECHO "$hardcode_shlibpath_var" | $SED
"$delay_single_quote_subst"`'
hardcode_automatic='`$ECHO "$hardcode_automatic" | $SED
"$delay_single_quote_subst"`'
inherit_rpath='`$ECHO "$inherit_rpath" | $SED
"$delay_single_quote_subst"`'
link_all_deplibs='`$ECHO "$link_all_deplibs" | $SED
"$delay_single_quote_subst"`'
always_export_symbols='`$ECHO "$always_export_symbols" | $SED
"$delay_single_quote_subst"`'
```



```
export_symbols_cmds='`$ECHO "$export_symbols_cmds" | $SED
"$delay_single_quote_subst"`'
exclude_expsyms='`$ECHO "$exclude_expsyms" | $SED
"$delay_single_quote_subst"`'
include_expsyms='`$ECHO "$include_expsyms" | $SED
"$delay_single_quote_subst"`'
prelink_cmds='`$ECHO "$prelink_cmds" | $SED
"$delay_single_quote_subst"`'
postlink_cmds='`$ECHO "$postlink_cmds" | $SED
"$delay_single_quote_subst"`'
file_list_spec='`$ECHO "$file_list_spec" | $SED
"$delay_single_quote_subst"`'
variables_saved_for_relink='`$ECHO "$variables_saved_for_relink" |
$SED "$delay_single_quote_subst"`'
need_lib_prefix='`$ECHO "$need_lib_prefix" | $SED
"$delay_single_quote_subst"`'
need_version='`$ECHO "$need_version" | $SED
"$delay_single_quote_subst"`'
version_type='`$ECHO "$version_type" | $SED
"$delay_single_quote_subst"`'
runpath_var='`$ECHO "$runpath_var" | $SED
"$delay_single_quote_subst"`'
shlibpath_var='`$ECHO "$shlibpath_var" | $SED
"$delay_single_quote_subst"`'
shlibpath_overrides_runpath='`$ECHO "$shlibpath_overrides_runpath" |
$SED "$delay_single_quote_subst"`'
libname_spec='`$ECHO "$libname_spec" | $SED
"$delay_single_quote_subst"`'
library_names_spec='`$ECHO "$library_names_spec" | $SED
"$delay_single_quote_subst"`'
soname_spec='`$ECHO "$soname_spec" | $SED
"$delay_single_quote_subst"`'
install_override_mode='`$ECHO "$install_override_mode" | $SED
"$delay_single_quote_subst"`'
postinstall_cmds='`$ECHO "$postinstall_cmds" | $SED
"$delay_single_quote_subst"`'
postuninstall_cmds='`$ECHO "$postuninstall_cmds" | $SED
"$delay_single_quote_subst"`'
finish_cmds='`$ECHO "$finish_cmds" | $SED
"$delay_single_quote_subst"`'
finish_eval='`$ECHO "$finish_eval" | $SED
"$delay_single_quote_subst"`'
hardcode_into_libs='`$ECHO "$hardcode_into_libs" | $SED
"$delay_single_quote_subst"`'
sys_lib_search_path_spec='`$ECHO "$sys_lib_search_path_spec" | $SED
"$delay_single_quote_subst"`'
sys_lib_dlsearch_path_spec='`$ECHO "$sys_lib_dlsearch_path_spec" |
$SED "$delay_single_quote_subst"`'
hardcode_action='`$ECHO "$hardcode_action" | $SED
"$delay_single_quote_subst"`'
enable_dlopen='`$ECHO "$enable_dlopen" | $SED
"$delay_single_quote_subst"`'
```

```

enable_dlopen_self='`$ECHO "$enable_dlopen_self" | $SED
"$delay_single_quote_subst"`'
enable_dlopen_self_static='`$ECHO "$enable_dlopen_self_static" | $SED
"$delay_single_quote_subst"`'
old_striplib='`$ECHO "$old_striplib" | $SED
"$delay_single_quote_subst"`'
striplib='`$ECHO "$striplib" | $SED "$delay_single_quote_subst"`'

LTCC='$LTCC'
LTCFLAGS='$LTCFLAGS'
compiler='$compiler_DEFAULT'

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
    eval 'cat <<_LTECHO_EOF'
    \ $1
    _LTECHO_EOF'
}

# Quote evaled strings.
for var in SHELL \
ECHO \
PATH_SEPARATOR \
SED \
GREP \
EGREP \
FGREP \
LD \
NM \
LN_S \
lt_SP2NL \
lt_NL2SP \
reload_flag \
OBJDUMP \
deplibs_check_method \
file_magic_cmd \
file_magic_glob \
want_nocaseglob \
DLLTOOL \
sharedlib_from_linklib_cmd \
AR \
AR_FLAGS \
archiver_list_spec \
STRIP \
RANLIB \
CC \
CFLAGS \
compiler \
lt_cv_sys_global_symbol_pipe \
lt_cv_sys_global_symbol_to_cdecl \
lt_cv_sys_global_symbol_to_c_name_address \

```

```

lt_cv_sys_global_symbol_to_c_name_address_lib_prefix \
nm_file_list_spec \
lt_prog_compiler_no_builtin_flag \
lt_prog_compiler_pic \
lt_prog_compiler_wl \
lt_prog_compiler_static \
lt_cv_prog_compiler_c_o \
need_locks \
MANIFEST_TOOL \
DSYMUTIL \
NMEDIT \
LIPO \
OTOOL \
OTOOL64 \
shrext_cmds \
export_dynamic_flag_spec \
whole_archive_flag_spec \
compiler_needs_object \
with_gnu_ld \
allow_undefined_flag \
no_undefined_flag \
hardcode_libdir_flag_spec \
hardcode_libdir_separator \
exclude_expsyms \
include_expsyms \
file_list_spec \
variables_saved_for_relink \
libname_spec \
library_names_spec \
soname_spec \
install_override_mode \
finish_eval \
old_striplib \
striplib; do
    case `eval \\\\\\\$ECHO \\\\\\\"\\\\\\\\\\$\\$var"\\\\\\\\\\"` in
    *[\\\\\\\\\\\`\\\\\\"\\\\\\\\\\$]*)
        eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\\\\\$ECHO \\\\\\\"\\\\\\\\\\$\\$var\\\\\\\\\\" | \\\\\\\$SED
\\\\\\\\\\$sed_quote_subst\\\\\\\\\\"\\\\\\\\\\`\\\\\\\\\\\\\\\\\\""
        ;;
    *)
        eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\\\\\$\\$var\\\\\\\\\\\\\\\\\\""
        ;;
    esac
done

# Double-quote double-eval'd strings.
for var in reload_cmds \
old_postinstall_cmds \
old_postuninstall_cmds \
old_archive_cmds \
extract_expsyms_cmds \
old_archive_from_new_cmds \

```

```

old_archive_from_expsyms_cmds \
archive_cmds \
archive_expsym_cmds \
module_cmds \
module_expsym_cmds \
export_symbols_cmds \
prelink_cmds \
postlink_cmds \
postinstall_cmds \
postuninstall_cmds \
finish_cmds \
sys_lib_search_path_spec \
sys_lib_dlsearch_path_spec; do
  case `eval \\\\\\\$ECHO \\\\\\\"\\\\\\$\\$var"\\\\\\"` in
    *[\\\\\\\`\\\\\\"\\\\\\$]*)
      eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\`\\\\\\$ECHO \\\\\\\"\\\\\\$\\$var\\\\\\" | \\\\\\\$SED -e
\\\\\\`\\\\\\$double_quote_subst\\\\\\` -e \\\\\\\`\\\\\\$sed_quote_subst\\\\\\` -e
\\\\\\`\\\\\\$delay_variable_subst\\\\\\`\\\\\\`\\\\\\\\\\\\\\\\\\""
      ;;
    *)
      eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\$\\$var\\\\\\\\\\\\\\\\\\""
      ;;
  esac
done

ac_aux_dir='$ac_aux_dir'
xsi_shell='$xsi_shell'
lt_shell_append='$lt_shell_append'

# See if we are running on zsh, and set the options which allow our
# commands through without removal of \ escapes INIT.
if test -n "\\$${ZSH_VERSION+set}" ; then
  setopt NO_GLOB_SUBST
fi

PACKAGE='$PACKAGE'
VERSION='$VERSION'
TIMESTAMP='$TIMESTAMP'
RM='$RM'
ofile='$ofile'

_ACEOF

cat >>$CONFIG_STATUS <<\\_ACEOF || ac_write_fail=1

# Handling of arguments.
for ac_config_target in $ac_config_targets
do

```

```

case $ac_config_target in
  "config.h") CONFIG_HEADERS="$CONFIG_HEADERS config.h" ;;
  "depfiles") CONFIG_COMMANDS="$CONFIG_COMMANDS depfiles" ;;
  "libtool") CONFIG_COMMANDS="$CONFIG_COMMANDS libtool" ;;
  "Makefile") CONFIG_FILES="$CONFIG_FILES Makefile" ;;
  "m4/Makefile") CONFIG_FILES="$CONFIG_FILES m4/Makefile" ;;
  "doc/Makefile") CONFIG_FILES="$CONFIG_FILES doc/Makefile" ;;
  "doc/reference/Makefile") CONFIG_FILES="$CONFIG_FILES
doc/reference/Makefile" ;;
  "doc/reference/version.xml") CONFIG_FILES="$CONFIG_FILES
doc/reference/version.xml" ;;
  "dbus/Makefile") CONFIG_FILES="$CONFIG_FILES dbus/Makefile" ;;
  "dbus/examples/Makefile") CONFIG_FILES="$CONFIG_FILES
dbus/examples/Makefile" ;;
  "dbus/examples/statemachine/Makefile") CONFIG_FILES="$CONFIG_FILES
dbus/examples/statemachine/Makefile" ;;
  "test/Makefile") CONFIG_FILES="$CONFIG_FILES test/Makefile" ;;
  "test/core/Makefile") CONFIG_FILES="$CONFIG_FILES
test/core/Makefile" ;;
  "test/interfaces/Makefile") CONFIG_FILES="$CONFIG_FILES
test/interfaces/Makefile" ;;
  "test/data/valid-service-files/debug-glib.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/debug-
glib.service" ;;
  "test/data/valid-service-files/debug-echo.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/debug-
echo.service" ;;
  "test/data/valid-service-files/interfaces-test.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files/interfaces-
test.service" ;;
  "test/lib/Makefile") CONFIG_FILES="$CONFIG_FILES
test/lib/Makefile" ;;
  "test/manual/Makefile") CONFIG_FILES="$CONFIG_FILES
test/manual/Makefile" ;;
  "tools/Makefile") CONFIG_FILES="$CONFIG_FILES tools/Makefile" ;;
  "dbus-glib-1.pc") CONFIG_FILES="$CONFIG_FILES dbus-glib-1.pc" ;;
  "dbus-glib-1-uninstalled.pc") CONFIG_FILES="$CONFIG_FILES dbus-
glib-1-uninstalled.pc" ;;

  *) as_fn_error $? "invalid argument: \`${ac_config_target}'" "$LINENO"
5;;
esac
done

# If the user did not use the arguments to specify the items to
# instantiate,
# then the envvar interface is used. Set only those that are not.
# We use the long form for the default assignment because of an
# extremely
# bizarre bug on SunOS 4.1.3.
if $ac_need_defaults; then

```

```

    test "${CONFIG_FILES+set}" = set || CONFIG_FILES=$config_files
    test "${CONFIG_HEADERS+set}" = set || CONFIG_HEADERS=$config_headers
    test "${CONFIG_COMMANDS+set}" = set ||
CONFIG_COMMANDS=$config_commands
fi

# Have a temporary directory for convenience.  Make it in the build
tree
# simply because there is no reason against having it here, and in
addition,
# creating and moving files from /tmp can sometimes cause problems.
# Hook for its removal unless debugging.
# Note that there is a small window in which the directory will not be
cleaned:
# after its creation but before its name has been assigned to `$tmp'.
$debug ||
{
    tmp= ac_tmp=
    trap 'exit_status=$?'
    : "${ac_tmp:= $tmp}"
    { test ! -d "$ac_tmp" || rm -fr "$ac_tmp"; } && exit $exit_status
' 0
    trap 'as_fn_exit 1' 1 2 13 15
}
# Create a (secure) tmp directory for tmp files.

{
    tmp=`(umask 077 && mktemp -d "./confXXXXXX") 2>/dev/null` &&
    test -d "$tmp"
} ||
{
    tmp=./conf$$-$RANDOM
    (umask 077 && mkdir "$tmp")
} || as_fn_error $? "cannot create a temporary directory in ."
"$LINENO" 5
ac_tmp=$tmp

# Set up the scripts for CONFIG_FILES section.
# No need to generate them if there are no CONFIG_FILES.
# This happens for instance with `./config.status config.h'.
if test -n "$CONFIG_FILES"; then

ac_cr=`echo X | tr X '\015'`
# On cygwin, bash can eat \r inside `` if the user requested igncr.
# But we know of no other shell where ac_cr would be empty at this
# point, so we can use a bashism as a fallback.
if test "x$ac_cr" = x; then
    eval ac_cr=\$\`\\r\`
fi
ac_cs_awk_cr=`$AWK 'BEGIN { print "a\rb" }' </dev/null 2>/dev/null`
if test "$ac_cs_awk_cr" = "a${ac_cr}b"; then

```

```

    ac_cs_awk_cr='\r'
else
    ac_cs_awk_cr=$ac_cr
fi

echo 'BEGIN {' >"$ac_tmp/subs1.awk" &&
_ACEOF

{
    echo "cat >conf$$subs.awk <<_ACEOF" &&
    echo "$ac_subst_vars" | sed 's/.*&!$&$ac_delim/' &&
    echo "_ACEOF"
} >conf$$subs.sh ||
    as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5
ac_delim_num=`echo "$ac_subst_vars" | grep -c '^`
ac_delim='%!_!# '
for ac_last_try in false false false false false ;; do
    . ./conf$$subs.sh ||
        as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5

    ac_delim_n=`sed -n "s/.*$ac_delim$/X/p" conf$$subs.awk | grep -c X`
    if test $ac_delim_n = $ac_delim_num; then
        break
    elif $ac_last_try; then
        as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5
    else
        ac_delim="$ac_delim!$ac_delim _$ac_delim!! "
    fi
done
rm -f conf$$subs.sh

cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
cat >>"$ac_tmp/subs1.awk" <<\_ACAWK &&
_ACEOF
sed -n '
h
s/^\S["/; s/!.*"/]=/
p
g
s/^[^!]*!//
:repl
t repl
s/"$ac_delim"$$/
t delim
:nl
h
s/\(.\\{148\\}\)\.*/\1/
t more1
s/["\\]\/\&/g; s/^\//; s/$/\\n"\\/
p
n

```

```

b repl
:more1
s/["\\]/\\&/g; s/^"/; s/$/"\\//
p
g
s/.\{148\}//
t nl
:delim
h
s/\(.\{148\}\)\..*/\1/
t more2
s/["\\]/\\&/g; s/^"/; s/$"/
p
b
:more2
s/["\\]/\\&/g; s/^"/; s/$/"\\//
p
g
s/.\{148\}//
t delim
' <conf$$subs.awk | sed '
/^[^"]*/{
    N
    s/\n//
}
' >>$CONFIG_STATUS || ac_write_fail=1
rm -f conf$$subs.awk
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
_ACAWK
cat >>"\$ac_tmp/subs1.awk" <<_ACAWK &&
    for (key in S) S_is_set[key] = 1
    FS = " "
}
{
    line = $ 0
    nfields = split(line, field, "@")
    substed = 0
    len = length(field[1])
    for (i = 2; i < nfields; i++) {
        key = field[i]
        keylen = length(key)
        if (S_is_set[key]) {
            value = S[key]
            line = substr(line, 1, len) "" value "" substr(line, len +
keylen + 3)
            len += length(value) + length(field[++i])
            substed = 1
        } else
            len += 1 + keylen
    }
}

```



```

    print line
}

_ACAWK
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
if sed "s/$ac_cr//" < /dev/null > /dev/null 2>&1; then
    sed "s/$ac_cr\\$//; s/$ac_cr/$ac_cs_awk_cr/g"
else
    cat
fi < "$ac_tmp/subs1.awk" > "$ac_tmp/subs.awk" \
|| as_fn_error $? "could not setup config files machinery" "$LINENO"
5
_ACEOF

# VPATH may cause trouble with some makes, so we remove sole
$(srcdir),
# ${srcdir} and @srcdir@ entries from VPATH if srcdir is ".", strip
leading and
# trailing colons and then remove the whole line if VPATH becomes
empty
# (actually we leave an empty line to preserve line numbers).
if test "x$srcdir" = x.; then
    ac_vpsub='/^[ ]*VPATH[ ]*=[ ]*{
h
s///
s/^\:/:/
s/[ ]*$\:/:/
s/:\$(srcdir)\:/:/g
s/:\${srcdir}\:/:/g
s/:\@srcdir@\:/:/g
s/^\:*\:/:/
s/:\:*\:/:/
x
s/\(=[ ]*\)\.*/\1/
G
s/\n//
s/^[^=]*=[ ]*$//
}'
fi

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
fi # test -n "$CONFIG_FILES"

# Set up the scripts for CONFIG_HEADERS section.
# No need to generate them if there are no CONFIG_HEADERS.
# This happens for instance with './config.status Makefile'.
if test -n "$CONFIG_HEADERS"; then
cat >"$ac_tmp/defines.awk" <<\_ACAOK ||
BEGIN {
_ACEOF

```

```

# Transform confdefs.h into an awk script `defines.awk', embedded as
# here-document in config.status, that substitutes the proper values
into
# config.h.in to produce config.h.

# Create a delimiter string that does not exist in confdefs.h, to ease
# handling of long lines.
ac_delim='%!_!# '
for ac_last_try in false false ;; do
  ac_tt=`sed -n "/$ac_delim/p" confdefs.h`
  if test -z "$ac_tt"; then
    break
  elif $ac_last_try; then
    as_fn_error $? "could not make $CONFIG_HEADERS" "$LINENO" 5
  else
    ac_delim="$ac_delim!$ac_delim _$ac_delim!! "
  fi
done

# For the awk script, D is an array of macro values keyed by name,
# likewise P contains macro parameters if any. Preserve backslash
# newline sequences.

ac_word_re=[_$as_cr_Letters][_ $as_cr_alnum]*
sed -n '
s/.\{148\}/&"$ac_delim"/g
t rset
:rset
s/^[ ]*#[ ]*define[ ]*[ ]*/ /
t def
d
:def
s/\\$/ /
t bsnl
s/["\\]/\\&/g
s/^\ ("$ac_word_re"\)\ ([[^\()]*\)\ [ ]*\ (.*)/P["\1"]="\2"\
D["\1"]=" \3"/p
s/^\ ("$ac_word_re"\)\ [ ]*\ (.*)/D["\1"]=" \2"/p
d
:bsnl
s/["\\]/\\&/g
s/^\ ("$ac_word_re"\)\ ([[^\()]*\)\ [ ]*\ (.*)/P["\1"]="\2"\
D["\1"]=" \3\\n"/p
t cont
s/^\ ("$ac_word_re"\)\ [ ]*\ (.*)/D["\1"]=" \2\\n"/p
t cont
d
:cont
n
s/.\{148\}/&"$ac_delim"/g
t clear
:clear

```

```

s/\\$//
t bsnlc
s/["\\]/\\&/g; s/^"/; s/$"/p
d
:bsnlc
s/["\\]/\\&/g; s/^"/; s/$/\\\\\\n"\\//p
b cont
' <confdefs.h | sed '
s/"$ac_delim"/"\\
"/g' >>$CONFIG_STATUS || ac_write_fail=1

cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
  for (key in D) D_is_set[key] = 1
  FS = " "
}
/^[\t ]*#[\t ]*(define|undef)[\t ]+$ac_word_re([\t ()|\$])/ {
  line = \$0
  split(line, arg, " ")
  if (arg[1] == "#") {
    defundef = arg[2]
    mac1 = arg[3]
  } else {
    defundef = substr(arg[1], 2)
    mac1 = arg[2]
  }
  split(mac1, mac2, "(") #)
  macro = mac2[1]
  prefix = substr(line, 1, index(line, defundef) - 1)
  if (D_is_set[macro]) {
    # Preserve the white space surrounding the "#".
    print prefix "define", macro P[macro] D[macro]
    next
  } else {
    # Replace #undef with comments. This is necessary, for example,
    # in the case of _POSIX_SOURCE, which is predefined and required
    # on some systems where configure will not decide to define it.
    if (defundef == "undef") {
      print "/*", prefix defundef, macro, "*/"
      next
    }
  }
}
}
{ print }
_ACAWK
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
  as_fn_error $? "could not setup config headers machinery" "$LINENO"
5
fi # test -n "$CONFIG_HEADERS"

```



```

    $as_echo "$*" | sed 's|^[^:]*/||;s|:[^:]*/|, |g'
  ` by configure.'
  if test x"$ac_file" != x-; then
    configure_input="$ac_file. $configure_input"
    { $as_echo "$as_me:${as_lineno-$LINENO}: creating $ac_file" >&5
$as_echo "$as_me: creating $ac_file" >&6;}
  fi
  # Neutralize special characters interpreted by sed in replacement
  strings.
  case $configure_input in
    *\&* | *\\|* | *\\* )
    ac_sed_conf_input=`$as_echo "$configure_input" |
sed 's/[\\&|]/\\\\&/g'`; ; # (
*) ac_sed_conf_input=$configure_input;;
  esac

  case $ac_tag in
    *:-:* | *:-) cat >"$ac_tmp/stdin" \
    || as_fn_error $? "could not create $ac_file" "$LINENO" 5 ;;
  esac
  ;;
esac

ac_dir=`$as_dirname -- "$ac_file" ||
$as_expr X"$ac_file" : 'X\([^/]\)\/*[^/][^/]*/*$' \| \
X"$ac_file" : 'X\(/\)\[^/]' \| \
X"$ac_file" : 'X\(/\)\$' \| \
X"$ac_file" : 'X\(/)' \| . 2>/dev/null ||
$as_echo X"$ac_file" |
sed '/^X\([^/]\)\/*[^/][^/]*\/*$/{
s//\1/
q
}
/^X\(\(\)\)\[^/].*${
s//\1/
q
}
/^X\(\(\)\)\$/{
s//\1/
q
}
/^X\(\)\.*${
s//\1/
q
}
s/././; q'`
ac_dir="$ac_dir"; as_fn_mkdir_p
ac_buildidir=.

case "$ac_dir" in
.) ac_dir_suffix= ac_top_buildidir_sub=. ac_top_build_prefix= ;;
*)

```

```

ac_dir_suffix=/`$as_echo "$ac_dir" | sed 's|^\.([\//]|||)`
# A ".." for each directory in $ac_dir_suffix.
ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/[^\\/]*/|/..|g;s|/|||`
case $ac_top_builddir_sub in
  "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
  *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
esac ;;
esac
ac_abs_top_builddir=$ac_pwd
ac_abs_builddir=$ac_pwd$ac_dir_suffix
# for backward compatibility:
ac_top_builddir=$ac_top_build_prefix

case $srcdir in
  .) # We are building in place.
    ac_srcdir=.
    ac_top_srcdir=$ac_top_builddir_sub
    ac_abs_top_srcdir=$ac_pwd ;;
  [\\/* | ?:[\\/*]* ) # Absolute name.
    ac_srcdir=$srcdir$ac_dir_suffix;
    ac_top_srcdir=$srcdir
    ac_abs_top_srcdir=$srcdir ;;
  *) # Relative name.
    ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
    ac_top_srcdir=$ac_top_build_prefix$srcdir
    ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

case $ac_mode in
:F)
#
# CONFIG_FILE
#

case $INSTALL in
[\\/$]* | ?:[\\/*]* ) ac_INSTALL=$INSTALL ;;
*) ac_INSTALL=$ac_top_build_prefix$INSTALL ;;
esac
ac_MKDIR_P=$MKDIR_P
case $MKDIR_P in
[\\/$]* | ?:[\\/*]* ) ;;
*/*) ac_MKDIR_P=$ac_top_build_prefix$MKDIR_P ;;
esac
_ACEOF

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# If the template does not know about datarootdir, expand it.
# FIXME: This hack should be removed a few years after 2.60.
ac_datarootdir_hack=; ac_datarootdir_seen=

```

```

ac_sed_dataroot='
/datarootdir/ {
  p
  q
}
/@datadir@/p
/@docdir@/p
/@infodir@/p
/@localedir@/p
/@mandir@/p'
case `eval "sed -n \"\$ac_sed_dataroot\" \$ac_file_inputs"` in
*datarootdir*) ac_datarootdir_seen=yes;;
*@datadir@*|*@docdir@*|*@infodir@*|*@localedir@*|*@mandir@*)
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file_inputs
seems to ignore the --datarootdir setting" >&5
$as_echo "$as_me: WARNING: $ac_file_inputs seems to ignore the --
datarootdir setting" >&2;}
_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
  ac_datarootdir_hack='
  s&@datadir@&${datadir}&g
  s&@docdir@&${docdir}&g
  s&@infodir@&${infodir}&g
  s&@localedir@&${localedir}&g
  s&@mandir@&${mandir}&g
  s&\\\$${datarootdir}&${datarootdir}&g' ;;
esac
_ACEOF

# Neutralize VPATH when `$srcdir' = `.'.
# Shell code in configure.ac might set extrasub.
# FIXME: do we really want to maintain this feature?
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
ac_sed_extra="$ac_vpsub
$extrasub
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
:t
/@[a-zA-Z_][a-zA-Z_0-9]*@/!b
s|@configure_input@|${ac_sed_conf_input}|;t t
s&@top_buildddir@&${ac_top_buildddir_sub}&;t t
s&@top_build_prefix@&${ac_top_build_prefix}&;t t
s&@srcdir@&${ac_srcdir}&;t t
s&@abs_srcdir@&${ac_abs_srcdir}&;t t
s&@top_srcdir@&${ac_top_srcdir}&;t t
s&@abs_top_srcdir@&${ac_abs_top_srcdir}&;t t
s&@builddir@&${ac_builddir}&;t t
s&@abs_builddir@&${ac_abs_builddir}&;t t
s&@abs_top_builddir@&${ac_abs_top_builddir}&;t t
s&@INSTALL@&${ac_INSTALL}&;t t
s&@MKDIR_P@&${ac_MKDIR_P}&;t t
$ac_datarootdir_hack

```

```

"
eval sed "\"\$ac_sed_extra\" \"$ac_file_inputs" | $AWK -f
"$ac_tmp/subs.awk" \
  >$ac_tmp/out || as_fn_error $? "could not create $ac_file" "$LINENO"
5

test -z "$ac_datarootdir_hack$ac_datarootdir_seen" &&
{ ac_out=`sed -n '/\${datarootdir}/p' "$ac_tmp/out"`; test -n
"$ac_out"; } &&
{ ac_out=`sed -n '/^[ ]*datarootdir[ ]*:*=/p' \
  "$ac_tmp/out"`; test -z "$ac_out"; } &&
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file contains
a reference to the variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&5
$as_echo "$as_me: WARNING: $ac_file contains a reference to the
variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&2;}

rm -f "$ac_tmp/stdin"
case $ac_file in
-) cat "$ac_tmp/out" && rm -f "$ac_tmp/out";;
*) rm -f "$ac_file" && mv "$ac_tmp/out" "$ac_file";;
esac \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
;;
:H)
#
# CONFIG_HEADER
#
if test x"$ac_file" != x-; then
{
  $as_echo "/* $configure_input */" \
  && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs"
} >"$ac_tmp/config.h" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5
if diff "$ac_file" "$ac_tmp/config.h" >/dev/null 2>&1; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: $ac_file is unchanged"
>&5
$as_echo "$as_me: $ac_file is unchanged" >&6;}
else
rm -f "$ac_file"
mv "$ac_tmp/config.h" "$ac_file" \
  || as_fn_error $? "could not create $ac_file" "$LINENO" 5
fi
else
$as_echo "/* $configure_input */" \
  && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs" \
  || as_fn_error $? "could not create -" "$LINENO" 5
fi
# Compute "$ac_file"'s index in $config_headers.
_am_arg="$ac_file"
_am_stamp_count=1

```



```

for _am_header in $config_headers ;; do
  case $_am_header in
    $_am_arg | $_am_arg:* )
      break ;;
    * )
      _am_stamp_count=`expr $_am_stamp_count + 1` ;;
  esac
done
echo "timestamp for $_am_arg" >`$as_dirname -- "$_am_arg" ||
$as_expr X"$_am_arg" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \|| \
  X"$_am_arg" : 'X\(//\)[^/]' \|| \
  X"$_am_arg" : 'X\(//\)$' \|| \
  X"$_am_arg" : 'X\(/\)' \|| . 2>/dev/null ||
$as_echo X"$_am_arg" |
  sed '/^X\(.*[^/]\)\|\|\/*[^/][^/]*\/*$/{
    s//\1/
    q
  }
/^X\(\|\|\|\)\[^/].*/{
  s//\1/
  q
}
/^X\(\|\|\|\)$/{
  s//\1/
  q
}
/^X\(\|\|\)\.*/{
  s//\1/
  q
}
s/.*\/.*/; q'`/stamp-h$_am_stamp_count
;;

:C) { $as_echo "$as_me:${as_lineno-$LINENO}: executing $ac_file
commands" >&5
$as_echo "$as_me: executing $ac_file commands" >&6;}
;;
esac

case $ac_file$ac_mode in
  "depfiles":C) test x"$SAMDEP_TRUE" != x"" || {
  # Autoconf 2.62 quotes --file arguments for eval, but not when files
  # are listed without --file. Let's play safe and only enable the
eval
  # if we detect the quoting.
  case $CONFIG_FILES in
    *\') eval set x "$CONFIG_FILES" ;;
    *) set x $CONFIG_FILES ;;
  esac
  shift
  for mf

```

```

do
  # Strip MF so we end up with the name of the file.
  mf=`echo "$mf" | sed -e 's/:.*$//'\`
  # Check whether this is an Automake generated Makefile or not.
  # We used to match only the files named 'Makefile.in', but
  # some people rename them; so instead we look at the file content.
  # Grep'ing the first line is not enough: some people post-process
  # each Makefile.in and add a new line on top of each file to say
so.
  # Grep'ing the whole file is not good either: AIX grep has a line
  # limit of 2048, but all sed's we know have understand at least
4000.
  if sed -n 's,^#.*generated by automake.*,X,p' "$mf" | grep X
>/dev/null 2>&1; then
    dirpart=`$as_dirname -- "$mf" ||
$as_expr X"$mf" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \| \
  X"$mf" : 'X\(//\)[^/]' \| \
  X"$mf" : 'X\(//\)$' \| \
  X"$mf" : 'X\(/\)' \| . 2>/dev/null ||
$as_echo X"$mf" |
  sed '/^X\(.*[^/]\)\/*[^/][^/]*/*$/{
    s//\1/
    q
  }
/^X\(\\/\)\)[^/].*/{
  s//\1/
  q
}
/^X\(\\/\)\)$/{
  s//\1/
  q
}
/^X\(\\/\).*/{
  s//\1/
  q
}
s/././; q'\`
  else
    continue
  fi
  # Extract the definition of DEPDIR, am__include, and am__quote
  # from the Makefile without running 'make'.
  DEPDIR=`sed -n 's/^DEPDIR = //p' < "$mf"`
  test -z "$DEPDIR" && continue
  am__include=`sed -n 's/^am__include = //p' < "$mf"`
  test -z "am__include" && continue
  am__quote=`sed -n 's/^am__quote = //p' < "$mf"`
  # Find all dependency output files, they are included files with
  # $(DEPDIR) in their names. We invoke sed twice because it is the
  # simplest approach to changing $(DEPDIR) to its actual value in
the
  # expansion.

```

```

    for file in `sed -n "
        s/^$sam__include $sam__quote\(.*(DEPDIR).*\) $sam__quote"'$/\1/p'
<"$mf" | \
        sed -e 's/\$(DEPDIR)/'"$DEPDIR"'/g`; do
        # Make sure the directory exists.
        test -f "$dirpart/$file" && continue
        fdir=`$as_dirname -- "$file" ||
$as_expr X"$file" : 'X\([^\/]\)\/*\([^\/]\)\/*$' \| \
        X"$file" : 'X\(\//\)\[\^\/]' \| \
        X"$file" : 'X\(\//\)$' \| \
        X"$file" : 'X\(\//\)' \| . 2>/dev/null ||
$as_echo X"$file" |
        sed '/^X\([^\/]\)\//\/*\([^\/]\)\/*$/{
            s//\1/
            q
        }
/^X\(\//\)\[\^\/].*/{
            s//\1/
            q
        }
/^X\(\//\)\$/{
            s//\1/
            q
        }
/^X\(\//\).*/{
            s//\1/
            q
        }
s/.*/./; q'`
        as_dir=$dirpart/$fdir; as_fn_mkdir_p
        # echo "creating $dirpart/$file"
        echo '# dummy' > "$dirpart/$file"
    done
done
}
;;
"libtool":C)

# See if we are running on zsh, and set the options which allow
our
# commands through without removal of \ escapes.
if test -n "${ZSH_VERSION+set}" ; then
    setopt NO_GLOB_SUBST
fi

cfgfile="${ofile}T"
trap "$RM \"\$cfgfile\"; exit 1" 1 2 15
$RM "$cfgfile"

cat <<_LT_EOF >> "$cfgfile"
#! $SHELL

```

```
# `ECHO "$ofile" | sed 's%^.*/%%'` - Provide generalized library-
building support services.
# Generated automatically by $as_me ($PACKAGE$TIMESTAMP) $VERSION
# Libtool was configured on host `(hostname || uname -n) 2>/dev/null |
sed lq`:
# NOTE: Changes made to this file will be lost: look at ltmain.sh.
#
# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005,
#           2006, 2007, 2008, 2009, 2010, 2011 Free Software
#           Foundation, Inc.
# Written by Gordon Matzigkeit, 1996
#
# This file is part of GNU Libtool.
#
# GNU Libtool is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of
# the License, or (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html, or
# obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

# The names of the tagged configurations supported by this script.
available_tags=""

# ### BEGIN LIBTOOL CONFIG

# Which release of libtool.m4 was used?
macro_version=$macro_version
macro_revision=$macro_revision

# Whether or not to build shared libraries.
build_libtool_libs=$enable_shared

# Whether or not to build static libraries.
build_old_libs=$enable_static
```

```
# What type of objects to build.
pic_mode=$pic_mode

# Whether or not to optimize for fast installation.
fast_install=$enable_fast_install

# Shell to use when invoking shell scripts.
SHELL=$lt_SHELL

# An echo program that protects backslashes.
ECHO=$lt_ECHO

# The PATH separator for the build system.
PATH_SEPARATOR=$lt_PATH_SEPARATOR

# The host system.
host_alias=$host_alias
host=$host
host_os=$host_os

# The build system.
build_alias=$build_alias
build=$build
build_os=$build_os

# A sed program that does not truncate output.
SED=$lt_SED

# Sed that helps us avoid accidentally triggering echo(1) options like
-n.
Xsed="\$SED -e 1s/^X//"

# A grep program that handles long lines.
GREP=$lt_GREP

# An ERE matcher.
EGREP=$lt_EGREP

# A literal string matcher.
FGREP=$lt_FGREP

# A BSD- or MS-compatible name lister.
NM=$lt_NM

# Whether we need soft or hard links.
LN_S=$lt_LN_S

# What is the maximum length of a command?
max_cmd_len=$max_cmd_len

# Object file suffix (normally "o").
objext=$ac_objext
```

```
# Executable file suffix (normally "").
exeext=$exeext

# whether the shell understands "unset".
lt_unset=$lt_unset

# turn spaces into newlines.
SP2NL=$lt_lt_SP2NL

# turn newlines into spaces.
NL2SP=$lt_lt_NL2SP

# convert \${build} file names to \${host} format.
to_host_file_cmd=$lt_cv_to_host_file_cmd

# convert \${build} files to toolchain format.
to_tool_file_cmd=$lt_cv_to_tool_file_cmd

# An object symbol dumper.
OBJDUMP=$lt_OBJDUMP

# Method to check whether dependent libraries are shared objects.
deplibs_check_method=$lt_deplibs_check_method

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd=$lt_file_magic_cmd

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=$lt_file_magic_glob

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob=$lt_want_nocaseglob

# DLL creation program.
DLLTOOL=$lt_DLLTOOL

# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd=$lt_sharedlib_from_linklib_cmd

# The archiver.
AR=$lt_AR

# Flags to create an archive.
AR_FLAGS=$lt_AR_FLAGS

# How to feed a file listing to the archiver.
archiver_list_spec=$lt_archiver_list_spec

# A symbol stripping program.
```

```
STRIP=$lt_STRIP

# Commands used to install an old-style archive.
RANLIB=$lt_RANLIB
old_postinstall_cmds=$lt_old_postinstall_cmds
old_postuninstall_cmds=$lt_old_postuninstall_cmds

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=$lock_old_archive_extraction

# A C compiler.
LTCC=$lt_CC

# LTCC compiler flags.
LTCFLAGS=$lt_CFLAGS

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe=$lt_lt_cv_sys_global_symbol_pipe

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl=$lt_lt_cv_sys_global_symbol_to_cdecl

# Transform the output of nm in a C name address pair.
global_symbol_to_c_name_address=$lt_lt_cv_sys_global_symbol_to_c_name_
address

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix=$lt_lt_cv_sys_global_symbol
_to_c_name_address_lib_prefix

# Specify filename containing input files for \${NM}.
nm_file_list_spec=$lt_nm_file_list_spec

# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=$lt_sysroot

# The name of the directory that contains temporary libtool files.
objdir=$objdir

# Used to examine libraries when file_magic_cmd begins with "file".
MAGIC_CMD=$MAGIC_CMD

# Must we lock files when doing compilation?
need_locks=$lt_need_locks

# Manifest tool.
MANIFEST_TOOL=$lt_MANIFEST_TOOL

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
```

```
DSYMUTIL=${lt_DSYMUTIL}

# Tool to change global to local symbols on Mac OS X.
NMEDIT=${lt_NMEDIT}

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=${lt_LIPO}

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=${lt_OTOOL}

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=${lt_OTOOL64}

# Old archive suffix (normally "a").
libext=${libext}

# Shared library suffix (normally ".so").
shrext_cmds=${lt_shrext_cmds}

# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=${lt_extract_expsyms_cmds}

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink=${lt_variables_saved_for_relink}

# Do we need the "lib" prefix for modules?
need_lib_prefix=${need_lib_prefix}

# Do we need a version for libraries?
need_version=${need_version}

# Library versioning type.
version_type=${version_type}

# Shared library runtime path variable.
runpath_var=${runpath_var}

# Shared library path variable.
shlibpath_var=${shlibpath_var}

# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=${shlibpath_overrides_runpath}

# Format of library name prefix.
libname_spec=${lt_libname_spec}

# List of archive names.  First name is the real one, the rest are
links.
```



```
# The last name is the one that the linker finds with -lNAME
library_names_spec=$lt_library_names_spec

# The coded name of the library, if different from the real name.
soname_spec=$lt_soname_spec

# Permission mode override for installation of shared libraries.
install_override_mode=$lt_install_override_mode

# Command to use after installation of a shared archive.
postinstall_cmds=$lt_postinstall_cmds

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=$lt_postuninstall_cmds

# Commands used to finish a libtool library installation in a
directory.
finish_cmds=$lt_finish_cmds

# As "finish_cmds", except a single script fragment to be eval'd but
# not shown.
finish_eval=$lt_finish_eval

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=$hardcode_into_libs

# Compile-time system search path for libraries.
sys_lib_search_path_spec=$lt_sys_lib_search_path_spec

# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec=$lt_sys_lib_dlsearch_path_spec

# Whether dlopen is supported.
dlopen_support=$enable_dlopen

# Whether dlopen of programs is supported.
dlopen_self=$enable_dlopen_self

# Whether dlopen of statically linked programs is supported.
dlopen_self_static=$enable_dlopen_self_static

# Commands to strip libraries.
old_striplib=$lt_old_striplib
striplib=$lt_striplib

# The linker used to build libraries.
LD=$lt_LD

# How to create reloadable object files.
reload_flag=$lt_reload_flag
reload_cmds=$lt_reload_cmds
```

```
# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds

# A language specific compiler.
CC=$lt_compiler

# Is the compiler the GNU compiler?
with_gcc=$GCC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds
archive_expsym_cmds=$lt_archive_expsym_cmds
```

```
# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds
module_expsym_cmds=$lt_module_expsym_cmds

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
```

```

inherit_rpath=$inherit_rpath

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action

# ### END LIBTOOL CONFIG

_LT_EOF

case $host_os in
aix3*)
  cat <<\_LT_EOF >> "$scfgfile"
# AIX sometimes has problems with the GCC collect2 program.  For some
# reason, if we set the COLLECT_NAMES environment variable, the
problems
# vanish in a puff of smoke.
if test "X${COLLECT_NAMES+set}" != Xset; then
  COLLECT_NAMES=
  export COLLECT_NAMES
fi
_LT_EOF
;;
esac

ltmain="$ac_aux_dir/ltmain.sh"

```

```

# We use sed instead of cat because bash on DJGPP gets confused if
# if finds mixed CR/LF and LF-only lines.  Since sed operates in
# text mode, it properly converts lines to CR/LF.  This bash problem
# is reportedly fixed, but why not run on old versions too?
sed '$q' "$ltmain" >> "$cfgfile" \
    || (rm -f "$cfgfile"; exit 1)

if test x"$xsi_shell" = xyes; then
  sed -e '/^func_dirname ()$/,/^{ # func_dirname /c\
func_dirname ()\
{\
\   case ${1} in\
\     */*) func_dirname_result="${1%/*}${2}" ;;\
\     * ) func_dirname_result="${3}" ;;\
\   esac\
} # Extended-shell func_dirname implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_basename ()$/,/^{ # func_basename /c\
func_basename ()\
{\
\   func_basename_result="${1##*/}"\
} # Extended-shell func_basename implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_dirname_and_basename ()$/,/^{ #
func_dirname_and_basename /c\
func_dirname_and_basename ()\
{\
\   case ${1} in\
\     */*) func_dirname_result="${1%/*}${2}" ;;\
\     * ) func_dirname_result="${3}" ;;\
\   esac\
\   func_basename_result="${1##*/}"\
} # Extended-shell func_dirname_and_basename implementation'
"$cfgfile" > $cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_striptime ()$/,/^{ # func_striptime /c\
func_striptime ()\
{\
\
\ # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are\
\ # positional parameters, so assign one to ordinary parameter\
first.\
\ func_striptime_result=${3}\
\ func_striptime_result=${func_striptime_result#"${1}"}\
\ func_striptime_result=${func_striptime_result%"${2}"}\
} # Extended-shell func_striptime implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_split_long_opt ()$/,/^{ # func_split_long_opt /c\
func_split_long_opt ()\
{\
\ func_split_long_opt_name=${1%%=*}\
\ func_split_long_opt_arg=${1#*=}\
} # Extended-shell func_split_long_opt implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_split_short_opt ()$/,/^{ # func_split_short_opt /c\
func_split_short_opt ()\
{\
\ func_split_short_opt_arg=${1#??}\
\ func_split_short_opt_name=${1%"$func_split_short_opt_arg"}\
} # Extended-shell func_split_short_opt implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_lo2o ()$/,/^{ # func_lo2o /c\
func_lo2o ()\
{\
\ case ${1} in\
\   *.lo) func_lo2o_result=${1%.lo}.${objext} ;;\
\   *)   func_lo2o_result=${1} ;;\
\   esac\

```

```

} # Extended-shell func_lo2o implementation' "$cfgfile" > $cfgfile.tmp
\
  && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_xform ()$/,/^{ } # func_xform /c\
func_xform ()\
{\
  func_xform_result=${1%.*}.lo\
} # Extended-shell func_xform implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_arith ()$/,/^{ } # func_arith /c\
func_arith ()\
{\
  func_arith_result=$(( $* ))\
} # Extended-shell func_arith implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_len ()$/,/^{ } # func_len /c\
func_len ()\
{\
  func_len_result=${#1}\
} # Extended-shell func_len implementation' "$cfgfile" > $cfgfile.tmp
\
  && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

fi

if test x"$lt_shell_append" = xyes; then
  sed -e '/^func_append ()$/,/^{ } # func_append /c\
func_append ()\
{\
  eval "${1}+=\\${2}"\
} # Extended-shell func_append implementation' "$cfgfile" >
$cfgfile.tmp \

```

```

    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_append_quoted ()$/ ,/^} # func_append_quoted /c\
func_append_quoted ()\
{\
\    func_quote_for_eval "${2}"\
\    eval "${1}+=\\\ \\\ $func_quote_for_eval_result"\
} # Extended-shell func_append_quoted implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    # Save a `func_append' function call where possible by direct use of
'+='
    sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1+= "%g' $cfgfile >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
    test 0 -eq $? || _lt_function_replace_fail=:
else
    # Save a `func_append' function call even when '+' is not available
    sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1=" $\1%g' $cfgfile >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
    test 0 -eq $? || _lt_function_replace_fail=:
fi

if test x"$ _lt_function_replace_fail" = x:"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: Unable to
substitute extended shell functions in $ofile" >&5
$as_echo "$as_me: WARNING: Unable to substitute extended shell
functions in $ofile" >&2;}
fi

    mv -f "$cfgfile" "$ofile" ||
    (rm -f "$ofile" && cp "$cfgfile" "$ofile" && rm -f "$cfgfile")
    chmod +x "$ofile"

;;

esac

```



```

done # for ac_tag

as_fn_exit 0
_ACEOF
ac_clean_files=$ac_clean_files_save

test $ac_write_fail = 0 ||
  as_fn_error $? "write failure creating $CONFIG_STATUS" "$LINENO" 5

# configure is writing to config.log, and then calls config.status.
# config.status does its own redirection, appending to config.log.
# Unfortunately, on DOS this fails, as config.log is still kept open
# by configure, so config.status won't be able to write to it; its
# output is simply discarded. So we exec the FD to /dev/null,
# effectively closing config.log, so it can be properly (re)opened and
# appended to by config.status. When coming back to configure, we
# need to make the FD available again.
if test "$no_create" != yes; then
  ac_cs_success=:
  ac_config_status_args=
  test "$silent" = yes &&
    ac_config_status_args="$ac_config_status_args --quiet"
  exec 5>/dev/null
  $SHELL $CONFIG_STATUS $ac_config_status_args || ac_cs_success=false
  exec 5>>config.log
  # Use ||, not &&, to avoid exiting from the if with $? = 1, which
  # would make configure fail if this is the last instruction.
  $ac_cs_success || as_fn_exit 1
fi
if test -n "$ac_unrecognized_opts" && test "$enable_option_checking"
!= no; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: unrecognized
options: $ac_unrecognized_opts" >&5
$as_echo "$as_me: WARNING: unrecognized options:
$ac_unrecognized_opts" >&2;}
fi

echo "
          D-BUS GLIB BINDINGS $VERSION
          =====

prefix:                ${prefix}
exec_prefix:           ${exec_prefix}
  libdir:                ${EXPANDED_LIBDIR}
  bindir:                ${EXPANDED_BINDIR}
  sysconfdir:           ${EXPANDED_SYSCONFDIR}
  localstatedir:        ${EXPANDED_LOCALSTATEDIR}
datadir:               ${EXPANDED_DATADIR}
source code location:  ${srcdir}

```

```

    compiler:          ${CC}
    cflags:            ${CFLAGS}
    cppflags:         ${CPPFLAGS}
"
echo "
    Maintainer mode:      ${USE_MAINTAINER_MODE}
    gcc coverage profiling:  ${enable_gcov}
    Building unit tests:   ${enable_tests}
    Building verbose mode:  ${enable_verbose_mode}
    Building assertions:   ${enable_asserts}
    Building checks:       ${enable_checks}
    Building Gtk-doc docs:  ${enable_gtk_doc}
    Bash Completion:      ${enable_bash_completion}
    Using XML parser:      ${with_xml}
    'make check' socket dir:  ${TEST_SOCKET_DIR}
"

if test x$enable_tests = xyes; then
    echo "NOTE: building with unit tests increases the size of the
installed library and renders it insecure."
fi
if test x$enable_tests = xyes -a x$enable_asserts = xno; then
    echo "NOTE: building with unit tests but without assertions
means tests may not properly report failures (this configuration is
only useful when doing something like profiling the tests)"
fi
if test x$enable_gcov = xyes; then
    echo "NOTE: building with coverage profiling is definitely for
developers only."
fi
if test x$enable_verbose_mode = xyes; then
    echo "NOTE: building with verbose mode increases library size,
may slightly increase security risk, and decreases performance."
fi
if test x$enable_asserts = xyes; then
    echo "NOTE: building with assertions increases library size
and decreases performance."
fi
if test x$enable_checks = xno; then
    echo "NOTE: building without checks for arguments passed to
public API makes it harder to debug apps using D-BUS, but will
slightly decrease D-BUS library size and _very_ slightly improve
performance."
fi

File = configure.ac

dnl -*- mode: m4 -*-
AC_PREREQ(2.52)

```

```

AC_INIT([dbus-glib], [0.100.2],
[https://bugs.freedesktop.org/enter_bug.cgi?product=dbus&component=GLi
b])

AC_CANONICAL_HOST

AM_INIT_AUTOMAKE([1.9])

AC_CONFIG_HEADERS(config.h)

# Honor aclocal flags
ACLOCAL="$ACLOCAL $ACLOCAL_FLAGS"
AC_CONFIG_MACRO_DIR(m4)

## must come before we use the $USE_MAINTAINER_MODE variable later
AM_MAINTAINER_MODE

m4_ifdef([AM_SILENT_RULES],[AM_SILENT_RULES([yes])])

# libtool versioning - this applies to libdbus
#
# See
http://sources.redhat.com/autobook/autobook/autobook\_91.html#SEC91 for
details
#

## increment if the interface has additions, changes, removals.
LT_CURRENT=4

## increment any time the source changes; set to
## 0 if you increment CURRENT
LT_REVISION=2

## increment if any interfaces have been added; set to 0
## if any interfaces have been changed or removed. removal has
## precedence over adding, so set to 0 if both happened.
LT_AGE=2

AC_SUBST(LT_CURRENT)
AC_SUBST(LT_REVISION)
AC_SUBST(LT_AGE)

AC_PROG_CC
AC_ISC_POSIX
AC_HEADER_STDC

AC_ARG_ENABLE(tests, AS_HELP_STRING([--enable-tests],[enable unit test
code]),enable_tests=$enableval,enable_tests=$USE_MAINTAINER_MODE)

```

```

AC_ARG_ENABLE(ansi, AS_HELP_STRING([--enable-ansi],[enable -ansi -
pedantic gcc flags]),enable_ansi=$enableval,enable_ansi=no)
AC_ARG_ENABLE(verbose-mode, AS_HELP_STRING([--enable-verbose-
mode],[support verbose debug
mode]),enable_verbose_mode=$enableval,enable_verbose_mode=$USE_MAINTAI
NER_MODE)
AC_ARG_ENABLE(asserts, AS_HELP_STRING([--enable-asserts],[include
assertion
checks]),enable_asserts=$enableval,enable_asserts=$USE_MAINTAINER_MODE
)
AC_ARG_ENABLE(checks, AS_HELP_STRING([--enable-checks],[include sanity
checks on public API]),enable_checks=$enableval,enable_checks=yes)
AC_ARG_ENABLE(gcov, AS_HELP_STRING([--enable-gcov],[compile with
coverage profiling instrumentation (gcc
only)]),enable_gcov=$enableval,enable_gcov=no)
AC_ARG_ENABLE(bash-completion, AS_HELP_STRING([--enable-bash-
completion],[install bash completion
scripts]),enable_bash_completion=$enableval,enable_bash_completion=yes
)
AC_ARG_WITH(test-socket-dir, AS_HELP_STRING([--with-test-socket-
dir=[dirname]], [Where to put sockets for make check]))

```

```

AC_ARG_WITH(introspect-xml, AS_HELP_STRING([--with-introspect-
xml=[filename]], [Pass in a pregenerated dbus daemon introspection xml
file (as generated by 'dbus-daemon --introspect') to use instead of
querying the installed dbus daemon]))

```

```

AM_CONDITIONAL(DBUS_BASH_COMPLETION, test x$enable_bash_completion =
xyes)
if test x$enable_bash_completion = xyes; then
  AC_DEFINE(DBUS_BASH_COMPLETION,1,[Enable bash completion])
fi

```

```

if test x$enable_verbose_mode = xyes; then
  AC_DEFINE(DBUS_ENABLE_VERBOSE_MODE,1,[Support a verbose mode])
fi

```

```

AC_ARG_WITH(dbus-binding-tool,
  AS_HELP_STRING([--with-dbus-binding-tool=[filename]], [Use external
dbus-binding-tool program]),

```

```

[DBUS_BINDING_TOOL=$withval],[DBUS_BINDING_TOOL=\$\\(top_builddir\) /dbu
s/dbus-binding-tool])
AC_SUBST(DBUS_BINDING_TOOL)

```

```

dnl DBUS_BUILD_TESTS controls unit tests built in to .c files
dnl and also some stuff in the test/ subdir
AM_CONDITIONAL(DBUS_BUILD_TESTS, test x$enable_tests = xyes)
if test x$enable_tests = xyes; then

```

```

        AC_DEFINE(DBUS_BUILD_TESTS,1,[Build test code])
    fi

    if test x$enable_verbose_mode = xyes; then
        AC_DEFINE(DBUS_ENABLE_VERBOSE_MODE,1,[Support a verbose mode])
    fi
    if test x$enable_asserts = xno; then
        AC_DEFINE(DBUS_DISABLE_ASSERT,1,[Disable assertion checking])
        AC_DEFINE(G_DISABLE_ASSERT,1,[Disable GLib assertion macros])
    fi
    if test x$enable_checks = xno; then
        AC_DEFINE(DBUS_DISABLE_CHECKS,1,[Disable public API sanity
checking])
        AC_DEFINE(G_DISABLE_CHECKS,1,[Disable GLib public API sanity
checking])
    fi

#### gcc warning flags

AC_DEFUN([AC_CC_TRY_FLAG], [
    AC_MSG_CHECKING([whether gcc understands $1])

    ac_save_CFLAGS="$CFLAGS"
    CFLAGS="$CFLAGS $1"

    AC_COMPILE_IFELSE([ ], [ac_cc_flag=yes], [ac_cc_flag=no])
    CFLAGS="$ac_save_CFLAGS"

    if test "x$ac_cc_flag" = "xyes"; then
        ifelse([$2], , :, [$2])
    else
        ifelse([$3], , :, [$3])
    fi
    AC_MSG_RESULT([$ac_cc_flag])
])

if test "x$GCC" = "xyes"; then
    AC_CC_TRY_FLAG([-Wfloat-equal],
                    [ac_flag_float_equal=yes],
                    [ac_flag_float_equal=no])

    changequote(, )dnl
    case " $CFLAGS " in
    *[\ \ ]-Wall[\ \ ]*) ;;
    *) CFLAGS="$CFLAGS -Wall" ;;
    esac

    case " $CFLAGS " in
    *[\ \ ]-Wchar-subscripts[\ \ ]*) ;;
    *) CFLAGS="$CFLAGS -Wchar-subscripts" ;;
    esac

```

```

case " $CFLAGS " in
*[\ \ ]-Wmissing-declarations[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wmissing-declarations" ;;
esac

case " $CFLAGS " in
*[\ \ ]-Wmissing-prototypes[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wmissing-prototypes" ;;
esac

case " $CFLAGS " in
*[\ \ ]-Wnested-externs[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wnested-externs" ;;
esac

case " $CFLAGS " in
*[\ \ ]-Wpointer-arith[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wpointer-arith" ;;
esac

case " $CFLAGS " in
*[\ \ ]-Wcast-align[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wcast-align" ;;
esac

if test "x$ac_flag_float_equal" = "xyes"; then
  case " $CFLAGS " in
  *[\ \ ]-Wfloat-equal[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -Wfloat-equal" ;;
  esac
fi

case " $CFLAGS " in
*[\ \ ]-Wsign-compare[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -Wsign-compare" ;;
esac

# This one is special - it's not a warning override.
# http://bugs.freedesktop.org/show\_bug.cgi?id=10599
# is the bug for DBus core.
case " $CFLAGS " in
*[\ \ ]-fno-strict-aliasing[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -fno-strict-aliasing" ;;
esac

if test "x$enable_ansi" = "xyes"; then
  case " $CFLAGS " in
  *[\ \ ]-ansi[\ \ ]*) ;;
  *) CFLAGS="$CFLAGS -ansi" ;;
  esac

  case " $CFLAGS " in

```

```

*[\ \ ]-D_POSIX_C_SOURCE*) ;;
*) CFLAGS="$CFLAGS -D_POSIX_C_SOURCE=199309L" ;;
esac

case " $CFLAGS " in
*[\ \ ]-D_BSD_SOURCE[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -D_BSD_SOURCE" ;;
esac

case " $CFLAGS " in
*[\ \ ]-pedantic[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -pedantic" ;;
esac
fi
if test x$enable_gcov = xyes; then
  case " $CFLAGS " in
*[\ \ ]-fprofile-arcs[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -fprofile-arcs" ;;
esac
  case " $CFLAGS " in
*[\ \ ]-ftest-coverage[\ \ ]*) ;;
*) CFLAGS="$CFLAGS -ftest-coverage" ;;
esac

  ## remove optimization
  CFLAGS=`echo "$CFLAGS" | sed -e 's/-O[0-9]*//g'`
fi
changequote(,[,])dnl
else
  if test x$enable_gcov = xyes; then
    AC_MSG_ERROR([--enable-gcov can only be used with gcc])
  fi
fi
fi

AM_PROG_LIBTOOL

changequote(,)dnl
# compress spaces in flags
CFLAGS=`echo "$CFLAGS" | sed -e 's/ +/ /g'`
CPPFLAGS=`echo "$CPPFLAGS" | sed -e 's/ +/ /g'`
changequote(,[,])dnl

if test x$enable_gcov = xyes; then
  # so that config.h changes when you toggle gcov support
  AC_DEFINE_UNQUOTED([DBUS_GCOV_ENABLED],
    [__GNUC__ * 10000 + __GNUC_MINOR__ * 100 +
__GNUC_PATCHLEVEL__],
    [Defined to the gcc version if gcov is enabled, to force a
rebuild due to config.h changing])
fi

#### Various functions

```

```

AC_SEARCH_LIBS([socket], [socket])

dnl check for socklen_t
AC_MSG_CHECKING(whether socklen_t is defined)
AC_TRY_COMPILE([
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
],[
socklen_t foo;
foo = 1;
],dbus_have_socklen_t=yes,dbus_have_socklen_t=no)
AC_MSG_RESULT($dbus_have_socklen_t)

if test "x$dbus_have_socklen_t" = "xyes"; then
    AC_DEFINE(HAVE_SOCKLEN_T,1,[Have socklen_t type])
fi

#### Abstract sockets

AC_LANG_PUSH(C)
AC_CACHE_CHECK([abstract socket namespace],
                ac_cv_have_abstract_sockets,
                [AC_RUN_IFELSE([AC_LANG_PROGRAM(
[[
#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <errno.h>
]],
[[
int listen_fd;
struct sockaddr_un addr;

listen_fd = socket (PF_UNIX, SOCK_STREAM, 0);

if (listen_fd < 0)
{
    fprintf (stderr, "socket() failed: %s\n", strerror (errno));
    exit (1);
}

memset (&addr, '\0', sizeof (addr));
addr.sun_family = AF_UNIX;
strcpy (addr.sun_path, "X/tmp/dbus-fake-socket-path-used-in-
configure-test");
addr.sun_path[0] = '\0'; /* this is what makes it abstract */

if (bind (listen_fd, (struct sockaddr*) &addr, SUN_LEN (&addr)) < 0)

```



```

        {
            fprintf (stderr, "Abstract socket namespace bind() failed:
%s\n",
                    strerror (errno));
            exit (1);
        }
        else
            exit (0);
    ]]],
        [ac_cv_have_abstract_sockets=yes],
        [ac_cv_have_abstract_sockets=no]
    ))
AC_LANG_POP(C)

if test x$enable_abstract_sockets = xyes; then
    if test x$ac_cv_have_abstract_sockets = xno; then
        AC_MSG_ERROR([Abstract sockets explicitly required, and support
not detected.])
    fi
fi

if test x$enable_abstract_sockets = xno; then
    ac_cv_have_abstract_sockets=no;
fi

if test x$ac_cv_have_abstract_sockets = xyes ; then
    DBUS_PATH_OR_ABSTRACT=abstract
    AC_DEFINE(HAVE_ABSTRACT_SOCKETS,1,[Have abstract socket namespace])
else
    DBUS_PATH_OR_ABSTRACT=path
fi

# this is used in addresses to prefer abstract, e.g.
# unix:path=/foo or unix:abstract=/foo
AC_SUBST(DBUS_PATH_OR_ABSTRACT)

#### Sort out XML library

# see what we have
AC_CHECK_LIB(expat, XML_ParserCreate_MM,
             [ AC_CHECK_HEADERS(expat.h, have_expat=true,
have_expat=false) ],
             have_expat=false)

if ! $have_expat ; then
    AC_MSG_ERROR([expat library not found, check config.log for failed
attempts])
fi

XML_LIBS=-lexpat
XML_CFLAGS=

```

```

#### Set up final flags

PKG_CHECK_MODULES([DBUS], [dbus-1 >= 1.2.16])
AC_SUBST([DBUS_CFLAGS])
AC_SUBST([DBUS_LIBS])

# Glib detection
PKG_CHECK_MODULES([DBUS_GLIB], [gobject-2.0 >= 2.26, gio-2.0 >= 2.26])
PKG_CHECK_MODULES(DBUS_GLIB_THREADS, gthread-2.0 >= 2.6,
have_glib_threads=yes, have_glib_threads=no)

AM_CONDITIONAL(HAVE_GLIB_THREADS, test x$have_glib_threads = xyes)

GLIB_GENMARSHAL=`$PKG_CONFIG --variable=glib_genmarshal glib-2.0`
AC_SUBST(GLIB_GENMARSHAL)

dnl GLib flags
AC_SUBST(DBUS_GLIB_CFLAGS)
AC_SUBST(DBUS_GLIB_LIBS)
AC_SUBST(DBUS_GLIB_THREADS_LIBS)

DBUS_GLIB_TOOL_CFLAGS=$XML_CFLAGS
DBUS_GLIB_TOOL_LIBS="$XML_LIBS"
AC_SUBST(DBUS_GLIB_TOOL_CFLAGS)
AC_SUBST(DBUS_GLIB_TOOL_LIBS)

### gtk-doc Documentation
GTK_DOC_CHECK(1.4)

#### Have to go $localstatedir->$prefix/var->/usr/local/var
#### someone please fix this a better way...

#### find the actual value for $prefix that we'll end up with
## (I know this is broken and should be done in the Makefile, but
## that's a major pain and almost nobody actually seems to care)
REAL_PREFIX=
if test "x$prefix" = "xNONE"; then
    REAL_PREFIX=$ac_default_prefix
else
    REAL_PREFIX=$prefix
fi

## temporarily change prefix and exec_prefix
old_prefix=$prefix
prefix=$REAL_PREFIX

if test "x$exec_prefix" = xNONE ; then
    REAL_EXEC_PREFIX=$REAL_PREFIX
else
    REAL_EXEC_PREFIX=$exec_prefix
fi

```

```

old_exec_prefix=$exec_prefix
exec_prefix=$REAL_EXEC_PREFIX

## eval everything
LOCALSTATEDIR_TMP="$localstatedir"
EXPANDED_LOCALSTATEDIR=`eval echo $LOCALSTATEDIR_TMP`
AC_SUBST(EXPANDED_LOCALSTATEDIR)

SYSCONFDIR_TMP="$sysconfdir"
EXPANDED_SYSCONFDIR=`eval echo $SYSCONFDIR_TMP`
AC_SUBST(EXPANDED_SYSCONFDIR)

BINDIR_TMP="$bindir"
EXPANDED_BINDIR=`eval echo $BINDIR_TMP`
AC_SUBST(EXPANDED_BINDIR)

LIBDIR_TMP="$libdir"
EXPANDED_LIBDIR=`eval echo $LIBDIR_TMP`
AC_SUBST(EXPANDED_LIBDIR)

DATADIR_TMP="$datadir"
EXPANDED_DATADIR=`eval echo $DATADIR_TMP`
AC_SUBST(EXPANDED_DATADIR)

## put prefix and exec_prefix back
prefix=$old_prefix
exec_prefix=$old_exec_prefix

#### Tell tests where to find certain stuff in builddir
ABSOLUTE_TOP_BUILDDIR=`cd ${ac_top_builddir}. && pwd`

AC_DEFUN([TEST_PATH], [
TEST_$1=${ABSOLUTE_TOP_BUILDDIR}/test/$2
AC_DEFINE_UNQUOTED(TEST_$1, "$TEST_$1",
                    [Full path to test file test/$2 in builddir])
AC_SUBST(TEST_$1)
])

TEST_PATH(SERVICE_DIR, data/valid-service-files)
TEST_PATH(SERVICE_BINARY, test-service)
TEST_PATH(SHELL_SERVICE_BINARY, test-shell-service)
TEST_PATH(CORE_SERVICE_BINARY, core/test-service-glib)
TEST_PATH(INTERFACES_SERVICE_BINARY, interfaces/test-service)
TEST_PATH(EXIT_BINARY, test-exit)
TEST_PATH(SEGFAULT_BINARY, test-segfault)
TEST_PATH(SLEEP_FOREVER_BINARY, test-sleep-forever)
AC_SUBST(ABSOLUTE_TOP_BUILDDIR)

if ! test -z "$with_test_socket_dir" ; then
    TEST_SOCKET_DIR="$with_test_socket_dir"
else
    TEST_SOCKET_DIR=$DEFAULT_SOCKET_DIR

```

```

fi
AC_SUBST(TEST_SOCKET_DIR)
AC_DEFINE_UNQUOTED(DBUS_TEST_SOCKET_DIR, "$TEST_SOCKET_DIR", [Where to
put test sockets])

AC_OUTPUT([
Makefile
m4/Makefile
doc/Makefile
doc/reference/Makefile
doc/reference/version.xml
dbus/Makefile
dbus/examples/Makefile
dbus/examples/statemachine/Makefile
test/Makefile
test/core/Makefile
test/interfaces/Makefile
test/data/valid-service-files/debug-glib.service
test/data/valid-service-files/debug-echo.service
test/data/valid-service-files/interfaces-test.service
test/lib/Makefile
test/manual/Makefile
tools/Makefile
dbus-glib-1.pc
dbus-glib-1-uninstalled.pc
])

dnl
=====
====
echo "
                D-BUS GLIB BINDINGS $VERSION
                =====

prefix:                ${prefix}
exec_prefix:           ${exec_prefix}
  libdir:                ${EXPANDED_LIBDIR}
  bindir:                ${EXPANDED_BINDIR}
  sysconfdir:           ${EXPANDED_SYSCONFDIR}
  localstatedir:        ${EXPANDED_LOCALSTATEDIR}
datadir:               ${EXPANDED_DATADIR}
source code location:  ${srcdir}
compiler:              ${CC}
cflags:                ${CFLAGS}
cppflags:              ${CPPFLAGS}
"

echo "
  Maintainer mode:      ${USE_MAINTAINER_MODE}
  gcc coverage profiling:  ${enable_gcov}
  Building unit tests:   ${enable_tests}
  Building verbose mode:  ${enable_verbose_mode}

```

```

Building assertions:      ${enable_asserts}
Building checks:         ${enable_checks}
Building Gtk-doc docs:   ${enable_gtk_doc}
Bash Completion:        ${enable_bash_completion}
Using XML parser:        ${with_xml}
'make check' socket dir: ${TEST_SOCKET_DIR}
"

if test x$enable_tests = xyes; then
    echo "NOTE: building with unit tests increases the size of the
installed library and renders it insecure."
fi
if test x$enable_tests = xyes -a x$enable_asserts = xno; then
    echo "NOTE: building with unit tests but without assertions
means tests may not properly report failures (this configuration is
only useful when doing something like profiling the tests)"
fi
if test x$enable_gcov = xyes; then
    echo "NOTE: building with coverage profiling is definitely for
developers only."
fi
if test x$enable_verbose_mode = xyes; then
    echo "NOTE: building with verbose mode increases library size,
may slightly increase security risk, and decreases performance."
fi
if test x$enable_asserts = xyes; then
    echo "NOTE: building with assertions increases library size
and decreases performance."
fi
if test x$enable_checks = xno; then
    echo "NOTE: building without checks for arguments passed to
public API makes it harder to debug apps using D-BUS, but will
slightly decrease D-BUS library size and _very_ slightly improve
performance."
fi

```

File = configure.ac~1~

```

dnl -*- mode: m4 -*-

```

```

AC_PREREQ([2.63])

```

```

m4_define([dbus_major_version], [1])

```

```

m4_define([dbus_minor_version], [6])

```

```

m4_define([dbus_micro_version], [8])

```

```

m4_define([dbus_version],

```

```

    [dbus_major_version.dbus_minor_version.dbus_micro_version])

```

```

AC_INIT([dbus], [dbus_version], [https://bugs.freedesktop.org/enter_bug.
cgi?product=dbus], [dbus])

```

```

AC_CANONICAL_HOST

```

```

AC_CONFIG_HEADERS([config.h])
AC_CONFIG_MACRO_DIR([m4])

AM_INIT_AUTOMAKE([1.10 tar-ustar -Wno-portability])

GETTEXT_PACKAGE=dbus-1
AC_SUBST(GETTEXT_PACKAGE)
AC_DEFINE_UNQUOTED(GETTEXT_PACKAGE,"$GETTEXT_PACKAGE",[The name of the
gettext domain])

# By default, rebuild autotools files on demand; only use ./missing if
the
# user says --disable-maintainer-mode (some distributions like to do
this)
AM_MAINTAINER_MODE([enable])

m4_ifdef([AM_SILENT_RULES],[AM_SILENT_RULES([yes])])

AC_DEFINE_UNQUOTED(DBUS_DAEMON_NAME,"dbus-daemon",[Name of
executable])

# libtool versioning - this applies to libdbus
#
# See
http://sources.redhat.com/autobook/autobook/autobook\_91.html#SEC91 for
details
#

## increment if the interface has additions, changes, removals.
LT_CURRENT=10

## increment any time the source changes; set to
## 0 if you increment CURRENT
LT_REVISION=2

## increment if any interfaces have been added; set to 0
## if any interfaces have been changed or removed. removal has
## precedence over adding, so set to 0 if both happened.
LT_AGE=7

AC_SUBST(LT_CURRENT)
AC_SUBST(LT_REVISION)
AC_SUBST(LT_AGE)

DBUS_MAJOR_VERSION=dbus_major_version
DBUS_MINOR_VERSION=dbus_minor_version
DBUS_MICRO_VERSION=dbus_micro_version
DBUS_VERSION=dbus_major_version.dbus_minor_version.dbus_micro_version

AC_SUBST(DBUS_MAJOR_VERSION)
AC_SUBST(DBUS_MINOR_VERSION)

```

```

AC_SUBST(DBUS_MICRO_VERSION)
AC_SUBST(DBUS_VERSION)

AC_PROG_CC
AM_PROG_CC_C_O
AC_PROG_CXX
AC_USE_SYSTEM_EXTENSIONS
AC_ISC_POSIX
AC_HEADER_STDC
AC_C_INLINE
AM_PROG_LIBTOOL
AC_PROG_MKDIR_P
COMPILER_COVERAGE
COMPILER_OPTIMISATIONS
PKG_PROG_PKG_CONFIG

# Initialize libtool
LT_INIT([win32-dll])
LT_LANG([Windows Resource])

# Set some internal variables depending on the platform for later use.
dbus_win=no
dbus_cygwin=no
dbus_unix=no
case "${host}" in
  *-mingw32ce*)
    dbus_win=yes
    dbus_wince=yes
    ;;
  *-mingw32*)
    dbus_win=yes
    ;;
  *-cygwin*)
    dbus_cygwin=yes
    dbus_unix=yes
    ;;
  *)
    dbus_unix=yes
    ;;
esac

# Special defines for certain platforms
if test "$dbus_win" = yes; then
  AC_DEFINE(DBUS_WIN,1,[Defined if we run on a W32 API based
system])
  BUILD_TIMESTAMP=`date --iso-8601=minutes`
  AC_SUBST(BUILD_TIMESTAMP)
  # Assume DBUS_VERSION is always three numbers
  BUILD_FILEVERSION=`echo "$DBUS_VERSION" | sed -e 's/\./,/g'`,0
  AC_SUBST(BUILD_FILEVERSION)
  AC_CHECK_TOOL(WINDRES, windres, no)
  if test "$WINDRES" = no; then

```

```

        AC_MSG_ERROR([*** Could not find an implementation of windres in
your PATH.])
    fi
    if test "$dbus_wince" = yes; then
        AC_DEFINE(DBUS_WINCE,1,[Defined if we run on a W32 CE API based
system])
        AC_DEFINE(_WIN32_WCE, 0x0502, [Defined to get newer W32 CE
APIs])
    fi
else
    AC_DEFINE(DBUS_UNIX,1,[Defined if we run on a Unix-based system])
fi
if test "$dbus_cygwin" = yes; then
    AC_DEFINE(DBUS_CYGWIN,1,[Defined if we run on a cygwin API based
system])
fi

AM_CONDITIONAL(DBUS_WIN, test "$dbus_win" = yes)
AM_CONDITIONAL(DBUS_WINCE, test "$dbus_wince" = yes)
AM_CONDITIONAL(DBUS_UNIX, test "$dbus_unix" = yes)
AM_CONDITIONAL(DBUS_CYGWIN, test "$dbus_cygwin" = yes)

# this must come first: other options use this to set their defaults
AC_ARG_ENABLE([developer],
    [AS_HELP_STRING([--enable-developer],
    [set defaults to be appropriate for a D-Bus developer instead of a
distribution/end-user])],
    [],
    [enable_developer=no])

DBUS_STATIC_BUILD_CPPFLAGS=
if test "x$enable_shared" = xno; then
    # On Windows, linking against the static library requires special
    effort
    # to turn off DLL import/export processing. We normally link some
    things
    # against the dynamic library, but if we're not building that,
    we'll
    # have to link everything statically.
    DBUS_STATIC_BUILD_CPPFLAGS=-DDBUS_STATIC_BUILD
fi
AC_SUBST([DBUS_STATIC_BUILD_CPPFLAGS])

AC_ARG_ENABLE(ansi, AS_HELP_STRING([--enable-ansi],[enable -ansi -
pedantic gcc flags]),enable_ansi=$enableval,enable_ansi=no)
AC_ARG_ENABLE(verbose-mode, AS_HELP_STRING([--enable-verbose-
mode],[support verbose debug
mode]),enable_verbose_mode=$enableval,enable_verbose_mode=$enable_deve
loper)
AC_ARG_ENABLE(asserts, AS_HELP_STRING([--enable-asserts],[include
assertion
checks]),enable_asserts=$enableval,enable_asserts=$enable_developer)

```



```

AC_ARG_ENABLE(checks, AS_HELP_STRING([--enable-checks],[include sanity
checks on public API]),enable_checks=$enableval,enable_checks=yes)
AC_ARG_ENABLE(xml-docs, AS_HELP_STRING([--enable-xml-docs],[build XML
documentation (requires
xmlto)]),enable_xml_docs=$enableval,enable_xml_docs=auto)
AC_ARG_ENABLE(doxygen-docs, AS_HELP_STRING([--enable-doxygen-
docs],[build DOXYGEN documentation (requires
Doxygen)]),enable_doxygen_docs=$enableval,enable_doxygen_docs=auto)
AC_ARG_ENABLE(abstract-sockets, AS_HELP_STRING([--enable-abstract-
sockets],[use abstract socket namespace (linux
only)]),enable_abstract_sockets=$enableval,enable_abstract_sockets=aut
o)
AC_ARG_ENABLE(selinux, AS_HELP_STRING([--enable-selinux],[build with
SELinux support]),enable_selinux=$enableval,enable_selinux=auto)
AC_ARG_ENABLE(libaudit,AS_HELP_STRING([--enable-libaudit],[build audit
daemon support for
SELinux]),enable_libaudit=$enableval,enable_libaudit=auto)
AC_ARG_ENABLE(dnotify, AS_HELP_STRING([--enable-dnotify],[build with
dnotify support (linux
only)]),enable_dnotify=$enableval,enable_dnotify=auto)
AC_ARG_ENABLE(inotify, AS_HELP_STRING([--enable-inotify],[build with
inotify support (linux
only)]),enable_inotify=$enableval,enable_inotify=auto)
AC_ARG_ENABLE(kqueue, AS_HELP_STRING([--enable-kqueue],[build with
kqueue support]),enable_kqueue=$enableval,enable_kqueue=auto)
AC_ARG_ENABLE(console-owner-file, AS_HELP_STRING([--enable-console-
owner-file],[enable console owner
file]),enable_console_owner_file=$enableval,enable_console_owner_file=
auto)
AC_ARG_ENABLE(userdb-cache, AS_HELP_STRING([--enable-userdb-
cache],[build with userdb-cache
support]),enable_userdb_cache=$enableval,enable_userdb_cache=yes)
AC_ARG_ENABLE(launchd, AS_HELP_STRING([--enable-launchd],[build with
launchd auto-launch
support]),enable_launchd=$enableval,enable_launchd=auto)
AC_ARG_ENABLE(systemd, AS_HELP_STRING([--enable-systemd],[build with
systemd at_console
support]),enable_systemd=$enableval,enable_systemd=auto)

AC_ARG_WITH(xml, AS_HELP_STRING([--with-xml=[libxml/expat]], [XML
library to use (libxml may be named libxml2 on some systems)])
AC_ARG_WITH(init-scripts, AS_HELP_STRING([--with-init-
scripts=[redhat]], [Style of init scripts to install]))
AC_ARG_WITH(session-socket-dir, AS_HELP_STRING([--with-session-socket-
dir=[dirname]], [Where to put sockets for the per-login-session message
bus]))
AC_ARG_WITH(test-socket-dir, AS_HELP_STRING([--with-test-socket-
dir=[dirname]], [Where to put sockets for make check]))
AC_ARG_WITH(system-pid-file, AS_HELP_STRING([--with-system-pid-
file=[pidfile]], [PID file for systemwide daemon]))
AC_ARG_WITH(system-socket, AS_HELP_STRING([--with-system-
socket=[filename]], [UNIX domain socket for systemwide daemon]))

```

```

AC_ARG_WITH(console-auth-dir, AS_HELP_STRING([--with-console-auth-
dir=[dirname]], [directory to check for console ownership]))
AC_ARG_WITH(console-owner-file, AS_HELP_STRING([--with-console-owner-
file=[filename]], [file whose owner determines current console owner]))
AC_ARG_WITH(launchd-agent-dir, AS_HELP_STRING([--with-launchd-agent-
dir=[dirname]], [directory to put the launchd agent (default:
/Library/LaunchAgents)]))
AC_ARG_WITH(dbus-user, AS_HELP_STRING([--with-dbus-user=<user>], [User
for running the DBUS daemon (messagebus)]))
AC_ARG_WITH(dbus-daemon-dir, AS_HELP_STRING([--with-dbus-
daemon-dir=[dirname]], [Directory for installing the DBUS daemon]))
AC_ARG_WITH(dbus-session-bus-default-address, AS_HELP_STRING([--with-
dbus-session-bus-default-address=[nonce-
tcp:/autolaunch:/tcp:host:port]], [Transport Type to be used (default:
nonce-
tcp:)]), with_dbus_session_bus_default_address=$withval, with_dbus_sessi
on_bus_default_address=nonce-tcp:)

AC_ARG_ENABLE([embedded-tests],
  AS_HELP_STRING([--enable-embedded-tests],
    [enable unit test code in the library and binaries]),
  [], [enable_embedded_tests=$enable_developer])
AC_ARG_ENABLE([modular-tests],
  AS_HELP_STRING([--enable-modular-tests],
    [enable modular regression tests (requires GLib)]),
  [], [enable_modular_tests=auto])
# --enable-tests overrides both --enable-embedded-tests and
# --enable-modular-tests
AC_ARG_ENABLE([tests],
  AS_HELP_STRING([--enable-tests],
    [enable/disable all tests, overriding embedded-tests/modular-
tests]),
  [
    if test "x$enableval" = xyes; then
      AC_MSG_NOTICE([Full test coverage was requested with --enable-
tests=yes])
      AC_MSG_NOTICE([This has many dependencies (GLib, dbus-glib,
Python)])
    fi
    enable_embedded_tests=$enableval
    enable_modular_tests=$enableval
  ],
  [])

# DBUS_ENABLE_EMBEDDED_TESTS controls unit tests built in to .c files
# and also some stuff in the test/ subdir. DBUS_BUILD_TESTS was an
older
# name for this.
AM_CONDITIONAL([DBUS_BUILD_TESTS], [test "x$enable_embedded_tests" =
xyes])
AM_CONDITIONAL([DBUS_ENABLE_EMBEDDED_TESTS],
  [test "x$enable_embedded_tests" = xyes])

```

```

if test "x$enable_embedded_tests" = xyes; then
    AC_DEFINE([DBUS_ENABLE_EMBEDDED_TESTS], [1],
        [Define to build test code into the library and binaries])
    AC_DEFINE([DBUS_BUILD_TESTS], [1],
        [Define to build test code into the library and binaries])
fi

# DBUS_ENABLE_MODULAR_TESTS controls tests that work based on public
API.
# These use GTest, from GLib, because life's too short. They're
enabled by
# default (unless you don't have GLib), because they don't bloat the
library
# or binaries.

with_glib=yes

if test "x$enable_modular_tests" != xno; then
    PKG_CHECK_MODULES([GLIB], [glib-2.0 >= 2.24, gio-2.0 >= 2.24],
        [],
        [if test "x$enable_modular_tests" = xyes; then
            AC_MSG_NOTICE([Full test coverage (--enable-modular-tests=yes or
--enable-tests=yes) requires GLib])
            AC_MSG_ERROR([$GLIB_ERRORS])
            else # assumed to be "auto"
                with_glib=no
            fi])
    # If dbus-gmain.[ch] returned to libdbus then we wouldn't need this
    PKG_CHECK_MODULES([DBUS_GLIB], [dbus-glib-1],
        [],
        [if test "x$enable_modular_tests" = xyes; then
            AC_MSG_NOTICE([Full test coverage (--enable-modular-tests=yes or
--enable-tests=yes) requires dbus-glib])
            AC_MSG_ERROR([$DBUS_GLIB_ERRORS])
            else # assumed to be "auto"
                with_glib=no
            fi])
    fi
if test "x$enable_modular_tests" != xno; then
    AC_DEFINE([DBUS_ENABLE_MODULAR_TESTS], [1],
        [Define to build independent test binaries])
fi
AM_CONDITIONAL([DBUS_ENABLE_MODULAR_TESTS],
    [test "x$enable_modular_tests" != xno])

if test "x$with_glib" != xno; then
    AC_DEFINE([DBUS_WITH_GLIB], [1],
        [Define if GLib, GObject, GIO are available])
fi
AM_CONDITIONAL([DBUS_WITH_GLIB], [test "x$with_glib" != xno])

AC_ARG_ENABLE([installed-tests],

```

```

AS_HELP_STRING([--enable-installed-tests],
  [enable unit test code in the library and binaries]),
  [], [enable_installed_tests=no])
AM_CONDITIONAL([DBUS_ENABLE_INSTALLED_TESTS],
  [test "x$enable_installed_tests" = xyes])

if test "x$enable_tests" = xyes; then
  # full test coverage is required, Python is a hard dependency
  AC_MSG_NOTICE([Full test coverage (--enable-tests=yes) requires
Python, dbus-python, pygobject])
  AM_PATH_PYTHON([2.6])
  AC_MSG_CHECKING([for Python modules for full test coverage])
  if "$PYTHON" -c "import dbus, gobject, dbus.mainloop.glib"; then
    AC_MSG_RESULT([yes])
  else
    AC_MSG_RESULT([no])
    AC_MSG_ERROR([cannot import dbus, gobject, dbus.mainloop.glib
Python modules])
  fi
else
  # --enable-tests not given: do not abort if Python is missing
  AM_PATH_PYTHON([2.6], [], [:])
fi

if test x$enable_verbose_mode = xyes; then
  AC_DEFINE(DBUS_ENABLE_VERBOSE_MODE,1,[Support a verbose mode])
fi

if test x$enable_asserts = xno; then
  AC_DEFINE(DBUS_DISABLE_ASSERT,1,[Disable assertion checking])
  DISABLE_UNUSED_WARNINGS="unused-label"
  R_DYNAMIC_LDFLAG=""
else
  # -rdynamic is needed for glibc's backtrace_symbols to work.
  # No clue how much overhead this adds, but it's useful
  # to do this on any assertion failure,
  # so for now it's enabled anytime asserts are (currently not
  # in production builds).

  # To get -rdynamic you pass -export-dynamic to libtool.
  AC_DEFINE(DBUS_BUILT_R_DYNAMIC,1,[whether -export-dynamic was
passed to libtool])
  R_DYNAMIC_LDFLAG=-export-dynamic
fi
AC_SUBST(R_DYNAMIC_LDFLAG)

if test x$enable_checks = xno; then
  AC_DEFINE(DBUS_DISABLE_CHECKS,1,[Disable public API sanity
checking])
  AC_DEFINE(G_DISABLE_CHECKS,1,[Disable GLib public API sanity
checking])
  DISABLE_UNUSED_WARNINGS="unused-label"

```

```

fi

if test x$enable_userdb_cache = xyes; then
    AC_DEFINE(DBUS_ENABLE_USERDB_CACHE,1,[Build with caching of user
data])
fi

if test x$enable_compiler_coverage = xyes; then
    ## so that config.h changes when you toggle gcov support
    AC_DEFINE_UNQUOTED(DBUS_GCOV_ENABLED, 1, [Defined if gcov is
enabled to force a rebuild due to config.h changing])
fi

# glibc21.m4 serial 3
dnl Copyright (C) 2000-2002, 2004 Free Software Foundation, Inc.
dnl This file is free software; the Free Software Foundation
dnl gives unlimited permission to copy and/or distribute it,
dnl with or without modifications, as long as this notice is
preserved.

# Test for the GNU C Library, version 2.1 or newer.
# From Bruno Haible.

AC_CACHE_CHECK(whether we are using the GNU C Library 2.1 or newer,
ac_cv_gnu_library_2_1,
[AC_EGREP_CPP([Lucky GNU user],
[
#include <features.h>
#ifdef __GNU_LIBRARY__
#if ( __GLIBC__ == 2 && __GLIBC_MINOR__ >= 1) || ( __GLIBC__ > 2)
Lucky GNU user
#endif
#endif
],
ac_cv_gnu_library_2_1=yes,
ac_cv_gnu_library_2_1=no)
])

#### Integer sizes

AC_CHECK_SIZEOF(char)
AC_CHECK_SIZEOF(short)
AC_CHECK_SIZEOF(long)
AC_CHECK_SIZEOF(int)
AC_CHECK_SIZEOF(void *)
AC_CHECK_SIZEOF(long long)
AC_CHECK_SIZEOF(__int64)

AC_ARG_WITH([64-bit],
[AS_HELP_STRING([--without-64-bit],
[If you have to use this option, please report it as a bug])],

```

```

[],
[with_64_bit=yes])

### See what our 64 bit type is called
AC_MSG_CHECKING([64-bit integer type])

case 8 in
$ac_cv_sizeof_int)
  dbusint64=int
  dbusint64_constant='(val) '
  dbusuint64_constant='(val) '
  dbusint64_printf_modifier='""'
  ;;
$ac_cv_sizeof_long)
  dbusint64=long
  dbusint64_constant='(val##L) '
  dbusuint64_constant='(val##UL) '
  dbusint64_printf_modifier='"l"'
  ;;
$ac_cv_sizeof_long_long)
  dbusint64='long long'
  dbusint64_constant='(val##LL) '
  dbusuint64_constant='(val##ULL) '
  # Ideally we discover what the format is, but this is
  # only used in verbose mode, so eh...
  if test x"$ac_cv_gnu_library_2_1" = xyes; then
    dbusint64_printf_modifier='"ll"'
  fi
  ;;
$ac_cv_sizeof__int64)
  dbusint64=__int64
  dbusint64_constant='(val##i64) '
  dbusuint64_constant='(val##ui64) '
  # See above case
  if test x"$ac_cv_gnu_library_2_1" = xyes; then
    dbusint64_printf_modifier='"ll"'
  fi
  ;;
esac

AS_IF(
  [test "x$with_64_bit" = xno],
  [
    DBUS_INT64_TYPE="no_int64_type_detected"
    DBUS_HAVE_INT64=0
    DBUS_INT64_CONSTANT=
    DBUS_UINT64_CONSTANT=
    AC_MSG_RESULT([disabled via --without-64-bit])
  ],
  [
    dn1 else if
    [test -z "$dbusint64"],
    [AC_MSG_RESULT([not found])

```

```
AC_MSG_ERROR([Could not find a 64-bit integer type.
```

Please report a bug here with details of your platform and compiler:

http://bugs.freedesktop.org/enter_bug.cgi?product=DBus&component=core

To compile D-Bus with all 64-bit integer types removed (not recommended), use the option "--without-64-bit".

This option is likely to be removed in future, unless you report that your platform needs it.])

```
    ],
    dnl else
    [
        DBUS_INT64_TYPE="$dbusint64"
        DBUS_HAVE_INT64=1
        DBUS_INT64_CONSTANT="$dbusint64_constant"
        DBUS_UINT64_CONSTANT="$dbusuint64_constant"
        if test x"$dbusint64_printf_modifier" != x; then
            AC_DEFINE_UNQUOTED(DBUS_INT64_PRINTF_MODIFIER,
[$dbusint64_printf_modifier], [Define to printf modifier for 64 bit
integer type])
        fi
        AC_MSG_RESULT($DBUS_INT64_TYPE)
    ])

```

```
AC_SUBST(DBUS_INT64_TYPE)
AC_SUBST(DBUS_INT64_CONSTANT)
AC_SUBST(DBUS_UINT64_CONSTANT)
AC_SUBST(DBUS_HAVE_INT64)

```

```
### see what 32-bit int is called
AC_MSG_CHECKING([32-bit integer type])

```

```
case 4 in
$ac_cv_sizeof_short)
    dbusint32=short
    ;;
$ac_cv_sizeof_int)
    dbusint32=int
    ;;
$ac_cv_sizeof_long)
    dbusint32=long
    ;;
esac

```

```
if test -z "$dbusint32" ; then
    DBUS_INT32_TYPE="no_int32_type_detected"
    AC_MSG_ERROR([No 32-bit integer type found])

```

```

else
    DBUS_INT32_TYPE="$dbusint32"
    AC_MSG_RESULT($DBUS_INT32_TYPE)
fi

AC_SUBST(DBUS_INT32_TYPE)

### see what 16-bit int is called
AC_MSG_CHECKING([16-bit integer type])

case 2 in
$ac_cv_sizeof_short)
    dbusint16=short
    ;;
$ac_cv_sizeof_int)
    dbusint16=int
    ;;
esac

if test -z "$dbusint16" ; then
    DBUS_INT16_TYPE="no_int16_type_detected"
    AC_MSG_ERROR([No 16-bit integer type found])
else
    DBUS_INT16_TYPE="$dbusint16"
    AC_MSG_RESULT($DBUS_INT16_TYPE)
fi

AC_SUBST(DBUS_INT16_TYPE)

## byte order
case $host_os in
darwin*)
    # check at compile-time, so that it is possible to build
universal
    # (with multiple architectures at once on the compile line)
    AH_VERBATIM([WORDS_BIGENDIAN_DARWIN], [
        /* Use the compiler-provided endianness defines to
allow universal compiling. */
        #if defined(__BIG_ENDIAN__)
        #define WORDS_BIGENDIAN 1
        #endif
    ])
    ;;
*)
    AC_C_BIGENDIAN
    ;;
esac

dnl *****
dnl *** va_copy checks (from GLib) ***
dnl *****
dnl we currently check for all three va_copy possibilities, so we get

```



```

dnl all results in config.log for bug reports.
AC_CACHE_CHECK([for an implementation of va_copy()],dbus_cv_va_copy,[
  AC_LINK_IFELSE([AC_LANG_SOURCE([#include <stdarg.h>
#include <stdlib.h>
  static void f (int i, ...) {
    va_list args1, args2;
    va_start (args1, i);
    va_copy (args2, args1);
    if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
      exit (1);
    va_end (args1); va_end (args2);
  }
  int main() {
    f (0, 42);
    return 0;
  }
  ])],
  [dbus_cv_va_copy=yes],
  [dbus_cv_va_copy=no])
])
AC_CACHE_CHECK([for an implementation of
__va_copy()],dbus_cv__va_copy,[
  AC_LINK_IFELSE([AC_LANG_SOURCE([#include <stdarg.h>
#include <stdlib.h>
  static void f (int i, ...) {
    va_list args1, args2;
    va_start (args1, i);
    __va_copy (args2, args1);
    if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
      exit (1);
    va_end (args1); va_end (args2);
  }
  int main() {
    f (0, 42);
    return 0;
  }
  ])],
  [dbus_cv__va_copy=yes],
  [dbus_cv__va_copy=no])
])

if test "x$dbus_cv_va_copy" = "xyes"; then
  dbus_va_copy_func=va_copy
else if test "x$dbus_cv__va_copy" = "xyes"; then
  dbus_va_copy_func=__va_copy
fi
fi

if test -n "$dbus_va_copy_func"; then
  AC_DEFINE_UNQUOTED(DBUS_VA_COPY,$dbus_va_copy_func,[A 'va_copy'
style function])
fi

AC_LANG_PUSH(C)

```

```

AC_CACHE_CHECK([whether va_lists can be copied by value],
  dbus_cv_va_val_copy,
  [AC_RUN_IFELSE([AC_LANG_PROGRAM(
[[
#include <stdarg.h>
#include <stdlib.h>
]],
)],
  [
static void f (int i, ...) {
va_list args1, args2;
va_start (args1, i);
args2 = args1;
if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
  exit (1);
va_end (args1); va_end (args2);
}
int main() {
  f (0, 42);
  return 0;
}
]])],
  [dbus_cv_va_val_copy=yes],
  [dbus_cv_va_val_copy=no],
  [dbus_cv_va_val_copy=yes])
])
AC_LANG_POP(C)

if test "x$dbus_cv_va_val_copy" = "xno"; then
  AC_DEFINE(DBUS_VA_COPY_AS_ARRAY,1, ['va_lists' cannot be copied as
values])
fi

#### Atomic integers

AC_CACHE_CHECK([whether $CC knows __sync_sub_and_fetch()],
  dbus_cv_sync_sub_and_fetch,
  [AC_LINK_IFELSE([
AC_LANG_PROGRAM([[ ]], [[int a = 4; int b =
__sync_sub_and_fetch(&a, 4); exit(b); ]]]),
  [dbus_cv_sync_sub_and_fetch=yes],
  [dbus_cv_sync_sub_and_fetch=no])
])

if test "x$dbus_cv_sync_sub_and_fetch" = "xyes" ; then
  have_sync=1
else
  have_sync=0
fi

AC_DEFINE_UNQUOTED([DBUS_USE_SYNC], [$have_sync], [Use the gcc __sync
extension])

```

```

#### Various functions
AC_SEARCH_LIBS(socket,[socket network])
AC_CHECK_FUNC(gethostbyname,,[AC_CHECK_LIB(nsl,gethostbyname)])

AC_CHECK_FUNCS(vsnprintf vasprintf nanosleep usleep setenv clearenv
unsetenv socketpair getgrouplist fpathconf setrlimit poll setlocale
localeconv strtoll strtoull issetugid getresuid)

AC_CHECK_HEADERS([syslog.h])
if test "x$ac_cv_header_syslog_h" = "xyes"; then
  AC_CHECK_DECLS([LOG_PERROR], [], [], [[#include <syslog.h>]])
fi

#### Check for broken poll; taken from Glib's configure

AC_MSG_CHECKING([for broken poll])
AC_RUN_IFELSE([AC_LANG_SOURCE([[
  #include <stdlib.h>
  #include <fcntl.h>
  #include <poll.h>
  #ifdef HAVE_SYS_POLL_H
  #include <sys/poll.h>
  #endif
  int main(void) {
    struct pollfd fds[1];
    int fd;
    fd = open("/dev/null", 1);
    fds[0].fd = fd;
    fds[0].events = POLLIN;
    fds[0].revents = 0;
    if (poll(fds, 1, 0) < 0 || (fds[0].revents & POLLNVAL) != 0) {
      exit(1); /* Does not work for devices -- fail */
    }
    exit(0);
  }
]]),
[broken_poll=no],
[broken_poll=yes
  AC_DEFINE(BROKEN_POLL,1,[poll doesn't work on devices])],
[broken_poll="no (cross compiling)"])
AC_MSG_RESULT($broken_poll)

AC_MSG_CHECKING(for dirfd)
AC_LINK_IFELSE([AC_LANG_PROGRAM([[
#include <sys/types.h>
#include <dirent.h>
]], [[
DIR *dirp;
dirp = opendir(".");
dirfd(dirp);
closedir(dirp);
]])],

```

```

[dbus_have_dirfd=yes],
[dbus_have_dirfd=no])
AC_MSG_RESULT($dbus_have_dirfd)
if test "$dbus_have_dirfd" = yes; then
    AC_DEFINE(HAVE_DIRFD,1,[Have dirfd function])
else
    AC_MSG_CHECKING(for DIR *dirp->dd_fd)
    AC_LINK_IFELSE([AC_LANG_PROGRAM([[
#include <sys/types.h>
#include <dirent.h>
    ]], [[
DIR *dirp;
int fd;
dirp = opendir(".");
fd = dirp->dd_fd;
closedir(dirp);
    ]]]),
    [dbus_have_ddfd=yes],
    [dbus_have_ddfd=no])
    AC_MSG_RESULT($dbus_have_ddfd)
    if test "$dbus_have_ddfd" = yes; then
        AC_DEFINE(HAVE_DDFD,1,[Have the ddfd member of DIR])
    fi
fi

AC_CHECK_HEADERS(sys/resource.h)

AC_CHECK_HEADERS(dirent.h)

AC_CHECK_HEADERS(execinfo.h, [AC_CHECK_FUNCS(backtrace)])

AC_CHECK_HEADERS(errno.h)

AC_CHECK_HEADERS(signal.h)

AC_CHECK_HEADERS(locale.h)

AC_CHECK_HEADERS(byteswap.h)

AC_CHECK_HEADERS(unistd.h)

AC_CHECK_HEADERS(ws2tcpip.h)

AC_CHECK_HEADERS(wspiapi.h)

# Add -D_POSIX_PTHREAD_SEMANTICS if on Solaris
#
case $host_os in
    solaris*)
        CFLAGS="$CFLAGS -D_POSIX_PTHREAD_SEMANTICS" ;;
esac

```

```

# checking for a posix version of getpwnam_r
# if we are cross compiling and can not run the test
# assume getpwnam_r is the posix version
# it is up to the person cross compiling to change
# this behavior if desired
AC_LANG_PUSH(C)
AC_CACHE_CHECK([for posix getpwnam_r],
                ac_cv_func_posix_getpwnam_r,
                [AC_RUN_IFELSE([AC_LANG_PROGRAM(
[[
#include <errno.h>
#include <pwd.h>
]],
[[
char buffer[10000];
struct passwd pwd, *pwptr = &pwd;
int error;
errno = 0;
error = getpwnam_r ("", &pwd, buffer,
                    sizeof (buffer), &pwptr);
return (error < 0 && errno == ENOSYS)
        || error == ENOSYS;
]]),
[ac_cv_func_posix_getpwnam_r=yes],
[ac_cv_func_posix_getpwnam_r=no],
[ac_cv_func_posix_getpwnam_r=yes]
)])
AC_LANG_POP(C)

if test "$ac_cv_func_posix_getpwnam_r" = yes; then
    AC_DEFINE(HAVE_POSIX_GETPWNAM_R,1,
              [Have POSIX function getpwnam_r])
else
    AC_CACHE_CHECK([for nonposix getpwnam_r],
                    ac_cv_func_nonposix_getpwnam_r,
                    [AC_LINK_IFELSE([AC_LANG_PROGRAM([[#include <pwd.h>]]),
[[char buffer[10000];
                                struct passwd pwd;
                                getpwnam_r ("", &pwd, buffer,
                                                sizeof (buffer));]]),
[ac_cv_func_nonposix_getpwnam_r=yes],
[ac_cv_func_nonposix_getpwnam_r=no]])
    if test "$ac_cv_func_nonposix_getpwnam_r" = yes; then
        AC_DEFINE(HAVE_NONPOSIX_GETPWNAM_R,1,
                  [Have non-POSIX function getpwnam_r])
    fi
fi

dnl check for socklen_t
AC_MSG_CHECKING(whether socklen_t is defined)
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[
#include <sys/types.h>

```

```

#include <sys/socket.h>
#include <netdb.h>
]], [[
socklen_t foo;
foo = 1;
]]],
[dbus_have_socklen_t=yes],
[dbus_have_socklen_t=no])
AC_MSG_RESULT($dbus_have_socklen_t)

if test "x$dbus_have_socklen_t" = "xyes"; then
    AC_DEFINE(HAVE_SOCKLEN_T,1,[Have socklen_t type])
fi

dnl check for writev header and writev function so we're
dnl good to go if HAVE_WRITEV gets defined.
AC_CHECK_HEADERS(sys/uio.h, [AC_CHECK_FUNCS(writev)])

dnl needed on darwin for NAME_MAX
AC_CHECK_HEADERS(sys/syslimits.h)

dnl Make it easy to check if we have MSG_NOSIGNAL without actually
having to include sys/socket.h
AC_CHECK_DECLS([MSG_NOSIGNAL], [], [], [[ #include <sys/types.h>
#include <sys/socket.h> ]])

dnl check for flavours of varargs macros (test from GLib)
AC_MSG_CHECKING(for ISO C99 varargs macros in C)
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[]], [[
int a(int p1, int p2, int p3);
#define call_a(...) a(1, __VA_ARGS__)
call_a(2,3);
]]]),
[dbus_have_iso_c_varargs=yes],
[dbus_have_iso_c_varargs=no])
AC_MSG_RESULT($dbus_have_iso_c_varargs)

AC_MSG_CHECKING(for GNUC varargs macros)
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[]], [[
int a(int p1, int p2, int p3);
#define call_a(params...) a(1,params)
call_a(2,3);
]]]),
[dbus_have_gnuc_varargs=yes],
[dbus_have_gnuc_varargs=no])
AC_MSG_RESULT($dbus_have_gnuc_varargs)

dnl Output varargs tests
if test x$dbus_have_iso_c_varargs = xyес; then
    AC_DEFINE(HAVE_ISO_VARARGS,1,[Have ISO C99 varargs macros])
fi
if test x$dbus_have_gnuc_varargs = xyес; then

```

```

    AC_DEFINE(HAVE_GNUC_VARARGS,1,[Have GNU-style varargs macros])
fi

dnl Check for various credentials.
AC_MSG_CHECKING(for struct cmsgcred)
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[
#include <sys/types.h>
#include <sys/socket.h>
]], [[
struct cmsgcred cred;

cred.cmcred_pid = 0;
]])],
[dbus_have_struct_cmsgcred=yes],
[dbus_have_struct_cmsgcred=no])
AC_MSG_RESULT($dbus_have_struct_cmsgcred)

if test x$dbus_have_struct_cmsgcred = xyes; then
    AC_DEFINE(HAVE_CMSGCRED,1,[Have cmsgcred structure])
fi

AC_CHECK_FUNCS(getpeerucred getpeereid)

AC_CHECK_FUNCS(pipe2 accept4)

#### Abstract sockets

if test x$enable_abstract_sockets = xauto; then
AC_LANG_PUSH(C)
warn_on_xcompile=no
AC_CACHE_CHECK([abstract socket namespace],
    ac_cv_have_abstract_sockets,
    [AC_RUN_IFELSE([AC_LANG_PROGRAM(
[[
#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <errno.h>
]],
[[
    size_t slen;
    int listen_fd;
    struct sockaddr_un addr;

    listen_fd = socket (PF_UNIX, SOCK_STREAM, 0);

    if (listen_fd < 0)
    {
        fprintf (stderr, "socket() failed: %s\n", strerror (errno));

```

```

        exit (1);
    }

    memset (&addr, '\0', sizeof (addr));
    addr.sun_family = AF_UNIX;
    strcpy (addr.sun_path, "X/tmp/dbus-fake-socket-path-used-in-
configure-test");
    /* SUN_LEN uses strlen() so need to calculate it before adding \0 at
the
* beginning.
*/
    slen = SUN_LEN(&addr);
    addr.sun_path[0] = '\0'; /* this is what makes it abstract */

    if (bind (listen_fd, (struct sockaddr*) &addr, slen) < 0)
    {
        fprintf (stderr, "Abstract socket namespace bind() failed:
%s\n",
                strerror (errno));
        exit (1);
    }
    else
        exit (0);
]]],

        [ac_cv_have_abstract_sockets=yes],
        [ac_cv_have_abstract_sockets=no],
        [
            ac_cv_have_abstract_sockets=no
            warn_on_xcompile=yes
        ]
    ))
if test x$warn_on_xcompile = xyes ; then
    AC_MSG_WARN([Cannot check for abstract sockets when cross-compiling,
please use --enable-abstract-sockets])
fi
AC_LANG_POP(C)
fi

if test x$enable_abstract_sockets = xyes; then
    if test x$ac_cv_have_abstract_sockets = xno; then
        AC_MSG_ERROR([Abstract sockets explicitly required, and support
not detected.])
    fi
fi

if test x$enable_abstract_sockets = xno; then
    ac_cv_have_abstract_sockets=no;
fi

if test x$ac_cv_have_abstract_sockets = xyes ; then
    DBUS_PATH_OR_ABSTRACT=abstract
    AC_DEFINE(HAVE_ABSTRACT_SOCKETS,1,[Have abstract socket namespace])

```



```

else
    DBUS_PATH_OR_ABSTRACT=path
fi

# this is used in addresses to prefer abstract, e.g.
# unix:path=/foo or unix:abstract=/foo
AC_SUBST(DBUS_PATH_OR_ABSTRACT)

PKG_PROG_PKG_CONFIG

#### Sort out XML library

# see what we have
AC_CHECK_LIB(expat, XML_ParserCreate_MM,
    [ AC_CHECK_HEADERS(expat.h, have_expat=true,
    have_expat=false) ],
    have_expat=false)

# see what we want to use
dbus_use_libxml=false
dbus_use_expat=false
if test x$with_xml = xexpat; then
    if ! $have_expat ; then
        AC_MSG_ERROR([Explicitly requested expat but expat not
found])
    fi
    dbus_use_expat=true
elif test x$with_xml = xlibxml; then
    PKG_CHECK_MODULES(LIBXML, libxml-2.0 >= 2.6.0,
    have_libxml=true, have_libxml=false)
    if ! $have_libxml ; then
        AC_MSG_ERROR([Explicitly requested libxml but libxml not
found])
    fi
    dbus_use_libxml=true
else
    ### expat is the default because libxml can't currently
survive
    ### our brutal OOM-handling unit test setup.
    ### http://bugzilla.gnome.org/show\_bug.cgi?id=109368
    if test x$have_expat = xfalse; then
        AC_MSG_ERROR([Could not find expat.h, check config.log
for failed attempts])
    fi
    ### By default, only use Expat since it's tested and known to
work.  If you're a
    ### general-purpose OS vendor, please don't enable libxml.  For
embedded use
    ### if your OS is built around libxml, that's another case.
    dbus_use_expat=true
fi

```

```

AM_CONDITIONAL(DBUS_USE_EXPAT, $dbus_use_expat)
AM_CONDITIONAL(DBUS_USE_LIBXML, $dbus_use_libxml)

if $dbus_use_expat; then
  XML_LIBS=-lexpat
  XML_CFLAGS=
fi
if $dbus_use_libxml; then
  XML_LIBS=$LIBXML_LIBS
  XML_CFLAGS=$LIBXML_CFLAGS
fi
AC_SUBST([XML_CFLAGS])
AC_SUBST([XML_LIBS])

# Thread lib detection
AC_CHECK_FUNC(pthread_cond_timedwait, [AC_CHECK_LIB(pthread, pthread_cond_timedwait,

                                                                    [THREAD_LIBS="-
lpthread" ]])
save_libs="$LIBS"
LIBS="$LIBS $THREAD_LIBS"
AC_CHECK_FUNC(pthread_condattr_setclock, have_pthread_condattr_setclock
=true, have_pthread_condattr_setclock=false)
if test x$have_pthread_condattr_setclock = xtrue; then
  AC_SEARCH_LIBS([clock_getres], [rt], [THREAD_LIBS="$THREAD_LIBS -
lrt"])
  AC_MSG_CHECKING([for CLOCK_MONOTONIC])
  AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[#include <time.h>
#include <pthread.h>
]], [[
struct timespec monotonic_timer;
pthread_condattr_t attr;
pthread_condattr_init (&attr);
pthread_condattr_setclock (&attr, CLOCK_MONOTONIC);
clock_getres (CLOCK_MONOTONIC, &monotonic_timer);
]]), [
[have_clock_monotonic=true],
[have_clock_monotonic=false]
if test x$have_clock_monotonic = xtrue; then
  AC_MSG_RESULT([found])
  AC_DEFINE(HAVE_MONOTONIC_CLOCK, 1, [Define if we have
CLOCK_MONOTONIC])
else
  AC_MSG_RESULT([not found])
fi
fi
LIBS="$save_libs"

AC_SUBST([THREAD_LIBS])

# SELinux detection
if test x$enable_selinux = xno ; then

```

```

    have_selinux=no;
else
    # See if we have SELinux library
    AC_CHECK_LIB(selinux, is_selinux_enabled,
        have_selinux=yes, have_selinux=no)

    # see if we have the SELinux header with the new D-Bus stuff in it
    if test x$have_selinux = xyes ; then
        AC_MSG_CHECKING([for DBUS Flask permissions in
selinux/av_permissions.h])
        AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[#include
<selinux/av_permissions.h>]],
            [[#ifdef DBUS__ACQUIRE_SVC return 0;
            #else
            #error DBUS__ACQUIRE_SVC not defined
            #endif]])],
            [have_selinux=yes],
            [have_selinux=no])
        AC_MSG_RESULT($have_selinux)
    fi

    if test x$enable_selinux = xauto ; then
        if test x$have_selinux = xno ; then
            AC_MSG_WARN([Sufficiently new SELinux library not
found])
        fi
    else
        if test x$have_selinux = xno ; then
            AC_MSG_ERROR([SELinux explicitly required, and SELinux
library not found])
        fi
    fi
fi

AM_CONDITIONAL(HAVE_SELINUX, test x$have_selinux = xyes)

if test x$have_selinux = xyes ; then
    # the selinux code creates threads
    # which requires libpthread even on linux

AC_CHECK_FUNC(pthread_create,, [AC_CHECK_LIB(pthread,pthread_create,
[SELINUX_THREAD_LIBS="-lpthread"])]))

    SELINUX_LIBS="-lselinux $SELINUX_THREAD_LIBS"
    AC_DEFINE(HAVE_SELINUX,1,[SELinux support])
else
    SELINUX_LIBS=
fi

# inotify checks
if test x$enable_inotify = xno ; then

```

```

    have_inotify=no;
else
    AC_CHECK_HEADERS(sys/inotify.h, have_inotify=yes, have_inotify=no)
fi

dnl check if inotify backend is enabled
if test x$have_inotify = xyes; then
    AC_DEFINE(DBUS_BUS_ENABLE_INOTIFY,1,[Use inotify])
    AC_CHECK_FUNCS(inotify_init1)
fi

AM_CONDITIONAL(DBUS_BUS_ENABLE_INOTIFY, test x$have_inotify = xyes)

# dnotify checks
if test x$enable_dnotify = xno ; then
    have_dnotify=no;
else
    if test x$have_inotify = xno -a x$host_os = xlinux-gnu -o
x$host_os = xlinux; then
        have_dnotify=yes;
    else
        have_dnotify=no;
    fi
fi

dnl check if dnotify backend is enabled
if test x$have_dnotify = xyes; then
    AC_DEFINE(DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX,1,[Use dnotify on
Linux])
fi

AM_CONDITIONAL(DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX, test x$have_dnotify =
xyes)

# For simplicity, we require the userland API for epoll_create1 at
# compile-time (glibc 2.9), but we'll run on kernels that turn out
# not to have it at runtime.
AC_ARG_ENABLE([epoll],
              [AS_HELP_STRING([--enable-epoll],[use epoll(4) on
Linux])],
              [enable_epoll=$enableval], [enable_epoll=auto])
if test x$enable_epoll = xno; then
    have_linux_epoll=no
else
    AC_MSG_CHECKING([for Linux epoll(4)])
    AC_LINK_IFELSE([AC_LANG_PROGRAM(
        [
            #ifndef __linux__
            #error This is not Linux
            #endif
            #include <sys/epoll.h>
        ],

```

```

        [epoll_create1 (EPOLL_CLOEXEC);]],
        [have_linux_epoll=yes],
        [have_linux_epoll=no])
AC_MSG_RESULT([$have_linux_epoll])
fi
if test x$enable_epoll,$have_linux_epoll = xyes,no; then
    AC_MSG_ERROR([epoll support explicitly enabled but not available])
fi
if test x$have_linux_epoll = xyes; then
    AC_DEFINE([DBUS_HAVE_LINUX_EPOLL], 1, [Define to use epoll(4) on
Linux])
fi
AM_CONDITIONAL([HAVE_LINUX_EPOLL], [test x$have_linux_epoll = xyes])

# kqueue checks
if test x$enable_kqueue = xno ; then
    have_kqueue=no
else
    have_kqueue=yes
    AC_CHECK_HEADER(sys/event.h, , have_kqueue=no)
    AC_CHECK_FUNC(kqueue, , have_kqueue=no)

    if test x$enable_kqueue = xyes -a x$have_kqueue = xno; then
        AC_MSG_ERROR(kqueue support explicitly enabled but not
available)
    fi
fi

dnl check if kqueue backend is enabled
if test x$have_kqueue = xyes; then
    AC_DEFINE(DBUS_BUS_ENABLE_KQUEUE,1,[Use kqueue])
fi

AM_CONDITIONAL(DBUS_BUS_ENABLE_KQUEUE, test x$have_kqueue = xyes)

# launchd checks
if test x$enable_launchd = xno ; then
    have_launchd=no
else
    have_launchd=yes
    AC_CHECK_HEADER([launch.h], , have_launchd=no)
    AC_PATH_PROG([LAUNCHCTL], [launchctl])
    if test "x$LAUNCHCTL" = "x"; then
        have_launchd=no
    fi

    if test x$enable_launchd = xyes && test x$have_launchd = xno ;
then
        AC_MSG_ERROR([launchd support explicitly enabled but not
available])
    fi
fi

```

```

dnl check if launchd is enabled
if test x$have_launchd = xyes; then
    AC_DEFINE(DBUS_ENABLE_LAUNCHD,1,[Use launchd autolaunch])
fi

AM_CONDITIONAL(DBUS_ENABLE_LAUNCHD, test x$have_launchd = xyes)

#### Directory to place launchd agent file
if test "x$with_launchd_agent_dir" = "x"; then
    LAUNCHD_AGENT_DIR="/Library/LaunchAgents"
else
    LAUNCHD_AGENT_DIR="$with_launchd_agent_dir"
fi

AC_SUBST(LAUNCHD_AGENT_DIR)

dnl console owner file
if test x$enable_console_owner_file = xno ; then
    have_console_owner_file=no;
else
    case $host_os in
        solaris*)
            have_console_owner_file=yes;
            AC_DEFINE(HAVE_CONSOLE_OWNER_FILE,1,[Have console owner file])
            ;;
        *)
            have_console_owner_file=no;;
    esac
fi

AM_CONDITIONAL(HAVE_CONSOLE_OWNER_FILE, test x$have_console_owner_file
= xyes)

dnl systemd detection
if test x$enable_systemd = xno ; then
    have_systemd=no;
else
    PKG_CHECK_MODULES(SYSTEMD,
        [libsystemd-login >= 32, libsystemd-daemon >= 32],
        have_systemd=yes,
        have_systemd=no)
fi

if test x$have_systemd = xyes; then
    AC_DEFINE(HAVE_SYSTEMD,1,[Have systemd])
fi

if test x$enable_systemd = xyes -a x$have_systemd != xyes ; then
    AC_MSG_ERROR([Explicitly requested systemd support, but systemd
not found])
fi

```

```

# libaudit detection
if test x$enable_libaudit = xno ; then
    have_libaudit=no;
else
    # See if we have audit daemon & capabilities library
    AC_CHECK_LIB(audit, audit_log_user_avc_message,
                 have_libaudit=yes, have_libaudit=no)
    if test x$have_libaudit = xyes ; then
        AC_CHECK_LIB(cap-ng, capng_clear,
                     have_libaudit=yes, have_libaudit=no)
    fi
fi

AM_CONDITIONAL(HAVE_LIBAUDIT, test x$have_libaudit = xyes)

if test x$have_libaudit = xyes ; then
    SELINUX_LIBS="$SELINUX_LIBS -laudit -lcap-ng"
    AC_DEFINE(HAVE_LIBAUDIT,1,[audit daemon SELinux support])
fi

AC_SUBST([SELINUX_LIBS])

# Check for ADT API (Solaris Basic Security Mode auditing)
AC_MSG_CHECKING(for ADT API)
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[
#include <bsm/adt.h>
adt_user_context = ADT_USER;
]], [[]])], [ check_adt_audit=yes ], [ check_adt_audit=no ])

if test ${check_adt_audit} = yes
then
    AC_DEFINE([HAVE_ADT], [], [Adt audit API])
    ADT_LIBS="-lbsm"
    LIBS="-lbsm $LIBS"
    AC_MSG_RESULT(yes)
else
    AC_MSG_RESULT(no)
fi
AC_SUBST([ADT_LIBS])

# Check for SCM_RIGHTS
AC_MSG_CHECKING([for SCM_RIGHTS])
AC_COMPILE_IFELSE([AC_LANG_PROGRAM([[
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
static int x = SCM_RIGHTS;
]], [[]])],
[ AC_MSG_RESULT([supported])
  AC_DEFINE([HAVE_UNIX_FD_PASSING], [1], [Supports sending UNIX file
descriptors]) ],

```

```

[ AC_MSG_RESULT([not supported]) ])

NETWORK_libs=
if test x$dbus_win = xyes ; then
  if test x$dbus_wince = xyes ; then
    NETWORK_libs="-lws2"
  else
    NETWORK_libs="-lws2_32"
  fi
fi

AC_SUBST([NETWORK_libs])

AC_ARG_WITH([valgrind],
  [AS_HELP_STRING([--with-valgrind],
    [Add instrumentation to help valgrind to understand our
allocator])],
  [],
  [with_valgrind=no])

if test x$with_valgrind != xno; then
  PKG_CHECK_MODULES([VALGRIND], [valgrind >= 3.6])
  AC_DEFINE([WITH_VALGRIND], [1], [Define to add Valgrind
instrumentation])
fi

#### Set up final flags
LIBDBUS_LIBS="$THREAD_LIBS $NETWORK_libs"
AC_SUBST([LIBDBUS_LIBS])

### X11 detection
DBUS_X_LIBS=
DBUS_X_CFLAGS=

AC_ARG_ENABLE([x11-autolaunch],
  AS_HELP_STRING([--enable-x11-autolaunch], [build with X11 auto-
launch support]),
  [], [enable_x11_autolaunch=auto])

if test "x$dbus_win" = xyes; then
  if test "x$enable_x11_autolaunch" = xyes; then
    AC_MSG_ERROR([X11 auto-launch is not supported on Windows])
  fi

  enable_x11_autolaunch=no
  have_x11=no
else
  AC_PATH_XTRA

  if test "x$no_x" = xyes; then
    have_x11=no
  else

```



```

        have_x11=yes
        DBUS_X_LIBS="$X_LIBS $X_PRE_LIBS -lX11 $X_EXTRA_LIBS"
        DBUS_X_CFLAGS="$X_CFLAGS"
    fi
fi

if test "x$enable_x11_autolaunch,$have_x11" = xyes,no; then
    AC_MSG_ERROR([X11 auto-launch requires X headers/libraries])
else
    # move from "auto" to "yes" or "no" if necessary
    if test "x$enable_x11_autolaunch" != xno; then
        enable_x11_autolaunch="$have_x11"
    fi
fi

if test "x$have_x11" = xyes ; then
    AC_DEFINE([DBUS_BUILD_X11], [1], [Define to build X11
functionality])
fi

if test "x$enable_x11_autolaunch" = xyes ; then
    AC_DEFINE([DBUS_ENABLE_X11_AUTOLAUNCH], [1], [Define to enable X11
auto-launch])
fi

AC_SUBST([DBUS_X_CFLAGS])
AC_SUBST([DBUS_X_LIBS])

#### gcc warning flags

cc_supports_flag() {
    AC_MSG_CHECKING(whether $CC supports "$*")
    save_CFLAGS="$CFLAGS"
    CFLAGS="$*"
    AC_TRY_COMPILE([], [], [rc=yes], [rc=no])
    CFLAGS="$save_CFLAGS"
    AC_MSG_RESULT([$rc])
    test "x$rc" = xyes
}

TP_COMPILER_WARNINGS([WARNING_CFLAGS],
    dnl Use -Werror by default if:
    dnl - we're not on Windows (too many warnings), and
    dnl - we're in developer mode (a D-Bus developer, not a distro or
end-user)
    dnl Override with --enable-Werror or --disable-Werror
    [test x$dbus_win != xyes -a x$dbus_cygwin != xyes -a
x$enable_developer = xyes],

    dnl Enable these warnings if possible:
    [all \
extra \

```

```

char-subscripts \
missing-declarations \
missing-prototypes \
nested-externs \
pointer-arith \
cast-align \
no-address \
float-equal \
declaration-after-statement \
],

dnl Disable these warnings if possible, make them non-fatal if
possible,
dnl and don't enable -Werror unless we succeeded.
dnl
dnl Intentional:
dnl - $DISABLE_UNUSED_WARNINGS disables unused-label warnings if not
dnl   checking or not asserting
dnl - missing field initializers being 0 is a C feature, not a bug
dnl - unused-parameter is to make writing callbacks less annoying
dnl
dnl To be fixed one day:
dnl - sign-compare and pointer-sign are workarounds for fd.o #17433
dnl - type-limits is probably a bug too, but having the rest of -
Wextra
dnl   is better than nothing
[$DISABLE_UNUSED_WARNINGS \
missing-field-initializers \
unused-parameter \
sign-compare \
pointer-sign \
type-limits \
])

if test "x$GCC" = "xyes"; then
# We're treating -fno-common like a warning: it makes the linker
more
# strict, because on some systems the linker is *always* this strict
TP_ADD_COMPILER_FLAG([WARNING_CFLAGS], [-fno-common])

# http://bugs.freedesktop.org/show_bug.cgi?id=10599
TP_ADD_COMPILER_FLAG([WARNING_CFLAGS], [-fno-strict-aliasing])

if test "x$enable_ansi" = "xyes"; then
TP_ADD_COMPILER_FLAG([WARNING_CFLAGS],
[-ansi -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -pedantic])
fi
fi

dnl In principle we should put WARNING_CFLAGS in each Makefile.am like
dnl telepathy-glib does, since CFLAGS is meant to be reserved for the
user...

```

```

dnl but prepending to CFLAGS (so the user can override it with later
CFLAGS)
dnl is the next best thing
CFLAGS="$WARNING_CFLAGS $CFLAGS"

case $host_os in
    solaris*)
        # Solaris' C library apparently needs these runes to be
        threadsafe...
        CFLAGS="$CFLAGS -D_POSIX_PTHREAD_SEMANTICS -D_REENTRANT"
        # ... this opt-in to get sockaddr_in6 and sockaddr_storage...
        CFLAGS="$CFLAGS -D__EXTENSIONS__"
        # ... and this opt-in to get file descriptor passing support
        CFLAGS="$CFLAGS -D_XOPEN_SOURCE=500"
        ;;
esac

### Doxygen Documentation

AC_PATH_PROG(DOXYGEN, doxygen, no)

AC_MSG_CHECKING([whether to build Doxygen documentation])

if test x$DOXYGEN = xno ; then
    have_doxygen=no
else
    have_doxygen=yes
fi

if test x$enable_doxygen_docs = xauto ; then
    if test x$have_doxygen = xno ; then
        enable_doxygen_docs=no
    else
        enable_doxygen_docs=yes
    fi
fi

if test x$enable_doxygen_docs = xyes; then
    if test x$have_doxygen = xno; then
        AC_MSG_ERROR([Building Doxygen docs explicitly required, but
Doxygen not found])
    fi
fi

AM_CONDITIONAL(DBUS_DOXYGEN_DOCS_ENABLED, test x$enable_doxygen_docs =
xyes)
AC_MSG_RESULT($enable_doxygen_docs)

AC_CHECK_PROGS([XSLTPROC], [xsltproc])
AM_CONDITIONAL([DBUS_HAVE_XSLTPROC], [test "x$XSLTPROC" != "x"])

### XML Documentation

```

```

AC_PATH_PROG(XMLTO, xmlto, no)

AC_MSG_CHECKING([whether to build XML documentation])

if test x$XMLTO = xno ; then
    have_xmlto=no
else
    have_xmlto=yes
fi

if test x$enable_xml_docs = xauto ; then
    if test x$have_xmlto = xno ; then
        enable_xml_docs=no
    else
        enable_xml_docs=yes
    fi
fi

if test x$enable_xml_docs = xyes; then
    if test x$have_xmlto = xno; then
        AC_MSG_ERROR([Building XML docs explicitly required, but xmlto
not found])
    fi
fi

AM_CONDITIONAL(DBUS_XML_DOCS_ENABLED, test x$enable_xml_docs = xyes)
AC_MSG_RESULT($enable_xml_docs)

AC_PATH_PROG([MAN2HTML], [man2html])
AC_ARG_VAR([MAN2HTML], [Path to man2html (optional)])
AM_CONDITIONAL(DBUS_HAVE_MAN2HTML, test x$MAN2HTML != x)

AM_CONDITIONAL(DBUS_CAN_UPLOAD_DOCS,
    test x$enable_doxygen_docs = xyes -a x$enable_xml_docs = xyes -a \
    x$MAN2HTML != x)

#### Have to go $localstatedir->$prefix/var->/usr/local/var

#### find the actual value for $prefix that we'll end up with
## (I know this is broken and should be done in the Makefile, but
## that's a major pain and almost nobody actually seems to care)
AS_AC_EXPAND(EXPANDED_PREFIX, "$prefix")
AS_AC_EXPAND(EXPANDED_LOCALSTATEDIR, "$localstatedir")
AS_AC_EXPAND(EXPANDED_SYSCONFDIR, "$sysconfdir")
AS_AC_EXPAND(EXPANDED_BINDIR, "$bindir")
AS_AC_EXPAND(EXPANDED_LIBDIR, "$libdir")
AS_AC_EXPAND(EXPANDED_LIBEXECDIR, "$libexecdir")
AS_AC_EXPAND(EXPANDED_DATADIR, "$datadir")

#### Check our operating system
operating_system=unknown

```

```

if test -f /etc/redhat-release || test -f $EXPANDED_SYSCONFDIR/redhat-
release ; then
    operating_system=redhat
fi

if test -f /etc/slackware-version || test -f
$EXPANDED_SYSCONFDIR/slackware-version ; then
    operating_system=slackware
fi

if test -f /usr/bin/cygwin1.dll || test -f
$EXPANDED_BINDIR/cygwin1.dll ; then
    operating_system=cygwin
fi

#### Sort out init scripts

if test x$with_init_scripts = x; then
    case x$operating_system in
        xredhat) with_init_scripts=redhat ;;
        xslackware) with_init_scripts=slackware ;;
        xcygwin) with_init_scripts=cygwin ;;
        *) with_init_scripts=none ;;
    esac
fi

AM_CONDITIONAL(DBUS_INIT_SCRIPTS_RED_HAT, test x$with_init_scripts =
xredhat)
AM_CONDITIONAL(DBUS_INIT_SCRIPTS_SLACKWARE, test x$with_init_scripts =
xslackware)
AM_CONDITIONAL(DBUS_INIT_SCRIPTS_CYGWIN, test x$with_init_scripts =
xcygwin)

##### systemd unit files
AC_ARG_WITH([systemdsystemunitdir],
AS_HELP_STRING([--with-systemdsystemunitdir=DIR], [Directory for
systemd service files]),
    [],
    [
        [
            PKG_CHECK_EXISTS([systemd],
                [with_systemdsystemunitdir=${($PKG_CONFIG --
variable=systemdsystemunitdir systemd)}],
                [with_systemdsystemunitdir=no])
        ]
    ])
if test "x$with_systemdsystemunitdir" != xno; then
    AC_SUBST([systemdsystemunitdir], [$with_systemdsystemunitdir])
fi
AM_CONDITIONAL(HAVE_SYSTEMD, [test -n "$with_systemdsystemunitdir" -a
"x$with_systemdsystemunitdir" != xno ])

##### Set up location for system bus socket
if ! test -z "$with_system_socket"; then

```

```

    DBUS_SYSTEM_SOCKET=$with_system_socket
else

DBUS_SYSTEM_SOCKET=${EXPANDED_LOCALSTATEDIR}/run/dbus/system_bus_socket
fi

AC_SUBST(DBUS_SYSTEM_SOCKET)
AC_DEFINE_UNQUOTED(DBUS_SYSTEM_SOCKET, "$DBUS_SYSTEM_SOCKET", [The name
of the socket the system bus listens on by default])

## system bus only listens on local domain sockets, and never
## on an abstract socket (so only root can create the socket)
DBUS_SYSTEM_BUS_DEFAULT_ADDRESS="unix:path=$DBUS_SYSTEM_SOCKET"
AC_SUBST(DBUS_SYSTEM_BUS_DEFAULT_ADDRESS)
AC_DEFINE_UNQUOTED(DBUS_SYSTEM_BUS_DEFAULT_ADDRESS,
"$DBUS_SYSTEM_BUS_DEFAULT_ADDRESS", [The default D-Bus address of the
system bus])

#### Set up the pid file
if ! test -z "$with_system_pid_file"; then
    DBUS_SYSTEM_PID_FILE=$with_system_pid_file
elif test x$with_init_scripts = xredhat ; then
    DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/messagebus.pid
else
    DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/dbus/pid
fi

AC_SUBST(DBUS_SYSTEM_PID_FILE)

#### Directory to check for console ownership
if ! test -z "$with_console_auth_dir"; then
    DBUS_CONSOLE_AUTH_DIR=$with_console_auth_dir
else
    DBUS_CONSOLE_AUTH_DIR=/var/run/console/
fi

AC_SUBST(DBUS_CONSOLE_AUTH_DIR)
AC_DEFINE_UNQUOTED(DBUS_CONSOLE_AUTH_DIR, "$DBUS_CONSOLE_AUTH_DIR",
[Directory to check for console ownership])

#### File to check for console ownership
if test x$have_console_owner_file = xyes; then
    if ! test -z "$with_console_owner_file"; then
        DBUS_CONSOLE_OWNER_FILE=$with_console_owner_file
    else
        DBUS_CONSOLE_OWNER_FILE=/dev/console
    fi
else
    DBUS_CONSOLE_OWNER_FILE=
fi

```

```

AC_SUBST(DBUS_CONSOLE_OWNER_FILE)
AC_DEFINE_UNQUOTED(DBUS_CONSOLE_OWNER_FILE,
"$DBUS_CONSOLE_OWNER_FILE", [File to check for console ownership])

#### User to start the system bus as
if test -z "$with_dbus_user" ; then
    DBUS_USER=messagebus
else
    DBUS_USER=$with_dbus_user
fi
AC_SUBST(DBUS_USER)
AC_DEFINE_UNQUOTED(DBUS_USER,"$DBUS_USER", [User for running the
system BUS daemon])

#### Prefix to install into
DBUS_PREFIX=$EXPANDED_PREFIX
AC_SUBST(DBUS_PREFIX)
AC_DEFINE_UNQUOTED(DBUS_PREFIX,"$DBUS_PREFIX", [Prefix for installing
DBUS])

#### Directory to install data files into
DBUS_DATADIR=$EXPANDED_DATADIR
AC_SUBST(DBUS_DATADIR)
AC_DEFINE_UNQUOTED(DBUS_DATADIR,"$DBUS_DATADIR", [Directory for
installing DBUS data files])

#### Directory to install dbus-daemon
if test -z "$with_dbus_daemon_dir" ; then
    DBUS_DAEMONDIR=$EXPANDED_BINDIR
else
    DBUS_DAEMONDIR=$with_dbus_daemon_dir
fi
AC_SUBST(DBUS_DAEMONDIR)
AC_DEFINE_UNQUOTED(DBUS_DAEMONDIR,"$DBUS_DAEMONDIR", [Directory for
installing the DBUS daemon])

#### Directory to install the other binaries
DBUS_BINDIR="$EXPANDED_BINDIR"
AC_SUBST(DBUS_BINDIR)
AC_DEFINE_UNQUOTED(DBUS_BINDIR,"$DBUS_BINDIR", [Directory for
installing the binaries])

#### Directory to install the libexec binaries
DBUS_LIBEXECDIR="$EXPANDED_LIBEXECDIR"
AC_SUBST(DBUS_LIBEXECDIR)
AC_DEFINE_UNQUOTED(DBUS_LIBEXECDIR,"$DBUS_LIBEXECDIR", [Directory for
installing the libexec binaries])

#### Tell tests where to find certain stuff in builddir

DBUS_PWD=`pwd`

```

```

# Useful in a cross-compilation environment, where the tests are run
on the host system.
AC_ARG_WITH(dbus-test-dir, AS_HELP_STRING([--with-dbus-test-
dir=[dirname]], [path where the tests tools are available]),
        DBUS_PWD=$withval)

DBUS_TEST_EXEC="$DBUS_PWD/test"
DBUS_TEST_DATA="$DBUS_PWD/test/data"

AC_SUBST([DBUS_TEST_DATA])
AC_SUBST([DBUS_TEST_EXEC])

AC_DEFINE_UNQUOTED([DBUS_TEST_EXEC], ["$DBUS_TEST_EXEC"],
        [Full path to the daemon in the builddir])
AC_DEFINE_UNQUOTED([DBUS_EXEEXT], ["$EXEEXT"],
        [Extension for executables, typically empty or
.exe])

AC_DEFINE_UNQUOTED(TEST_BUS_BINARY, ["$DBUS_PWD/bus/dbus-
daemon$EXEEXT"],
        [Full path to the daemon in the builddir])
AC_SUBST(TEST_BUS_BINARY)

## Export the non-setuid external helper
TEST_LAUNCH_HELPER_BINARY="$DBUS_PWD/bus/dbus-daemon-launch-helper-
test$EXEEXT"
AC_SUBST(TEST_LAUNCH_HELPER_BINARY)
AC_DEFINE_UNQUOTED(DBUS_TEST_LAUNCH_HELPER_BINARY,
"$TEST_LAUNCH_HELPER_BINARY",
        [Full path to the launch helper test program in the
builddir])

DEFAULT_SOCKET_DIR=/tmp

DEFAULT_SOCKET_DIR=`echo $DEFAULT_SOCKET_DIR | sed 's/+/%2B/g'`

if ! test -z "$with_test_socket_dir" ; then
    TEST_SOCKET_DIR="$with_test_socket_dir"
else
    TEST_SOCKET_DIR=$DEFAULT_SOCKET_DIR
fi
AC_SUBST(TEST_SOCKET_DIR)
AC_DEFINE_UNQUOTED(DBUS_TEST_SOCKET_DIR, "$TEST_SOCKET_DIR", [Where to
put test sockets])

if test "x$dbus_unix" = xyes; then
    TEST_LISTEN="unix:tmpdir=$TEST_SOCKET_DIR"
else
    TEST_LISTEN="tcp:host=localhost"
fi
AC_SUBST([TEST_LISTEN])
AC_DEFINE_UNQUOTED([TEST_LISTEN], ["$TEST_LISTEN"],

```



```

    [Listening address for regression tests])

if ! test -z "$with_session_socket_dir" ; then
    DBUS_SESSION_SOCKET_DIR="$with_session_socket_dir"
else
    DBUS_SESSION_SOCKET_DIR=$DEFAULT_SOCKET_DIR
fi
AC_DEFINE_UNQUOTED(DBUS_SESSION_SOCKET_DIR,
"$DBUS_SESSION_SOCKET_DIR", [Where per-session bus puts its sockets])
AC_SUBST(DBUS_SESSION_SOCKET_DIR)

if test x$dbus_win = xyes; then

DBUS_SESSION_BUS_DEFAULT_ADDRESS="$with_dbus_session_bus_default_address"
elif test x$have_launchd = xyes; then

DBUS_SESSION_BUS_DEFAULT_ADDRESS="launchd:env=DBUS_LAUNCHD_SESSION_BUS_SOCKET"
else

DBUS_SESSION_BUS_DEFAULT_ADDRESS="unix:tmpdir=$DBUS_SESSION_SOCKET_DIR"
fi
AC_SUBST(DBUS_SESSION_BUS_DEFAULT_ADDRESS)

# darwin needs this to initialize the environment
AC_CHECK_HEADERS(crt_externs.h)
AC_CHECK_FUNC(_NSGetEnviron, [AC_DEFINE(HAVE_NSGETENVIRON, 1, [Define if your system needs _NSGetEnviron to set up the environment])])
AH_VERBATIM(_DARWIN_ENVIRON,
[
#if defined(HAVE_NSGETENVIRON) && defined(HAVE_CRT_EXTERNS_H)
# include <sys/time.h>
# include <crt_externs.h>
# define environ (*_NSGetEnviron())
#endif
])

AC_ARG_ENABLE([stats],
[AS_HELP_STRING([--enable-stats],
[enable bus daemon usage statistics]),
[], [enable_stats=no])
if test "x$enable_stats" = xyes; then
    AC_DEFINE([DBUS_ENABLE_STATS], [1],
        [Define to enable bus daemon usage statistics])
fi

AC_CONFIG_FILES([
Doxyfile
dbus/versioninfo.rc
dbus/dbus-arch-deps.h

```

```
bus/system.conf
bus/session.conf
bus/messagebus
bus/messagebus-config
bus/org.freedesktop.dbus-session.plist
bus/rc.messagebus
bus/dbus.service
bus/dbus.socket
Makefile
dbus/Makefile
bus/Makefile
tools/Makefile
test/Makefile
test/name-test/Makefile
doc/Makefile
doc/dbus-daemon.1
dbus-1.pc
dbus-1-uninstalled.pc
test/data/valid-config-files/debug-allow-all.conf
test/data/valid-config-files/debug-allow-all-sha1.conf
test/data/valid-config-files-system/debug-allow-all-pass.conf
test/data/valid-config-files-system/debug-allow-all-fail.conf
test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service
])
AC_OUTPUT
```

dnl

=====
====

echo "

D-Bus \$VERSION

=====

prefix: \${EXPANDED_PREFIX}
exec_prefix: \${exec_prefix}
 libdir: \${EXPANDED_LIBDIR}
 libexecdir: \${EXPANDED_LIBEXECDIR}
 bindir: \${EXPANDED_BINDIR}
 sysconfdir: \${EXPANDED_SYSCONFDIR}
 localstatedir: \${EXPANDED_LOCALSTATEDIR}
datadir: \${EXPANDED_DATADIR}
source code location: \${srcdir}
compiler: \${CC}
cflags: \${CFLAGS}
cppflags: \${CPPFLAGS}
cxxflags: \${CXXFLAGS}
64-bit int: \${DBUS_INT64_TYPE}
32-bit int: \${DBUS_INT32_TYPE}
16-bit int: \${DBUS_INT16_TYPE}
 Doxygen: \${DOXYGEN:-not found}
 xmlto: \${XMLTO:-not found}
 man2html: \${MAN2HTML:-not found}"

echo "

Rebuilding generated files: \${USE_MAINTAINER_MODE}
gcc coverage profiling: \${enable_compiler_coverage}
Building embedded tests: \${enable_embedded_tests}
Building modular tests: \${enable_modular_tests}
 - with GLib: \${with_glib}
Building verbose mode: \${enable_verbose_mode}
Building assertions: \${enable_asserts}
Building checks: \${enable_checks}
Building bus stats API: \${enable_stats}
Building SELinux support: \${have_selinux}
Building inotify support: \${have_inotify}
Building dnotify support: \${have_dnotify}
Building kqueue support: \${have_kqueue}
Building systemd support: \${have_systemd}
Building X11 code: \${enable_x11}
Building Doxygen docs: \${enable_doxygen_docs}
Building XML docs: \${enable_xml_docs}
Building cache support: \${enable_userdb_cache}
Building launchd support: \${have_launchd}
Using XML parser: \${with_xml}
Init scripts style: \${with_init_scripts}
Abstract socket names: \${ac_cv_have_abstract_sockets}
System bus socket: \${DBUS_SYSTEM_SOCKET}
System bus address: \${DBUS_SYSTEM_BUS_DEFAULT_ADDRESS}

```

    System bus PID file:      ${DBUS_SYSTEM_PID_FILE}
    Session bus address:     ${DBUS_SESSION_BUS_DEFAULT_ADDRESS}
    Console auth dir:       ${DBUS_CONSOLE_AUTH_DIR}
    Console owner file:     ${have_console_owner_file}
    Console owner file path: ${DBUS_CONSOLE_OWNER_FILE}
    System bus user:        ${DBUS_USER}
    Session bus services dir: ${EXPANDED_DATADIR}/dbus-1/services
    'make check' socket dir: ${TEST_SOCKET_DIR}
"
if test x$have_launched = xyes; then
    echo "        launched agent dir:      ${LAUNCHD_AGENT_DIR}"
fi
echo

if test x$enable_embedded_tests = xyes; then
    echo "NOTE: building with unit tests increases the size of the
installed library and renders it insecure."
fi
if test x$enable_embedded_tests = xyes -a x$enable_asserts = xno; then
    echo "NOTE: building with embedded tests but without
assertions means tests may not properly report failures (this
configuration is only useful when doing something like profiling the
tests)"
fi
if test x$enable_compiler_coverage = xyes; then
    echo "NOTE: building with coverage profiling is definitely for
developers only."
fi
if test x$enable_verbose_mode = xyes; then
    echo "NOTE: building with verbose mode increases library size,
may slightly increase security risk, and decreases performance."
fi
if test x$enable_asserts = xyes; then
    echo "NOTE: building with assertions increases library size
and decreases performance."
fi
if test x$enable_checks = xno; then
    echo "NOTE: building without checks for arguments passed to
public API makes it harder to debug apps using D-Bus, but will
slightly decrease D-Bus library size and _very_ slightly improve
performance."
fi
if test x$dbus_use_libxml = xtrue; then
    echo
    echo "WARNING: You have chosen to use libxml as your xml parser
however this code path is not maintained by the D-Bus developers and
if it breaks you get to keep the pieces.  If you have selected this
option in err please reconfigure with expat (e.g. --with-xml=expat)."

```

This removes parts of the standard D-Bus API and ABI (the 't' and 'x' typecodes, the dbus_int64_t and dbus_uint64_t types, etc.) and should only be used if your compiler lacks support for 64-bit integers. Please report a bug with details of your platform and compiler.

This option is likely to be removed in future, unless the D-Bus developers receive reports that it is still needed.

```
])  
fi
```

```
File = configure.~1~
```

```
#!/bin/sh  
# Guess values for system-dependent variables and create Makefiles.  
# Generated by GNU Autoconf 2.69 for dbus 1.6.8.  
#  
# Report bugs to  
<https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus>.  
#  
#  
# Copyright (C) 1992-1996, 1998-2012 Free Software Foundation, Inc.  
#  
# This configure script is free software; the Free Software Foundation  
# gives unlimited permission to copy, distribute and modify it.  
## ----- ##  
## M4sh Initialization. ##  
## ----- ##  
  
# Be more Bourne compatible  
DUALCASE=1; export DUALCASE # for MKS sh  
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then  
:  
  emulate sh  
  NULLCMD=:  
  # Pre-4.2 versions of Zsh do word splitting on ${1+"$@"}, which  
  # is contrary to our usage.  Disable this feature.  
  alias -g '${1+"$@"}'='"$@"'  
  setopt NO_GLOB_SUBST  
else  
  case `(set -o) 2>/dev/null` in #(   
    *posix*) :  
      set -o posix ;; #(   
    *) :  
      ;;  
  esac
```



```

    }
fi

# IFS
# We need space, tab and new line, in precisely that order. Quoting
is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " "$as_nl"

# Find who we are. Look in the path if we contain no directory
separator.
as_myself=
case $0 in #((
    *[\ \/]* ) as_myself=$0 ;;
    *) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done
IFS=$as_save_IFS

    ;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
    as_myself=$0
fi
if test ! -f "$as_myself"; then
    $as_echo "$as_myself: error: cannot find myself; rerun with an
absolute file name" >&2
    exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
1"
# suppresses any "Segmentation fault" message there. ' ((' could
# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH
do eval test x\${$as_var+set} = xset \
    && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

```

```

# NLS nuisances.
LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# Use a proper internal environment variable to ensure we don't fall
# into an infinite loop, continuously re-executing ourselves.
if test x"${_as_can_reexec}" != xno && test "x$CONFIG_SHELL" != x;
then
    _as_can_reexec=no; export _as_can_reexec;
    # We cannot yet assume a decent shell, so we have to provide a
# neutralization value for shells without unset; and this also
# works around shells that cannot unset nonexistent variables.
# Preserve -v and -x to the replacement shell.
BASH_ENV=/dev/null
ENV=/dev/null
(unset BASH_ENV) >/dev/null 2>&1 && unset BASH_ENV ENV
case $- in # (((
    *v*x* | *x*v* ) as_opts=-vx ;;
    *v* ) as_opts=-v ;;
    *x* ) as_opts=-x ;;
    * ) as_opts= ;;
esac
exec $CONFIG_SHELL $as_opts "$as_myself" ${1+"$@"}
# Admittedly, this is quite paranoid, since all the known shells bail
# out after a failed `exec'.
$as_echo "$0: could not re-execute with $CONFIG_SHELL" >&2
as_fn_exit 255
fi
# We don't want this to propagate to other subprocesses.
    { _as_can_reexec; unset _as_can_reexec;}
if test "x$CONFIG_SHELL" = x; then
    as_bourne_compatible="if test -n \"\${ZSH_VERSION+set}\" && (emulate
sh) >/dev/null 2>&1; then :
    emulate sh
    NULLCMD=:
    # Pre-4.2 versions of Zsh do word splitting on \"\${1+\"\\$@\"}\", which
# is contrary to our usage. Disable this feature.
    alias -g \"\${1+\"\\$@\"}\"='\"\\$@\"'
    setopt NO_GLOB_SUBST
else
    case \" `(set -o) 2>/dev/null\" in # (
    *posix*) :
        set -o posix ;; #(
    *) :
        ;;
esac

```



```

else
  as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
as_found=false
for as_dir in /bin$PATH_SEPARATOR/usr/bin$PATH_SEPARATOR$PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  as_found=:
  case $as_dir in #(
    /*)
      for as_base in sh bash ksh sh5; do
        # Try only shells that exist, to save several forks.
        as_shell=$as_dir/$as_base
        if { test -f "$as_shell" || test -f "$as_shell.exe"; } &&
            { $as_echo "$as_bourne_compatible"$as_required" |
as_run=a "$as_shell"; } 2>/dev/null; then :
          CONFIG_SHELL=$as_shell as_have_required=yes
            if { $as_echo "$as_bourne_compatible"$as_suggested" |
as_run=a "$as_shell"; } 2>/dev/null; then :
              break 2
            fi
          fi
        done;;
      esac
    as_found=false
  done
  $as_found || { if { test -f "$SHELL" || test -f "$SHELL.exe"; } &&
      { $as_echo "$as_bourne_compatible"$as_required" | as_run=a
"$SHELL"; } 2>/dev/null; then :
    CONFIG_SHELL=$SHELL as_have_required=yes
    fi; }
  IFS=$as_save_IFS

  if test "x$CONFIG_SHELL" != x; then :
    export CONFIG_SHELL
      # We cannot yet assume a decent shell, so we have to
provide a
# neutralization value for shells without unset; and this also
# works around shells that cannot unset nonexistent variables.
# Preserve -v and -x to the replacement shell.
BASH_ENV=/dev/null
ENV=/dev/null
(unset BASH_ENV) >/dev/null 2>&1 && unset BASH_ENV ENV
case $- in # (((
  *v*x* | *x*v* ) as_opts=-vx ;;
  *v* ) as_opts=-v ;;
  *x* ) as_opts=-x ;;
  * ) as_opts= ;;
esac
exec $CONFIG_SHELL $as_opts "$as_myself" ${1+"$@"}
# Admittedly, this is quite paranoid, since all the known shells bail

```

```

# out after a failed `exec`.
$as_echo "$0: could not re-execute with $CONFIG_SHELL" >&2
exit 255
fi

    if test x$as_have_required = xno; then :
    $as_echo "$0: This script requires a shell more modern than all"
    $as_echo "$0: the shells that I found on your system."
    if test x${ZSH_VERSION+set} = xset ; then
        $as_echo "$0: In particular, zsh $ZSH_VERSION has bugs and should"
        $as_echo "$0: be upgraded to zsh 4.3.4 or later."
    else
        $as_echo "$0: Please tell bug-autoconf@gnu.org and
$0: https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus
$0: about your system, including any error possibly output
$0: before this message. Then install a modern shell, or
$0: manually run the script under such a shell if you do
$0: have one."
        fi
    exit 1
    fi
fi
fi
fi
SHELL=${CONFIG_SHELL-/bin/sh}
export SHELL
# Unset more variables known to interfere with behavior of common
tools.
CLICOLOR_FORCE= GREP_OPTIONS=
unset CLICOLOR_FORCE GREP_OPTIONS

## ----- ##
## M4sh Shell Functions. ##
## ----- ##
# as_fn_unset VAR
# -----
# Portably unset VAR.
as_fn_unset ()
{
    { eval $1=; unset $1;}
}
as_unset=as_fn_unset

# as_fn_set_status STATUS
# -----
# Set $? to STATUS, without forking.
as_fn_set_status ()
{
    return $1
} # as_fn_set_status

# as_fn_exit STATUS
# -----

```

```

# Exit the shell with STATUS, even in a "trap 0" or "set -e" context.
as_fn_exit ()
{
    set +e
    as_fn_set_status $1
    exit $1
} # as_fn_exit

# as_fn_mkdir_p
# -----
# Create "$as_dir" as a directory, including parents if necessary.
as_fn_mkdir_p ()
{
    case $as_dir in #(
    -*) as_dir=./$as_dir;;
    esac
    test -d "$as_dir" || eval $as_mkdir_p || {
        as_dirs=
        while :; do
            case $as_dir in #(
            *\'*) as_qdir=`$as_echo "$as_dir" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'/g"`;;
            #'(
            *) as_qdir=$as_dir;;
            esac
            as_dirs="'$as_qdir' $as_dirs"
            as_dir=`$as_dirname -- "$as_dir" ||
$as_expr X"$as_dir" : 'X\([^\/\\\]\)\/*\([^\/\\\]\)\/*\*' \|| \
    X"$as_dir" : 'X\([^\/\\\]\)\*' \|| \
    X"$as_dir" : 'X\([^\/\\\]\)\$' \|| \
    X"$as_dir" : 'X\([^\/\\\]\)' \|| . 2>/dev/null ||
$as_echo X"$as_dir" |
        sed '/^X\([^\/\\\]\)\*\/*\([^\/\\\]\)\*\/*\*$/ {
            s//\1/
            q
        }
/^X\([^\/\\\]\)\*$/ {
            s//\1/
            q
        }
/^X\([^\/\\\]\)\$/ {
            s//\1/
            q
        }
/^X\([^\/\\\]\)\.$'/{
            s//\1/
            q
        }
        s/././; q'`
            test -d "$as_dir" && break
        done
        test -z "$as_dirs" || eval "mkdir $as_dirs"
    }
}

```

```

    } || test -d "$sas_dir" || as_fn_error $? "cannot create directory
    $sas_dir"

} # as_fn_mkdir_p

# as_fn_executable_p FILE
# -----
# Test if FILE is an executable regular file.
as_fn_executable_p ()
{
    test -f "$1" && test -x "$1"
} # as_fn_executable_p
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
# VAR. Take
# advantage of any shell optimizations that allow amortized linear
# growth over
# repeated appends, instead of the typical quadratic growth present in
# naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\$as_var = x12") 2>/dev/null;
then :
    eval 'as_fn_append ()
    {
        eval $1+=\$2
    }'
else
    as_fn_append ()
    {
        eval $1=\${$1}\$2
    }
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
# the
# global $sas_val. Take advantage of shells that can avoid forks. The
# arguments
# must be portable across $(( )) and expr.
if (eval "test \${(( 1 + 1 ))} = 2") 2>/dev/null; then :
    eval 'as_fn_arith ()
    {
        as_val=$(( $* ))
    }'
else
    as_fn_arith ()
    {
        as_val=`expr "$@" || test $? -eq 1`
    }

```

```

fi # as_fn_arith

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
are
# provided, also output the error to LOG_FD, referencing LINENO. Then
exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
    as_status=$1; test $as_status -eq 0 && as_status=1
    if test "$4"; then
        as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
        $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
    fi
    $as_echo "$as_me: error: $2" >&2
    as_fn_exit $as_status
} # as_fn_error

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
"X/"; then
    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\/\([^\/]*\)'/*$' \|\ \
X"$0" : 'X\(/\)\$' \|\ \
X"$0" : 'X\(/)\$' \|\ \. 2>/dev/null ||
$as_echo X/"$0" |
sed '/^\.*\/\([^\/]*\)\/*$/{
s//\1/
q
}'

```

```

/^X\\(\\|\\|\\|)${/
  s//\\1/
  q
}
/^X\\(\\|\\|)\\.*/{
  s//\\1/
  q
}
s/\\.*/./; q'`

```

```

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyZ'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

```

```

as_lineno_1=$LINENO as_lineno_1a=$LINENO
as_lineno_2=$LINENO as_lineno_2a=$LINENO
eval 'test "x$as_lineno_1'$as_run'" != "x$as_lineno_2'$as_run'" &&
test "x`expr $as_lineno_1'$as_run' + 1`" = "x$as_lineno_2'$as_run'"'
|| {
# Blame Lee E. McMahon (1931-1989) for sed's syntax.  :-)
sed -n '
  p
  /[$]LINENO/=
  ' <$as_myself |
  sed '
    s/[$]LINENO.*/&-/
    t lineno
    b
    :lineno
    N
    :loop
    s/[$]LINENO\([^'$as_cr_alnum'_]*\n\)\(.*\)/\2\1\2/
    t loop
    s/-\n.*//
  ' >$as_me.lineno &&
chmod +x "$as_me.lineno" ||
{ $as_echo "$as_me: error: cannot create $as_me.lineno; rerun with
a POSIX shell" >&2; as_fn_exit 1; }

```

```

# If we had to re-execute with $CONFIG_SHELL, we're ensured to have
# already done that, so ensure we don't try to do so again and fall
# in an infinite loop. This has already happened in practice.
_as_can_reexec=no; export _as_can_reexec
# Don't try to exec as it changes $[0], causing all sort of problems
# (the dirname of $[0] is not the place where we might find the
# original and so on. Autoconf is especially sensitive to this).
. "$as_me.lineno"
# Exit status is that of the last command.

```

```

    exit
}

ECHO_C= ECHO_N= ECHO_T=
case `echo -n x` in #((((
-n*)
    case `echo 'xy\c'` in
    *c*) ECHO_T=' ';; # ECHO_T is single tab character.
    xy) ECHO_C='\c';;
    *) echo `echo ksh88 bug on AIX 6.1` > /dev/null
        ECHO_T=' ';;
    esac;;
*)
    ECHO_N='-n';;
esac

rm -f conf$$$ conf$$$exe conf$$$file
if test -d conf$$$dir; then
    rm -f conf$$$dir/conf$$$file
else
    rm -f conf$$$dir
    mkdir conf$$$dir 2>/dev/null
fi
if (echo >conf$$$file) 2>/dev/null; then
    if ln -s conf$$$file conf$$$ 2>/dev/null; then
        as_ln_s='ln -s'
        # ... but there are two gotchas:
        # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
        # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
        executable.
        # In both cases, we have to default to `cp -pR'.
        ln -s conf$$$file conf$$$dir 2>/dev/null && test ! -f conf$$$exe
    ||
        as_ln_s='cp -pR'
    elif ln conf$$$file conf$$$ 2>/dev/null; then
        as_ln_s=ln
    else
        as_ln_s='cp -pR'
    fi
else
    as_ln_s='cp -pR'
fi
rm -f conf$$$ conf$$$exe conf$$$dir/conf$$$file conf$$$file
rmdir conf$$$dir 2>/dev/null

if mkdir -p . 2>/dev/null; then
    as_mkdir_p='mkdir -p "$as_dir"'
else
    test -d ./-p && rmdir ./-p
    as_mkdir_p=false
fi

```



```

as_test_x='test -x'
as_executable_p=as_fn_executable_p

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed
'y%*$as_cr_letters%P$as_cr_LETTERS%;s%[^_$as_cr_alnum]%%_g'"

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_$as_cr_alnum]%%_g'"

SHELL=${CONFIG_SHELL-/bin/sh}

test -n "$DJDIR" || exec 7<&0 </dev/null
exec 6>&1

# Name of the host.
# hostname on some systems (SVR3.2, old GNU/Linux) returns a bogus
exit status,
# so uname gets run too.
ac_hostname=`(hostname || uname -n) 2>/dev/null | sed 1q`

#
# Initializations.
#
ac_default_prefix=/usr/local
ac_clean_files=
ac_config_libobj_dir=.
LIBOBJS=
cross_compiling=no
subdirs=
MFLAGS=
MAKEFLAGS=

# Identity of this package.
PACKAGE_NAME='dbus'
PACKAGE_TARNAME='dbus'
PACKAGE_VERSION='1.6.8'
PACKAGE_STRING='dbus 1.6.8'
PACKAGE_BUGREPORT='https://bugs.freedesktop.org/enter_bug.cgi?product=
dbus'
PACKAGE_URL=''

# Factoring default headers for most tests.
ac_includes_default="\
#include <stdio.h>
#ifdef HAVE_SYS_TYPES_H
# include <sys/types.h>
#endif
#ifdef HAVE_SYS_STAT_H
# include <sys/stat.h>
#endif

```

```
#ifndef STDC_HEADERS
# include <stdlib.h>
# include <stddef.h>
#else
# ifdef HAVE_STDLIB_H
# include <stdlib.h>
# endif
#endif
#ifdef HAVE_STRING_H
# if !defined STDC_HEADERS && defined HAVE_MEMORY_H
# include <memory.h>
# endif
# include <string.h>
#endif
#ifdef HAVE_STRINGS_H
# include <strings.h>
#endif
#ifdef HAVE_INTTYPES_H
# include <inttypes.h>
#endif
#ifdef HAVE_STDINT_H
# include <stdint.h>
#endif
#ifdef HAVE_UNISTD_H
# include <unistd.h>
#endif"
```

```
ac_subst_vars='am__EXEEXT_FALSE
am__EXEEXT_TRUE
LTLIBOBJS
LIBOBJS
DBUS_SESSION_BUS_DEFAULT_ADDRESS
DBUS_SESSION_SOCKET_DIR
TEST_LISTEN
TEST_SOCKET_DIR
TEST_LAUNCH_HELPER_BINARY
TEST_BUS_BINARY
DBUS_TEST_EXEC
DBUS_TEST_DATA
DBUS_LIBEXECDIR
DBUS_BINDIR
DBUS_DAEMONDIR
DBUS_DATADIR
DBUS_PREFIX
DBUS_USER
DBUS_CONSOLE_OWNER_FILE
DBUS_CONSOLE_AUTH_DIR
DBUS_SYSTEM_PID_FILE
DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
DBUS_SYSTEM_SOCKET
HAVE_SYSTEMD_FALSE
HAVE_SYSTEMD_TRUE'
```

systemdsystemunitdir
DBUS_INIT_SCRIPTS_CYGWIN_FALSE
DBUS_INIT_SCRIPTS_CYGWIN_TRUE
DBUS_INIT_SCRIPTS_SLACKWARE_FALSE
DBUS_INIT_SCRIPTS_SLACKWARE_TRUE
DBUS_INIT_SCRIPTS_RED_HAT_FALSE
DBUS_INIT_SCRIPTS_RED_HAT_TRUE
EXPANDED_DATADIR
EXPANDED_LIBEXECDIR
EXPANDED_LIBDIR
EXPANDED_BINDIR
EXPANDED_SYSCONFDIR
EXPANDED_LOCALSTATEDIR
EXPANDED_PREFIX
DBUS_CAN_UPLOAD_DOCS_FALSE
DBUS_CAN_UPLOAD_DOCS_TRUE
DBUS_HAVE_MAN2HTML_FALSE
DBUS_HAVE_MAN2HTML_TRUE
MAN2HTML
DBUS_XML_DOCS_ENABLED_FALSE
DBUS_XML_DOCS_ENABLED_TRUE
XMLTO
DBUS_HAVE_XSLTPROC_FALSE
DBUS_HAVE_XSLTPROC_TRUE
XSLTPROC
DBUS_DOXYGEN_DOCS_ENABLED_FALSE
DBUS_DOXYGEN_DOCS_ENABLED_TRUE
DOXYGEN
DBUS_X_LIBS
DBUS_X_CFLAGS
X_EXTRA_LIBS
X_LIBS
X_PRE_LIBS
X_CFLAGS
LIBDBUS_LIBS
VALGRIND_LIBS
VALGRIND_CFLAGS
NETWORK_libs
ADT_LIBS
SELINUX_LIBS
HAVE_LIBAUDIT_FALSE
HAVE_LIBAUDIT_TRUE
SYSTEMD_LIBS
SYSTEMD_CFLAGS
HAVE_CONSOLE_OWNER_FILE_FALSE
HAVE_CONSOLE_OWNER_FILE_TRUE
LAUNCHD_AGENT_DIR
DBUS_ENABLE_LAUNCHD_FALSE
DBUS_ENABLE_LAUNCHD_TRUE
LAUNCHCTL
DBUS_BUS_ENABLE_KQUEUE_FALSE
DBUS_BUS_ENABLE_KQUEUE_TRUE

HAVE_LINUX_EPOLL_FALSE
HAVE_LINUX_EPOLL_TRUE
DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE
DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE
DBUS_BUS_ENABLE_INOTIFY_FALSE
DBUS_BUS_ENABLE_INOTIFY_TRUE
HAVE_SELINUX_FALSE
HAVE_SELINUX_TRUE
THREAD_LIBS
XML_LIBS
XML_CFLAGS
DBUS_USE_LIBXML_FALSE
DBUS_USE_LIBXML_TRUE
DBUS_USE_EXPAT_FALSE
DBUS_USE_EXPAT_TRUE
LIBXML_LIBS
LIBXML_CFLAGS
DBUS_PATH_OR_ABSTRACT
DBUS_INT16_TYPE
DBUS_INT32_TYPE
DBUS_HAVE_INT64
DBUS_UINT64_CONSTANT
DBUS_INT64_CONSTANT
DBUS_INT64_TYPE
R_DYNAMIC_LDFLAG
pkgpyexecdir
pyexecdir
pkgpythondir
pythondir
PYTHON_PLATFORM
PYTHON_EXEC_PREFIX
PYTHON_LIB_PREFIX
PYTHON_PREFIX
PYTHON_VERSION
PYTHON
DBUS_ENABLE_INSTALLED_TESTS_FALSE
DBUS_ENABLE_INSTALLED_TESTS_TRUE
DBUS_WITH_GLIB_FALSE
DBUS_WITH_GLIB_TRUE
DBUS_ENABLE_MODULAR_TESTS_FALSE
DBUS_ENABLE_MODULAR_TESTS_TRUE
DBUS_GLIB_LIBS
DBUS_GLIB_CFLAGS
GLIB_LIBS
GLIB_CFLAGS
DBUS_ENABLE_EMBEDDED_TESTS_FALSE
DBUS_ENABLE_EMBEDDED_TESTS_TRUE
DBUS_BUILD_TESTS_FALSE
DBUS_BUILD_TESTS_TRUE
DBUS_STATIC_BUILD_CPPFLAGS
DBUS_CYGWIN_FALSE
DBUS_CYGWIN_TRUE

DBUS_UNIX_FALSE
DBUS_UNIX_TRUE
DBUS_WINCE_FALSE
DBUS_WINCE_TRUE
DBUS_WIN_FALSE
DBUS_WIN_TRUE
WINDRES
BUILD_FILEVERSION
BUILD_TIMESTAMP
RC
PKG_CONFIG
CXXCPP
OTOOL64
OTOOL
LIPO
NMEDIT
DSYMUTIL
MANIFEST_TOOL
RANLIB
ac_ct_AR
AR
DLLTOOL
OBJDUMP
LN_S
NM
ac_ct_DUMPBIN
DUMPBIN
LD
FGREP
SED
LIBTOOL
EGREP
GREP
CPP
am__fastdepCXX_FALSE
am__fastdepCXX_TRUE
CXXDEPMODE
ac_ct_CXX
CXXFLAGS
CXX
am__fastdepCC_FALSE
am__fastdepCC_TRUE
CCDEPMODE
am__nodep
AMDEPBACKSLASH
AMDEP_FALSE
AMDEP_TRUE
am__quote
am__include
DEPDIR
OBJEXT
EXEEXT

ac_ct_CC
CPPFLAGS
LDFLAGS
CFLAGS
CC
DBUS_VERSION
DBUS_MICRO_VERSION
DBUS_MINOR_VERSION
DBUS_MAJOR_VERSION
LT_AGE
LT_REVISION
LT_CURRENT
AM_BACKSLASH
AM_DEFAULT_VERBOSITY
AM_DEFAULT_V
AM_V
MAINT
MAINTAINER_MODE_FALSE
MAINTAINER_MODE_TRUE
GETTEXT_PACKAGE
am__untar
am__tar
AMTAR
am__leading_dot
SET_MAKE
AWK
mkdir_p
MKDIR_P
INSTALL_STRIP_PROGRAM
STRIP
install_sh
MAKEINFO
AUTOHEADER
AUTOMAKE
AUTOCONF
ACLOCAL
VERSION
PACKAGE
CYGPATH_W
am__isrc
INSTALL_DATA
INSTALL_SCRIPT
INSTALL_PROGRAM
host_os
host_vendor
host_cpu
host
build_os
build_vendor
build_cpu
build
target_alias

host_alias
build_alias
LIBS
ECHO_T
ECHO_N
ECHO_C
DEFS
mandir
localedir
libdir
psdir
pdfdir
dvidir
htmldir
infodir
docdir
oldincludedir
includedir
localstatedir
sharedstatedir
sysconfdir
datadir
datarootdir
libexecdir
sbindir
bindir
program_transform_name
prefix
exec_prefix
PACKAGE_URL
PACKAGE_BUGREPORT
PACKAGE_STRING
PACKAGE_VERSION
PACKAGE_TARNAME
PACKAGE_NAME
PATH_SEPARATOR
SHELL'
ac_subst_files=''
ac_user_opts='
enable_option_checking
enable_maintainer_mode
enable_silent_rules
enable_dependency_tracking
enable_shared
enable_static
with_pic
enable_fast_install
with_gnu_ld
with_libtool_sysroot
enable_libtool_lock
enable_compiler_coverage
enable_compiler_optimisations

```
enable_developer
enable_ansi
enable_verbose_mode
enable_asserts
enable_checks
enable_xml_docs
enable_doxygen_docs
enable_abstract_sockets
enable_selinux
enable_libaudit
enable_dnotify
enable_inotify
enable_kqueue
enable_console_owner_file
enable_userdb_cache
enable_launchd
enable_systemd
with_xml
with_init_scripts
with_session_socket_dir
with_test_socket_dir
with_system_pid_file
with_system_socket
with_console_auth_dir
with_console_owner_file
with_launchd_agent_dir
with_dbus_user
with_dbus_daemondir
with_dbus_session_bus_default_address
enable_embedded_tests
enable_modular_tests
enable_tests
enable_installed_tests
with_64_bit
enable_epoll
with_valgrind
enable_x11_autolaunch
with_x
enable_Werror
with_systemdsystemunitdir
with_dbus_test_dir
enable_stats
'
    ac_precious_vars='build_alias
host_alias
target_alias
CC
CFLAGS
LDFLAGS
LIBS
CPPFLAGS
CXX
```



```
CXXFLAGS
CCC
CPP
CXXCPP
PKG_CONFIG
GLIB_CFLAGS
GLIB_LIBS
DBUS_GLIB_CFLAGS
DBUS_GLIB_LIBS
PYTHON
LIBXML_CFLAGS
LIBXML_LIBS
SYSTEMD_CFLAGS
SYSTEMD_LIBS
VALGRIND_CFLAGS
VALGRIND_LIBS
MAN2HTML'
```

```
# Initialize some variables set by options.
```

```
ac_init_help=
ac_init_version=false
ac_unrecognized_opts=
ac_unrecognized_sep=
# The variables have the same names as the options, with
# dashes changed to underlines.
cache_file=/dev/null
exec_prefix=NONE
no_create=
no_recursion=
prefix=NONE
program_prefix=NONE
program_suffix=NONE
program_transform_name=s,x,x,
silent=
site=
srcdir=
verbose=
x_includes=NONE
x_libraries=NONE
```

```
# Installation directory options.
```

```
# These are left unexpanded so users can "make install
exec_prefix=/foo"
# and all the variables that are supposed to be based on exec_prefix
# by default will actually change.
# Use braces instead of parens because sh, perl, etc. also accept
them.
# (The list follows the same order as the GNU Coding Standards.)
bindir='${exec_prefix}/bin'
sbindir='${exec_prefix}/sbin'
libexecdir='${exec_prefix}/libexec'
```

```

datarootdir='${prefix}/share'
datadir='${datarootdir}'
sysconfdir='${prefix}/etc'
sharedstatedir='${prefix}/com'
localstatedir='${prefix}/var'
includedir='${prefix}/include'
oldincludedir='/usr/include'
docdir='${datarootdir}/doc/${PACKAGE_TARNAME}'
infodir='${datarootdir}/info'
htmldir='${docdir}'
dvidir='${docdir}'
pdfdir='${docdir}'
psdir='${docdir}'
libdir='${exec_prefix}/lib'
localedir='${datarootdir}/locale'
mandir='${datarootdir}/man'

ac_prev=
ac_dashdash=
for ac_option
do
  # If the previous option needs an argument, assign it.
  if test -n "$ac_prev"; then
    eval $ac_prev=\$ac_option
    ac_prev=
    continue
  fi

  case $ac_option in
    *=?*) ac_optarg=`expr "X$ac_option" : '[^=]*=\(.*\)'` ;;
    *=)   ac_optarg= ;;
    *)    ac_optarg=yes ;;
  esac

  # Accept the important Cygnus configure options, so we can diagnose
  typos.

  case $ac_dashdash$ac_option in
    --)
      ac_dashdash=yes ;;

    -bindir | --bindir | --bindi | --bind | --bin | --bi)
      ac_prev=bindir ;;
    -bindir=* | --bindir=* | --bindi=* | --bind=* | --bin=* | --bi=*)
      bindir=$ac_optarg ;;

    -build | --build | --buil | --bui | --bu)
      ac_prev=build_alias ;;
    -build=* | --build=* | --buil=* | --bui=* | --bu=*)
      build_alias=$ac_optarg ;;

    -cache-file | --cache-file | --cache-fil | --cache-fi \

```

```

| --cache-f | --cache- | --cache | --cach | --cac | --ca | --c)
  ac_prev=cache_file ;;
-cache-file=* | --cache-file=* | --cache-fil=* | --cache-fi=* \
| --cache-f=* | --cache-=* | --cache=* | --cach=* | --cac=* | --ca=*
| --c=*)
  cache_file=$ac_optarg ;;

--config-cache | -C)
  cache_file=config.cache ;;

-datadir | --datadir | --datadi | --datad)
  ac_prev=datadir ;;
-datadir=* | --datadir=* | --datadi=* | --datad=*)
  datadir=$ac_optarg ;;

-datarootdir | --datarootdir | --datarootdi | --datarootd | --
dataroot \
| --dataroo | --dataro | --datar)
  ac_prev=datarootdir ;;
-datarootdir=* | --datarootdir=* | --datarootdi=* | --datarootd=* \
| --dataroot=* | --dataroo=* | --dataro=* | --datar=*)
  datarootdir=$ac_optarg ;;

-disable-* | --disable-*)
  ac_useropt=`expr "x$ac_option" : 'x-*disable-\(.*\)'\`
  # Reject names that are not valid shell variable names.
  expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
  as_fn_error $? "invalid feature name: $ac_useropt"
  ac_useropt_orig=$ac_useropt
  ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'\`
  case $ac_user_opts in
    *)
"enable_$ac_useropt"
"*) ;;
    *)
ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--
disable-$ac_useropt_orig"
  ac_unrecognized_sep=', ';;
  esac
  eval enable_$ac_useropt=no ;;

-docdir | --docdir | --docdi | --doc | --do)
  ac_prev=docdir ;;
-docdir=* | --docdir=* | --docdi=* | --doc=* | --do=*)
  docdir=$ac_optarg ;;

-dvidir | --dvidir | --dvidi | --dvid | --dvi | --dv)
  ac_prev=dvidir ;;
-dvidir=* | --dvidir=* | --dvidi=* | --dvid=* | --dvi=* | --dv=*)
  dvidir=$ac_optarg ;;

-enable-* | --enable-*)

```

```

ac_useropt=`expr "x${ac_option}" : 'x-*enable-\([^=]*\)'`
# Reject names that are not valid shell variable names.
expr "x${ac_useropt}" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
  as_fn_error $? "invalid feature name: ${ac_useropt}"
ac_useropt_orig=${ac_useropt}
ac_useropt=`$as_echo "${ac_useropt}" | sed 's/[-+.]/_/g'`
case $ac_user_opts in
  *)
"enable_${ac_useropt}"
  *) ;;
  *)
ac_unrecognized_opts="$ac_unrecognized_opts${ac_unrecognized_sep}--
enable-${ac_useropt}_orig"
  ac_unrecognized_sep=', ';;
  esac
  eval enable_${ac_useropt}=\${ac_optarg} ;;

--exec-prefix | --exec_prefix | --exec-prefix | --exec-prefi \
| --exec-pref | --exec-pre | --exec-pr | --exec-p | --exec- \
| --exec | --exe | --ex)
  ac_prev=exec_prefix ;;
--exec-prefix=* | --exec_prefix=* | --exec-prefix=* | --exec-prefi=*
\
| --exec-pref=* | --exec-pre=* | --exec-pr=* | --exec-p=* | --exec-
=* \
| --exec=* | --exe=* | --ex=*)
  exec_prefix=${ac_optarg} ;;

--gas | --gas | --ga | --g)
  # Obsolete; use --with-gas.
  with_gas=yes ;;

--help | --help | --hel | --he | -h)
  ac_init_help=long ;;
--help=r* | --help=r* | --hel=r* | --he=r* | -hr*)
  ac_init_help=recursive ;;
--help=s* | --help=s* | --hel=s* | --he=s* | -hs*)
  ac_init_help=short ;;

--host | --host | --hos | --ho)
  ac_prev=host_alias ;;
--host=* | --host=* | --hos=* | --ho=*)
  host_alias=${ac_optarg} ;;

--htmldir | --htmldir | --htmldi | --htmld | --html | --htm | --ht)
  ac_prev=htmldir ;;
--htmldir=* | --htmldir=* | --htmldi=* | --htmld=* | --html=* | --
htm=* \
| --ht=*)
  htmldir=${ac_optarg} ;;

--includedir | --includedir | --includedi | --included | --include \

```

```

| --includ | --inclu | --incl | --inc)
  ac_prev=includedir ;;
-includedir=* | --includedir=* | --includedi=* | --included=* | --
include=* \
| --includ=* | --inclu=* | --incl=* | --inc=*)
  includedir=$ac_optarg ;;

-infodir | --infodir | --infodi | --infod | --info | --inf)
  ac_prev=infodir ;;
-infodir=* | --infodir=* | --infodi=* | --infod=* | --info=* | --
inf=*)
  infodir=$ac_optarg ;;

-libdir | --libdir | --libdi | --libd)
  ac_prev=libdir ;;
-libdir=* | --libdir=* | --libdi=* | --libd=*)
  libdir=$ac_optarg ;;

-libexecdir | --libexecdir | --libexecdi | --libexecd | --libexec \
| --libexe | --libex | --libe)
  ac_prev=libexecdir ;;
-libexecdir=* | --libexecdir=* | --libexecdi=* | --libexecd=* | --
libexec=* \
| --libexe=* | --libex=* | --libe=*)
  libexecdir=$ac_optarg ;;

-localedir | --localedir | --localedi | --localed | --locale)
  ac_prev=localedir ;;
-localedir=* | --localedir=* | --localedi=* | --localed=* | --
locale=*)
  localedir=$ac_optarg ;;

-localstatedir | --localstatedir | --localstatedi | --localstated \
| --localstate | --localstat | --localsta | --localst | --locals)
  ac_prev=localstatedir ;;
-localstatedir=* | --localstatedir=* | --localstatedi=* | --
localstated=* \
| --localstate=* | --localstat=* | --localsta=* | --localst=* | --
locals=*)
  localstatedir=$ac_optarg ;;

-mandir | --mandir | --mandi | --mand | --man | --ma | --m)
  ac_prev=mandir ;;
-mandir=* | --mandir=* | --mandi=* | --mand=* | --man=* | --ma=* | -
-m=*)
  mandir=$ac_optarg ;;

-nfp | --nfp | --nf)
  # Obsolete; use --without-fp.
  with_fp=no ;;

-no-create | --no-create | --no-creat | --no-crea | --no-cre \

```

```

| --no-cr | --no-c | -n)
  no_create=yes ;;

-no-recursion | --no-recursion | --no-recursio | --no-recursi \
| --no-recurs | --no-recur | --no-recu | --no-rec | --no-re | --no-
r)
  no_recursion=yes ;;

-oldincludedir | --oldincludedir | --oldincludedi | --oldincluded \
| --oldinclude | --oldinclud | --oldinclu | --oldincl | --oldinc \
| --oldin | --oldi | --old | --ol | --o)
  ac_prev=oldincludedir ;;
-oldincludedir=* | --oldincludedir=* | --oldincludedi=* | --
oldincluded=* \
| --oldinclude=* | --oldinclud=* | --oldinclu=* | --oldincl=* | --
oldinc=* \
| --oldin=* | --oldi=* | --old=* | --ol=* | --o=*)
  oldincludedir=$ac_optarg ;;

-prefix | --prefix | --prefi | --pref | --pre | --pr | --p)
  ac_prev=prefix ;;
-prefix=* | --prefix=* | --prefi=* | --pref=* | --pre=* | --pr=* | -
-p=*)
  prefix=$ac_optarg ;;

-program-prefix | --program-prefix | --program-prefi | --program-
pref \
| --program-pre | --program-pr | --program-p)
  ac_prev=program_prefix ;;
-program-prefix=* | --program-prefix=* | --program-prefi=* \
| --program-pref=* | --program-pre=* | --program-pr=* | --program-
p=*)
  program_prefix=$ac_optarg ;;

-program-suffix | --program-suffix | --program-suffi | --program-
suff \
| --program-suf | --program-su | --program-s)
  ac_prev=program_suffix ;;
-program-suffix=* | --program-suffix=* | --program-suffi=* \
| --program-suff=* | --program-suf=* | --program-su=* | --program-
s=*)
  program_suffix=$ac_optarg ;;

-program-transform-name | --program-transform-name \
| --program-transform-nam | --program-transform-na \
| --program-transform-n | --program-transform- \
| --program-transform | --program-transfor \
| --program-transfo | --program-transf \
| --program-trans | --program-tran \
| --progr-tra | --program-tr | --program-t)
  ac_prev=program_transform_name ;;
-program-transform-name=* | --program-transform-name=* \

```

```

| --program-transform-nam=* | --program-transform-na=* \
| --program-transform-n=* | --program-transform-==* \
| --program-transform=* | --program-transfor=* \
| --program-transfo=* | --program-transf=* \
| --program-trans=* | --program-tran=* \
| --progr-tra=* | --program-tr=* | --program-t=*)
  program_transform_name=$ac_optarg ;;

-pdfdir | --pdfdir | --pdfdi | --pdfd | --pdf | --pd)
  ac_prev=pdfdir ;;
-pdfdir=* | --pdfdir=* | --pdfdi=* | --pdfd=* | --pdf=* | --pd=*)
  pdfdir=$ac_optarg ;;

-psdir | --psdir | --psdi | --psd | --ps)
  ac_prev=psdir ;;
-psdir=* | --psdir=* | --psdi=* | --psd=* | --ps=*)
  psdir=$ac_optarg ;;

-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil)
  silent=yes ;;

-sbindir | --sbindir | --sbindi | --sbind | --sbin | --sbi | --sb)
  ac_prev=sbindir ;;
-sbindir=* | --sbindir=* | --sbindi=* | --sbind=* | --sbin=* \
| --sbi=* | --sb=*)
  sbindir=$ac_optarg ;;

-sharedstatedir | --sharedstatedir | --sharedstatedi \
| --sharedstated | --sharedstate | --sharedstat | --sharedsta \
| --sharedst | --shares | --shared | --share | --shar \
| --sha | --sh)
  ac_prev=sharedstatedir ;;
-sharedstatedir=* | --sharedstatedir=* | --sharedstatedi=* \
| --sharedstated=* | --sharedstate=* | --sharedstat=* | --
sharedsta=* \
| --sharedst=* | --shares=* | --shared=* | --share=* | --shar=* \
| --sha=* | --sh=*)
  sharedstatedir=$ac_optarg ;;

-site | --site | --sit)
  ac_prev=site ;;
-site=* | --site=* | --sit=*)
  site=$ac_optarg ;;

-srcdir | --srcdir | --srcdi | --srcd | --src | --sr)
  ac_prev=srcdir ;;
-srcdir=* | --srcdir=* | --srcdi=* | --srcd=* | --src=* | --sr=*)
  srcdir=$ac_optarg ;;

-sysconfdir | --sysconfdir | --sysconfdi | --sysconfd | --sysconf \
| --syscon | --sysco | --sysc | --sys | --sy)

```

```

    ac_prev=sysconfdir ;;
    -sysconfdir=* | --sysconfdir=* | --sysconfdi=* | --sysconfd=* | --
sysconf=* \
    | --syscon=* | --sysco=* | --sysc=* | --sys=* | --sy=*)
    sysconfdir=$ac_optarg ;;

    -target | --target | --targe | --targ | --tar | --ta | --t)
    ac_prev=target_alias ;;
    -target=* | --target=* | --targe=* | --targ=* | --tar=* | --ta=* | -
-t=*)
    target_alias=$ac_optarg ;;

    -v | -verbose | --verbose | --verbos | --verbo | --verb)
    verbose=yes ;;

    -version | --version | --versio | --versi | --vers | -V)
    ac_init_version=: ;;

    -with-* | --with-*)
    ac_useropt=`expr "x$ac_option" : 'x-*with-\([^=]*\)'`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
    as_fn_error $? "invalid package name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
    *)
"with_$ac_useropt"
"*) ;;
    *)
ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--with-
$ac_useropt_orig"
    ac_unrecognized_sep=', ';;
    esac
    eval with_$ac_useropt=\$ac_optarg ;;

    -without-* | --without-*)
    ac_useropt=`expr "x$ac_option" : 'x-*without-\(.*\)'`
    # Reject names that are not valid shell variable names.
    expr "x$ac_useropt" : ".*[^-+._$as_cr_alnum]" >/dev/null &&
    as_fn_error $? "invalid package name: $ac_useropt"
    ac_useropt_orig=$ac_useropt
    ac_useropt=`$as_echo "$ac_useropt" | sed 's/[-+.]/_/g'`
    case $ac_user_opts in
    *)
"with_$ac_useropt"
"*) ;;
    *)
ac_unrecognized_opts="$ac_unrecognized_opts$ac_unrecognized_sep--
without-$ac_useropt_orig"
    ac_unrecognized_sep=', ';;
    esac

```



```

eval with_${_sac}_useropt=no ;;

--x)
# Obsolete; use --with-x.
with_x=yes ;;

-x-includes | --x-includes | --x-include | --x-includ | --x-inclu \
| --x-incl | --x-inc | --x-in | --x-i)
ac_prev=x_includes ;;
-x-includes=* | --x-includes=* | --x-include=* | --x-includ=* | --x-
inclu=* \
| --x-incl=* | --x-inc=* | --x-in=* | --x-i=*)
x_includes=${_sac}_optarg ;;

-x-libraries | --x-libraries | --x-librarie | --x-librari \
| --x-librar | --x-libra | --x-libr | --x-lib | --x-li | --x-l)
ac_prev=x_libraries ;;
-x-libraries=* | --x-libraries=* | --x-librarie=* | --x-librari=* \
| --x-librar=* | --x-libra=* | --x-libr=* | --x-lib=* | --x-li=* | -
-x-l=*)
x_libraries=${_sac}_optarg ;;

-*) as_fn_error $? "unrecognized option: \`${_sac}_option'
Try \`${_sac} --help' for more information"
;;

*=*)
ac_envvar=`expr "x${_sac}_option" : 'x\([^=]*\)='`
# Reject names that are not valid shell variable names.
case $_sac_envvar in #(
'|' | [0-9]* | *[^$_sac_cr_alnum]* )
as_fn_error $? "invalid variable name: \`${_sac}_envvar'" ;;
esac
eval $_sac_envvar=${_sac}_optarg
export $_sac_envvar ;;

*)
# FIXME: should be removed in autoconf 3.0.
_sas_echo "$_sas_me: WARNING: you should use --build, --host, --
target" >&2
expr "x${_sac}_option" : ".*[^-$_sac_cr_alnum]" >/dev/null &&
_sas_echo "$_sas_me: WARNING: invalid host type: $_sac_option" >&2
: "${build_alias=$_sac_option} ${host_alias=$_sac_option}
${target_alias=$_sac_option}"
;;

esac
done

if test -n "$_sac_prev"; then
ac_option=--`echo $_sac_prev | sed 's/_/_/g'`
as_fn_error $? "missing argument to $_sac_option"

```

```

fi

if test -n "$ac_unrecognized_opts"; then
  case $enable_option_checking in
    no) ;;
    fatal) as_fn_error $? "unrecognized options:
$ac_unrecognized_opts" ;;
    *) $as_echo "$as_me: WARNING: unrecognized options:
$ac_unrecognized_opts" >&2 ;;
  esac
fi

# Check all directory arguments for consistency.
for ac_var in  exec_prefix prefix bindir sbindir libexecdir
datarootdir \
              datadir sysconfdir sharedstatedir localstatedir includedir
\
              oldincludedir docdir infodir htmdir dvidir pdfdir psdir \
libdir localedir mandir
do
  eval ac_val=\${ac_var}
  # Remove trailing slashes.
  case $ac_val in
    */ )
      ac_val=`expr "X$ac_val" : 'X\([^/]\)' \| "X$ac_val" :
'X\(.*\)'`
      eval $ac_var=\${ac_val};;
  esac
  # Be sure to have absolute directory names.
  case $ac_val in
    [\\/$]* | ?:[\\/$]* ) continue;;
    NONE | ' ' ) case $ac_var in *prefix ) continue;; esac;;
  esac
  as_fn_error $? "expected an absolute directory name for --$ac_var:
$ac_val"
done

# There might be people who depend on the old broken behavior: ` $host '
# used to hold the argument of --host etc.
# FIXME: To remove some day.
build=$build_alias
host=$host_alias
target=$target_alias

# FIXME: To remove some day.
if test "x$host_alias" != x; then
  if test "x$build_alias" = x; then
    cross_compiling=maybe
  elif test "x$build_alias" != "x$host_alias"; then
    cross_compiling=yes
  fi
fi

```

```

ac_tool_prefix=
test -n "$host_alias" && ac_tool_prefix=$host_alias-

test "$silent" = yes && exec 6>/dev/null

ac_pwd=`pwd` && test -n "$ac_pwd" &&
ac_ls_di=`ls -di .` &&
ac_pwd_ls_di=`cd "$ac_pwd" && ls -di .` ||
  as_fn_error $? "working directory cannot be determined"
test "X$ac_ls_di" = "X$ac_pwd_ls_di" ||
  as_fn_error $? "pwd does not report name of working directory"

# Find the source files, if location was not specified.
if test -z "$srcdir"; then
  ac_srcdir_defaulted=yes
  # Try the directory containing this script, then the parent
  directory.
  ac_confdir=`$as_dirname -- "$as_myself" ||
$as_expr X"$as_myself" : 'X\(.*[^/]\)\/*[^/][^/]*/*$' \| \
  X"$as_myself" : 'X\(//\)[^/]' \| \
  X"$as_myself" : 'X\(//\)$' \| \
  X"$as_myself" : 'X\(/\)' \| . 2>/dev/null ||
$as_echo X"$as_myself" |
  sed '/^X\(.*[^/]\)\/*[^/][^/]*\/*$/{
    s//\1/
    q
  }
/^X\(\\/\)\)[^/].*/{
  s//\1/
  q
}
/^X\(\\/\)\)$/{
  s//\1/
  q
}
/^X\(\\/\).*/{
  s//\1/
  q
}
s/.*\/./; q'`
  srcdir=$ac_confdir
  if test ! -r "$srcdir/$ac_unique_file"; then
    srcdir=..
  fi
else
  ac_srcdir_defaulted=no
fi
if test ! -r "$srcdir/$ac_unique_file"; then
  test "$ac_srcdir_defaulted" = yes && srcdir="$ac_confdir or .."

```

```

    as_fn_error $? "cannot find sources ($ac_unique_file) in $srcdir"
fi
ac_msg="sources are in $srcdir, but `cd $srcdir' does not work"
ac_abs_confdir=`(
    cd "$srcdir" && test -r "$ac_unique_file" || as_fn_error $?
"$ac_msg"
    pwd)`
# When building in place, set srcdir=.
if test "$ac_abs_confdir" = "$ac_pwd"; then
    srcdir=.
fi
# Remove unnecessary trailing slashes from srcdir.
# Double slashes in file names in object file debugging info
# mess up M-x gdb in Emacs.
case $srcdir in
*/) srcdir=`expr "X$srcdir" : 'X\([^/]\)' \| "X$srcdir" :
'X\(.*\)'`;
esac
for ac_var in $ac_precious_vars; do
    eval ac_env_${ac_var}_set=\${${ac_var}_set}
    eval ac_env_${ac_var}_value=\${${ac_var}_value}
    eval ac_cv_env_${ac_var}_set=\${${ac_var}_set}
    eval ac_cv_env_${ac_var}_value=\${${ac_var}_value}
done

#
# Report the --help message.
#
if test "$ac_init_help" = "long"; then
    # Omit some internal or obsolete options to make the list less
    imposing.
    # This message is too long to be a string in the A/UX 3.1 sh.
    cat <<_ACEOF
\`configure' configures dbus 1.6.8 to adapt to many kinds of systems.

Usage: $0 [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful
variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help                display this help and exit
    --help=short            display options specific to this package
    --help=recursive        display the short help of all the included
packages
  -V, --version             display version information and exit
  -q, --quiet, --silent    do not print `checking ...' messages
    --cache-file=FILE      cache test results in FILE [disabled]
  -C, --config-cache        alias for `--cache-file=config.cache'

```

```
-n, --no-create          do not create output files
--srcdir=DIR             find the sources in DIR [configure dir or
\`..'']
```

Installation directories:

```
--prefix=PREFIX         install architecture-independent files in
PREFIX
                        [$ac_default_prefix]
--exec-prefix=EPREFIX   install architecture-dependent files in
EPREFIX
                        [PREFIX]
```

By default, `\`make install'` will install all the files in `\`$ac_default_prefix/bin'`, `\`$ac_default_prefix/lib'` etc. You can specify an installation prefix other than `\`$ac_default_prefix'` using `\`--prefix'`, for instance `\`--prefix=$HOME'`.

For better control, use the options below.

Fine tuning of the installation directories:

```
--bindir=DIR            user executables [EPREFIX/bin]
--sbindir=DIR           system admin executables [EPREFIX/sbin]
--libexecdir=DIR        program executables [EPREFIX/libexec]
--sysconfdir=DIR        read-only single-machine data [PREFIX/etc]
--sharedstatedir=DIR    modifiable architecture-independent data
[PREFIX/com]
--localstatedir=DIR     modifiable single-machine data [PREFIX/var]
--libdir=DIR            object code libraries [EPREFIX/lib]
--includedir=DIR        C header files [PREFIX/include]
--oldincludedir=DIR     C header files for non-gcc [/usr/include]
--datarootdir=DIR       read-only arch.-independent data root
[PREFIX/share]
--datadir=DIR           read-only architecture-independent data
[DATAROOTDIR]
--infodir=DIR           info documentation [DATAROOTDIR/info]
--localedir=DIR         locale-dependent data [DATAROOTDIR/locale]
--mandir=DIR            man documentation [DATAROOTDIR/man]
--docdir=DIR            documentation root [DATAROOTDIR/doc/dbus]
--htmldir=DIR           html documentation [DOCDIR]
--dvidir=DIR            dvi documentation [DOCDIR]
--pdfdir=DIR            pdf documentation [DOCDIR]
--psdir=DIR             ps documentation [DOCDIR]
```

`_ACEOF`

```
cat <<\`_ACEOF
```

Program names:

```
--program-prefix=PREFIX      prepend PREFIX to installed
program names
```

--program-suffix=SUFFIX append SUFFIX to installed
program names
--program-transform-name=PROGRAM run sed PROGRAM on installed
program names

X features:

--x-includes=DIR X include files are in DIR
--x-libraries=DIR X library files are in DIR

System types:

--build=BUILD configure for building on BUILD [guessed]
--host=HOST cross-compile to build programs to run on HOST

[BUILD]

_ACEOF

fi

```
if test -n "$ac_init_help"; then
  case $ac_init_help in
    short | recursive ) echo "Configuration of dbus 1.6.8:";;
    esac
  cat <<\_ACEOF
```

Optional Features:

--disable-option-checking ignore unrecognized --enable/--with
options
--disable-FEATURE do not include FEATURE (same as --enable-
FEATURE=no)
--enable-FEATURE[=ARG] include FEATURE [ARG=yes]
--disable-maintainer-mode disable make rules and dependencies not
useful (and
 sometimes confusing) to the casual installer
--enable-silent-rules less verbose build output (undo: "make V=1")
--disable-silent-rules verbose build output (undo: "make V=0")
--enable-dependency-tracking do not reject slow dependency extractors
--disable-dependency-tracking speeds up one-time build
--enable-shared[=PKGS] build shared libraries [default=yes]
--enable-static[=PKGS] build static libraries [default=yes]
--enable-fast-install[=PKGS] optimize for fast installation [default=yes]
--disable-libtool-lock avoid locking (might break parallel builds)
--enable-compiler-coverage Enable generation of coverage data
--disable-compiler-optimisations Disable compiler optimisations
--enable-developer set defaults to be appropriate for a D-Bus
developer
 instead of a distribution/end-user
--enable-ansi enable -ansi -pedantic gcc flags
--enable-verbose-mode support verbose debug mode

```

--enable-asserts      include assertion checks
--enable-checks       include sanity checks on public API
--enable-xml-docs     build XML documentation (requires xmlto)
--enable-doxygen-docs build DOXYGEN documentation (requires
Doxygen)
--enable-abstract-sockets
                        use abstract socket namespace (linux only)
--enable-selinux      build with SELinux support
--enable-libaudit     build audit daemon support for SELinux
--enable-dnotify      build with dnotify support (linux only)
--enable-inotify      build with inotify support (linux only)
--enable-kqueue       build with kqueue support
--enable-console-owner-file
                        enable console owner file
--enable-userdb-cache build with userdb-cache support
--enable-launchd      build with launchd auto-launch support
--enable-systemd      build with systemd at_console support
--enable-embedded-tests enable unit test code in the library and
binaries
--enable-modular-tests enable modular regression tests (requires
GLib)
--enable-tests        enable/disable all tests, overriding
                        embedded-tests/modular-tests
--enable-installed-tests
                        enable unit test code in the library and
binaries
--enable-epoll        use epoll(4) on Linux
--enable-x11-autolaunch build with X11 auto-launch support
--disable-Werror      compile without -Werror (normally enabled in
                        development builds)
--enable-stats        enable bus daemon usage statistics

Optional Packages:
--with-PACKAGE[=ARG]  use PACKAGE [ARG=yes]
--without-PACKAGE     do not use PACKAGE (same as --with-
PACKAGE=no)
--with-pic[=PKGS]    try to use only PIC/non-PIC objects
[default=use
both]
--with-gnu-ld         assume the C compiler uses GNU ld
[default=no]
--with-libtool-sysroot=DIR Search for dependent libraries within DIR
                        (or the compiler's sysroot if not specified).
--with-xml=libxml/expat XML library to use (libxml may be named
libxml2 on
                        some systems)
--with-init-scripts=redhat
                        Style of init scripts to install
--with-session-socket-dir=dirname
                        Where to put sockets for the per-login-
session
                        message bus

```

```

--with-test-socket-dir=dirname
                        Where to put sockets for make check
--with-system-pid-file=pidfile
                        PID file for systemwide daemon
--with-system-socket=filename
                        UNIX domain socket for systemwide daemon
--with-console-auth-dir=dirname
                        directory to check for console ownership
--with-console-owner-file=filename
                        file whose owner determines current console
owner
--with-launchd-agent-dir=dirname
                        directory to put the launchd agent (default:
                        /Library/LaunchAgents)
--with-dbus-user=<user> User for running the DBUS daemon
(messagebus)
--with-dbus-daemon-dir=dirname
                        Directory for installing the DBUS daemon
--with-dbus-session-bus-default-address=nonce-
tcp:/autolaunch:/tcp:host:port
                        Transport Type to be used (default: nonce-
tcp:)
--without-64-bit       If you have to use this option, please
report it as
                        a bug
--with-valgrind        Add instrumentation to help valgrind to
understand
                        our allocator
--with-x               use the X Window System
--with-systemdsystemunitdir=DIR
                        Directory for systemd service files
--with-dbus-test-dir=dirname
                        path where the tests tools are available

```

Some influential environment variables:

```

CC          C compiler command
CFLAGS      C compiler flags
LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in
a
            nonstandard directory <lib dir>
LIBS        libraries to pass to the linker, e.g. -l<library>
CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include
dir> if
            you have headers in a nonstandard directory <include
dir>
CXX         C++ compiler command
CXXFLAGS    C++ compiler flags
CPP         C preprocessor
CXXCPP      C++ preprocessor
PKG_CONFIG  path to pkg-config utility
GLIB_CFLAGS C compiler flags for GLIB, overriding pkg-config
GLIB_LIBS   linker flags for GLIB, overriding pkg-config

```



```

DBUS_GLIB_CFLAGS
    C compiler flags for DBUS_GLIB, overriding pkg-config
DBUS_GLIB_LIBS
    linker flags for DBUS_GLIB, overriding pkg-config
PYTHON
    the Python interpreter
LIBXML_CFLAGS
    C compiler flags for LIBXML, overriding pkg-config
LIBXML_LIBS
    linker flags for LIBXML, overriding pkg-config
SYSTEMD_CFLAGS
    C compiler flags for SYSTEMD, overriding pkg-config
SYSTEMD_LIBS
    linker flags for SYSTEMD, overriding pkg-config
VALGRIND_CFLAGS
    C compiler flags for VALGRIND, overriding pkg-config
VALGRIND_LIBS
    linker flags for VALGRIND, overriding pkg-config
MAN2HTML
    Path to man2html (optional)

```

Use these variables to override the choices made by `configure' or to help it to find libraries and programs with nonstandard names/locations.

Report bugs to

<https://bugs.freedesktop.org/enter_bug.cgi?product=dbus>.

_ACEOF

```

ac_status=$?
fi

```

```

if test "$ac_init_help" = "recursive"; then
  # If there are subdirs, report their specific --help.
  for ac_dir in : $ac_subdirs_all; do test "x$ac_dir" = x: && continue
    test -d "$ac_dir" ||
      { cd "$srcdir" && ac_pwd=`pwd` && srcdir=. && test -d "$ac_dir";
    } ||
      continue
    ac_builddir=.

```

```

case "$ac_dir" in
  .) ac_dir_suffix= ac_top_builddir_sub=. ac_top_build_prefix= ;;
  *)

```

```

    ac_dir_suffix=`$as_echo "$ac_dir" | sed 's|^\.([\//]||)'`
    # A ".." for each directory in $ac_dir_suffix.
    ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/[^\//]*|/..|g;s|/||'`
    case $ac_top_builddir_sub in
      "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
      *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
    esac ;;

```

esac

```

ac_abs_top_builddir=$ac_pwd
ac_abs_builddir=$ac_pwd$ac_dir_suffix
# for backward compatibility:

```

```

ac_top_builddir=$ac_top_build_prefix

case $srcdir in
  .) # We are building in place.
    ac_srcdir=.
    ac_top_srcdir=$ac_top_builddir_sub
    ac_abs_top_srcdir=$ac_pwd ;;
  [\\/* | ?:[\\/*]* ) # Absolute name.
    ac_srcdir=$srcdir$ac_dir_suffix;
    ac_top_srcdir=$srcdir
    ac_abs_top_srcdir=$srcdir ;;
  *) # Relative name.
    ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
    ac_top_srcdir=$ac_top_build_prefix$srcdir
    ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

cd "$ac_dir" || { ac_status=$?; continue; }
# Check for gusted configure.
if test -f "$ac_srcdir/configure.gnu"; then
  echo &&
  $SHELL "$ac_srcdir/configure.gnu" --help=recursive
elif test -f "$ac_srcdir/configure"; then
  echo &&
  $SHELL "$ac_srcdir/configure" --help=recursive
else
  $as_echo "$as_me: WARNING: no configuration information is in
$ac_dir" >&2
  fi || ac_status=$?
  cd "$ac_pwd" || { ac_status=$?; break; }
done
fi

test -n "$ac_init_help" && exit $ac_status
if $ac_init_version; then
  cat <<\_ACEOF
dbus configure 1.6.8
generated by GNU Autoconf 2.69

Copyright (C) 2012 Free Software Foundation, Inc.
This configure script is free software; the Free Software Foundation
gives unlimited permission to copy, distribute and modify it.
\_ACEOF
  exit
fi

## ----- ##
## Autoconf initialization. ##
## ----- ##

# ac_fn_c_try_compile LINENO

```

```

# -----
# Try to compile confptest.$ac_ext, and return whether this succeeded.
ac_fn_c_try_compile ()
{
  as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  rm -f confptest.$ac_objext
  if { { ac_try="$ac_compile"
case "($ac_try" in
  *\"* | *\\* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_compile") 2>confptest.err
  ac_status=$?
  if test -s confptest.err; then
    grep -v '^ *+' confptest.err >confptest.er1
    cat confptest.er1 >&5
    mv -f confptest.er1 confptest.err
  fi
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; } && {
    test -z "$ac_c_werror_flag" ||
    test ! -s confptest.err
  } && test -s confptest.$ac_objext; then :
  ac_retval=0
else
  $as_echo "$as_me: failed program was:" >&5
  sed 's/^/| /' confptest.$ac_ext >&5

  ac_retval=1
fi
  eval $as_lineno_stack; ${as_lineno_stack:+} unset as_lineno
  as_fn_set_status $ac_retval
} # ac_fn_c_try_compile

# ac_fn_cxx_try_compile LINENO
# -----
# Try to compile confptest.$ac_ext, and return whether this succeeded.
ac_fn_cxx_try_compile ()
{
  as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  rm -f confptest.$ac_objext
  if { { ac_try="$ac_compile"
case "($ac_try" in
  *\"* | *\\* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""

```

```

$as_echo "$ac_try_echo"; } >&5
(eval "$ac_compile") 2>confptest.err
ac_status=$?
if test -s confptest.err; then
  grep -v '^ *+' confptest.err >confptest.er1
  cat confptest.er1 >&5
  mv -f confptest.er1 confptest.err
fi
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; } && {
  test -z "$ac_cxx_werror_flag" ||
  test ! -s confptest.err
} && test -s confptest.$ac_objext; then :
ac_retval=0
else
  $as_echo "$as_me: failed program was:" >&5
  sed 's/^/| /' confptest.$ac_ext >&5

  ac_retval=1
fi
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_cxx_try_compile

# ac_fn_c_try_cpp LINENO
# -----
# Try to preprocess confptest.$ac_ext, and return whether this
# succeeded.
ac_fn_c_try_cpp ()
{
  as_lineno=${as_lineno-"$1"}
  as_lineno_stack=as_lineno_stack=$as_lineno_stack
  if { { ac_try="$ac_cpp confptest.$ac_ext"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"\`$as_me:${as_lineno-$LINENO}: $ac_try_echo\`\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_cpp confptest.$ac_ext") 2>confptest.err
  ac_status=$?
  if test -s confptest.err; then
    grep -v '^ *+' confptest.err >confptest.er1
    cat confptest.er1 >&5
    mv -f confptest.er1 confptest.err
  fi
  $as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
  test $ac_status = 0; } > confptest.i && {
    test -z "$ac_c_preproc_warn_flag$ac_c_werror_flag" ||
    test ! -s confptest.err
  }; then :

```

```

    ac_retval=0
else
    $as_echo "$as_me: failed program was:" >&5
sed 's/^\| /' conftest.$ac_ext >&5

    ac_retval=1
fi
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_c_try_cpp

# ac_fn_c_check_header_mongrel LINENO HEADER VAR INCLUDES
# -----
# Tests whether HEADER exists, giving a warning if it cannot be
# compiled using
# the include files in INCLUDES and setting the cache variable VAR
# accordingly.
ac_fn_c_check_header_mongrel ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if eval \${$3+:} false; then :
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
    if eval \${$3+:} false; then :
        $as_echo_n "(cached) " >&6
    fi
    eval ac_res=\${$3}
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
    else
        # Is the header compilable?
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking $2 usability" >&5
$as_echo_n "checking $2 usability... " >&6; }
        cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
        /* end confdefs.h. */
        $4
        #include <$2>
        _ACEOF
        if ac_fn_c_try_compile "$LINENO"; then :
            ac_header_compiler=yes
        else
            ac_header_compiler=no
        fi
        rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_header_compiler"
>&5
$as_echo "$ac_header_compiler" >&6; }

# Is the header present?

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking $2 presence" >&5
$as_echo_n "checking $2 presence... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <$2>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
  ac_header_preproc=yes
else
  ac_header_preproc=no
fi
rm -f conftest.err conftest.i conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_header_preproc"
>&5
$as_echo "$ac_header_preproc" >&6; }

# So?  What about this header?
case $ac_header_compiler:$ac_header_preproc:$ac_c_preproc_warn_flag in
#((
  yes:no: )
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: accepted by
the compiler, rejected by the preprocessor!" >&5
$as_echo "$as_me: WARNING: $2: accepted by the compiler, rejected by
the preprocessor!" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: proceeding
with the compiler's result" >&5
$as_echo "$as_me: WARNING: $2: proceeding with the compiler's result"
>&2;}
    ;;
  no:yes:* )
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: present but
cannot be compiled" >&5
$as_echo "$as_me: WARNING: $2: present but cannot be compiled" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2:      check
for missing prerequisite headers?" >&5
$as_echo "$as_me: WARNING: $2:      check for missing prerequisite
headers?" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: see the
Autoconf documentation" >&5
$as_echo "$as_me: WARNING: $2: see the Autoconf documentation" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2:      section
\"Present But Cannot Be Compiled\"" >&5
$as_echo "$as_me: WARNING: $2:      section \"Present But Cannot Be
Compiled\"" >&2;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $2: proceeding
with the compiler's result" >&5
$as_echo "$as_me: WARNING: $2: proceeding with the compiler's result"
>&2;}
  ( $as_echo "## -----
----- ##
## Report this to
https://bugs.freedesktop.org/enter\_bug.cgi?product=dbus ##

```

```

## -----
--- ##"
    ) | sed "s/^/$as_me: WARNING:      /" >&2
    ;;
esac
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
if eval \${$3+:} false; then :
  $as_echo_n "(cached) " >&6
else
  eval "$3=\$ac_header_compiler"
fi
eval ac_res=\${$3}
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
fi
  eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
} # ac_fn_c_check_header_mongrel

# ac_fn_c_try_run LINENO
# -----
# Try to link conftest.$ac_ext, and return whether this succeeded.
# Assumes
# that executables *can* be run.
ac_fn_c_try_run ()
{
  as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  if { { ac_try="$ac_link"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; } && { ac_try='./conftest$ac_exeext'
  { { case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_try") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; }; }; then :
  ac_retval=0

```

```

else
  $as_echo "$as_me: program exited with status $ac_status" >&5
  $as_echo "$as_me: failed program was:" >&5
sed 's/^\|\/' conftest.$ac_ext >&5

    ac_retval=$ac_status
fi
rm -rf conftest.dSYM conftest_ipa8_conftest.o
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_c_try_run

# ac_fn_c_check_header_compile LINENO HEADER VAR INCLUDES
# -----
# Tests whether HEADER exists and can be compiled using the include
files in
# INCLUDES, setting the cache variable VAR accordingly.
ac_fn_c_check_header_compile ()
{
  as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
if eval \${$3+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
$4
#include <$2>
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  eval "$3=yes"
else
  eval "$3=no"
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
eval ac_res=\${$3}
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
  eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno

} # ac_fn_c_check_header_compile

# ac_fn_c_try_link LINENO
# -----
# Try to link conftest.$ac_ext, and return whether this succeeded.
ac_fn_c_try_link ()
{

```



```

    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    rm -f conftest.$ac_objext conftest$ac_exeext
    if { { ac_try="$ac_link"
case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_link") 2>conftest.err
    ac_status=$?
    if test -s conftest.err; then
        grep -v '^ *+' conftest.err >conftest.er1
        cat conftest.er1 >&5
        mv -f conftest.er1 conftest.err
    fi
    $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
    test $ac_status = 0; } && {
        test -z "$ac_c_werror_flag" ||
        test ! -s conftest.err
        } && test -s conftest$ac_exeext && {
        test "$cross_compiling" = yes ||
        test -x conftest$ac_exeext
        }; then :
    ac_retval=0
else
    $as_echo "$as_me: failed program was:" >&5
    sed 's/^/| /' conftest.$ac_ext >&5

        ac_retval=1
fi
    # Delete the IPA/IPO (Inter Procedural Analysis/Optimization)
information
    # created by the PGI compiler (conftest_ipa8_conftest.oo), as it
would
    # interfere with the next link command; also delete a directory that
is
    # left behind by Apple's compiler.  We do this before executing the
actions.
    rm -rf conftest.dSYM conftest_ipa8_conftest.oo
    eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
    as_fn_set_status $ac_retval

} # ac_fn_c_try_link

# ac_fn_c_check_func LINENO FUNC VAR
# -----
# Tests whether FUNC exists, setting the cache variable VAR
accordingly
ac_fn_c_check_func ()
{

```

```

    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $2" >&5
$as_echo_n "checking for $2... " >&6; }
if eval `:${3+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
/* Define $2 to an innocuous variant, in case <limits.h> declares $2.
   For example, HP-UX 11i <limits.h> declares gettimeofday.  */
#define $2 innocuous_$2

/* System header to define __stub macros and hopefully few prototypes,
   which can conflict with char $2 (); below.
   Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
   <limits.h> exists even on freestanding compilers.  */

#ifdef __STDC__
# include <limits.h>
#else
# include <assert.h>
#endif

#undef $2

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char $2 ();
/* The GNU C library defines this for functions which it implements
   to always fail with ENOSYS.  Some functions are actually named
   something starting with __ and the normal name is an alias.  */
#ifdef defined __stub_$2 || defined __stub___$2
choke me
#endif

int
main ()
{
return $2 ();
  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  eval "$3=yes"
else
  eval "$3=no"

```

```

fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
fi
eval ac_res=\${3}
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
    eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno

} # ac_fn_c_check_func

# ac_fn_cxx_try_cpp LINENO
# -----
# Try to preprocess conftest.$ac_ext, and return whether this
succeeded.
ac_fn_cxx_try_cpp ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if { { ac_try="conftest.$ac_ext"
case "($ac_try" in
    *\"* | *\\`* | *\\*) ac_try_echo=\$ac_try;;
    *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"\`$as_me:${as_lineno-$LINENO}: $ac_try_echo\`"
$as_echo "$ac_try_echo"; } >&5
    (eval "conftest.$ac_ext") 2>conftest.err
ac_status=$?
if test -s conftest.err; then
    grep -v '^ *+' conftest.err >conftest.er1
    cat conftest.er1 >&5
    mv -f conftest.er1 conftest.err
fi
$as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
test $ac_status = 0; } > conftest.i && {
    test -z "$ac_cxx_preproc_warn_flag$ac_cxx_werror_flag" ||
    test ! -s conftest.err
    }; then :
    ac_retval=0
else
    $as_echo "$as_me: failed program was:" >&5
sed 's/^/| /' conftest.$ac_ext >&5

    ac_retval=1
fi
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval

} # ac_fn_cxx_try_cpp

# ac_fn_cxx_try_link LINENO

```

```

# -----
# Try to link confptest.$ac_ext, and return whether this succeeded.
ac_fn_cxx_try_link ()
{
  as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
  rm -f confptest.$ac_objext confptest$ac_exeext
  if { { ac_try="$ac_link"
case "($ac_try" in
  *\`* | *\`* | *\`*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link") 2>confptest.err
  ac_status=$?
  if test -s confptest.err; then
    grep -v '^ *+' confptest.err >confptest.er1
    cat confptest.er1 >&5
    mv -f confptest.er1 confptest.err
  fi
  $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
  test $ac_status = 0; } && {
    test -z "$ac_cxx_werror_flag" ||
    test ! -s confptest.err
    } && test -s confptest$ac_exeext && {
    test "$cross_compiling" = yes ||
    test -x confptest$ac_exeext
    }; then :
  ac_retval=0
else
  $as_echo "$as_me: failed program was:" >&5
  sed 's/^/| /' confptest.$ac_ext >&5

    ac_retval=1
fi
  # Delete the IPA/IPO (Inter Procedural Analysis/Optimization)
information
  # created by the PGI compiler (confptest_ipa8_confptest.o), as it
would
  # interfere with the next link command; also delete a directory that
is
  # left behind by Apple's compiler. We do this before executing the
actions.
  rm -rf confptest.dSYM confptest_ipa8_confptest.o
  eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
  as_fn_set_status $ac_retval

} # ac_fn_cxx_try_link

# ac_fn_c_compute_int LINENO EXPR VAR INCLUDES
# -----

```

```

# Tries to find the compile-time value of EXPR in a program that
includes
# INCLUDES, setting VAR accordingly. Returns whether the value could
be
# computed
ac_fn_c_compute_int ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    if test "$cross_compiling" = yes; then
        # Depending upon the size, compute the lo and hi bounds.
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
int
main ()
{
static int test_array [1 - 2 * !((($2) >= 0)];
test_array [0] = 0;
return test_array [0];

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_lo=0 ac_mid=0
    while ;; do
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
int
main ()
{
static int test_array [1 - 2 * !((($2) <= $ac_mid)];
test_array [0] = 0;
return test_array [0];

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_hi=$ac_mid; break
else
    as_fn_arith $ac_mid + 1 && ac_lo=$as_val
        if test $ac_lo -le $ac_mid; then
            ac_lo= ac_hi=
            break
        fi
    as_fn_arith 2 '*' $ac_mid + 1 && ac_mid=$as_val
fi
fi

```

```

rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
done
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
int
main ()
{
static int test_array [1 - 2 * !(($2) < 0)];
test_array [0] = 0;
return test_array [0];

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_hi=-1 ac_mid=-1
  while ;; do
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
int
main ()
{
static int test_array [1 - 2 * !(($2) >= $ac_mid)];
test_array [0] = 0;
return test_array [0];

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_lo=$ac_mid; break
else
  as_fn_arith '(' $ac_mid ')' - 1 && ac_hi=$as_val
    if test $ac_mid -le $ac_hi; then
      ac_lo= ac_hi=
      break
    fi
  as_fn_arith 2 '*' $ac_mid && ac_mid=$as_val
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
done
else
  ac_lo= ac_hi=
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

```

```

# Binary search between lo and hi bounds.
while test "x$ac_lo" != "x$ac_hi"; do
  as_fn_arith '(' $ac_hi - $ac_lo ')' / 2 + $ac_lo && ac_mid=$as_val
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
int
main ()
{
static int test_array [1 - 2 * !((($2) <= $ac_mid)];
test_array [0] = 0;
return test_array [0];

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_hi=$ac_mid
else
  as_fn_arith '(' $ac_mid ')' + 1 && ac_lo=$as_val
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
done
case $ac_lo in #((
?*) eval "$3=\$ac_lo"; ac_retval=0 ;;
'') ac_retval=1 ;;
esac
else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$4
static long int longval () { return $2; }
static unsigned long int ulongval () { return $2; }
#include <stdio.h>
#include <stdlib.h>
int
main ()
{

FILE *f = fopen ("conftest.val", "w");
if (! f)
  return 1;
if (($2) < 0)
  {
    long int i = longval ();
    if (i != ($2))
      return 1;
    fprintf (f, "%ld", i);
  }
else
  {

```

```

        unsigned long int i = ulongval ();
        if (i != ($2))
        return 1;
        fprintf (f, "%lu", i);
    }
/* Do not output a trailing newline, as this causes \r\n confusion
   on some platforms.  */
return ferror (f) || fclose (f) != 0;

;
return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    echo >>confptest.val; read $3 <confptest.val; ac_retval=0
else
    ac_retval=1
fi
rm -f core *.core core.confptest.* gmon.out bb.out confptest$sac_exeext \
    confptest.$sac_objext confptest.beam confptest.$sac_ext
rm -f confptest.val

fi
eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno
as_fn_set_status $ac_retval
} # ac_fn_c_compute_int

# ac_fn_c_check_decl LINENO SYMBOL VAR INCLUDES
# -----
# Tests whether SYMBOL is declared in INCLUDES, setting cache variable
# VAR
# accordingly.
ac_fn_c_check_decl ()
{
    as_lineno=${as_lineno-"$1"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    as_decl_name=`echo $2|sed 's/ *(.*/'`
    as_decl_use=`echo $2|sed -e 's/(/(/' -e 's/)/)' 0&/' -e 's/,/,' 0&
(/g'`
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether
$as_decl_name is declared" >&5
$as_echo_n "checking whether $as_decl_name is declared... " >&6; }
if eval `echo ${3+} false`; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >confptest.$sac_ext
/* end confdefs.h.  */
$4
int
main ()
{

```



```

#endif $as_decl_name
#ifdef __cplusplus
    (void) $as_decl_use;
#else
    (void) $as_decl_name;
#endif
#endif

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    eval "$3=yes"
else
    eval "$3=no"
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
eval ac_res=\$3
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_res"
>&5
$as_echo "$ac_res" >&6; }
    eval $as_lineno_stack; ${as_lineno_stack:+:} unset as_lineno

} # ac_fn_c_check_decl
cat >config.log <<_ACEOF
This file contains any messages produced by compilers while
running configure, to aid debugging if configure makes a mistake.

It was created by dbus $as_me 1.6.8, which was
generated by GNU Autoconf 2.69.  Invocation command line was

    $ $0 $@

_ACEOF
exec 5>>config.log
{
cat <<_ASUNAME
## ----- ##
## Platform. ##
## ----- ##

hostname = `(hostname || uname -n) 2>/dev/null | sed 1q`
uname -m = `(uname -m) 2>/dev/null || echo unknown`
uname -r = `(uname -r) 2>/dev/null || echo unknown`
uname -s = `(uname -s) 2>/dev/null || echo unknown`
uname -v = `(uname -v) 2>/dev/null || echo unknown`

/usr/bin/uname -p = `(/usr/bin/uname -p) 2>/dev/null || echo unknown`
/bin/uname -X      = `(/bin/uname -X) 2>/dev/null      || echo unknown`

```

```

/bin/arch                = `(/bin/arch) 2>/dev/null          || echo
unknown`
/usr/bin/arch -k         = `(/usr/bin/arch -k) 2>/dev/null      || echo
unknown`
/usr/convex/getsysinfo  = `(/usr/convex/getsysinfo) 2>/dev/null  || echo
unknown`
/usr/bin/hostinfo       = `(/usr/bin/hostinfo) 2>/dev/null      || echo
unknown`
/bin/machine            = `(/bin/machine) 2>/dev/null          || echo
unknown`
/usr/bin/oslevel        = `(/usr/bin/oslevel) 2>/dev/null      || echo
unknown`
/bin/universe           = `(/bin/universe) 2>/dev/null        || echo
unknown`

```

```

_ASUNAME

```

```

as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  $as_echo "PATH: $as_dir"
done
IFS=$as_save_IFS

} >&5

```

```

cat >&5 <<_ACEOF

```

```

## ----- ##
## Core tests. ##
## ----- ##

```

```

_ACEOF

```

```

# Keep a trace of the command line.
# Strip out --no-create and --no-recursion so they do not pile up.
# Strip out --silent because we don't want to record it for future
runs.
# Also quote any args containing shell meta-characters.
# Make two passes to allow for proper duplicate-argument suppression.
ac_configure_args=
ac_configure_args0=
ac_configure_args1=
ac_must_keep_next=false
for ac_pass in 1 2
do
  for ac_arg
  do

```

```

case $ac_arg in
-no-create | --no-c* | -n | -no-recursion | --no-r*) continue ;;
-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil)
    continue ;;
*\'*)
    ac_arg=`$as_echo "$ac_arg" | sed "s/'/'\\\'\\\'\\\'\\\'/g"` ;;
esac
case $ac_pass in
1) as_fn_append ac_configure_args0 " '$ac_arg' " ;;
2)
    as_fn_append ac_configure_args1 " '$ac_arg' "
    if test $ac_must_keep_next = true; then
ac_must_keep_next=false # Got value, back to normal.
    else
        case $ac_arg in
        *=* | --config-cache | -C | -disable-* | --disable-* \
        | -enable-* | --enable-* | -gas | --g* | -nfp | --nf* \
        | -q | -quiet | --q* | -silent | --sil* | -v | -verb* \
        | -with-* | --with-* | -without-* | --without-* | --x)
            case "$ac_configure_args0 " in
                "$ac_configure_args1"* " '$ac_arg' "*" ) continue ;;
            esac
            ;;
        -* ) ac_must_keep_next=true ;;
        esac
    fi
    as_fn_append ac_configure_args " '$ac_arg' "
    ;;
esac
done
done
{ ac_configure_args0=; unset ac_configure_args0;}
{ ac_configure_args1=; unset ac_configure_args1;}

# When interrupted or exit'd, cleanup temporary files, and complete
# config.log. We remove comments because anyway the quotes in there
# would cause problems or look ugly.
# WARNING: Use '\'' to represent an apostrophe within the trap.
# WARNING: Do not start the trap code with a newline, due to a FreeBSD
4.0 bug.
trap 'exit_status=$?
    # Save into config.log some information that might help in
debugging.
    {
        echo

        $as_echo "## ----- ##"
## Cache variables. ##
## ----- ##"
        echo

```

```

# The following way of writing the cache mishandles newlines in
values,
(
  for ac_var in `(set) 2>&l | sed -n '\''s/^\([a-zA-Z_][a-zA-Z0-
9_]*\)ate.*\1/p'\''`; do
    eval ac_val=\$$ac_var
    case $ac_val in #(
      *${as_nl}*)
        case $ac_var in #(
          *_cv_*) { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: cache
variable $ac_var contains a newline" >&5
$as_echo "$as_me: WARNING: cache variable $ac_var contains a newline"
>&2;} ;;
          esac
          case $ac_var in #(
            _ | IFS | as_nl) ;; #(
            BASH_ARGV | BASH_SOURCE) eval $ac_var= ;; #(
            *) { eval $ac_var=; unset $ac_var;} ;;
          esac ;;
        esac
      done
      (set) 2>&l |
      case $as_nl`(_ac_space='\'' '\''; set) 2>&l` in #(
        *${as_nl}ac_space=\ *)
          sed -n \
            "s/'\''/'\''\\\`\\\`'\'''\'''\''/g;

s/^\`\\\`([\_$_as_cr_alnum]*_cv_[\_$_as_cr_alnum]*\\\`)=\\\`.\*\\\`)\1='\'''\`2'\'''\`
/p"
          ;; #(
        *)
          sed -n "/^[\_$_as_cr_alnum]*_cv_[\_$_as_cr_alnum]*=/p"
          ;;
        esac |
      sort
    )
    echo

    $as_echo "## ----- ##"
## Output variables. ##
## ----- ##"
    echo
    for ac_var in $ac_subst_vars
    do
      eval ac_val=\$$ac_var
      case $ac_val in
        *'\''*\''*) ac_val=`$as_echo "$ac_val" | sed
"s/'\''/'\''\\\`\\\`'\'''\'''\''/g"`;;
        esac
      $as_echo "$ac_var='\'''$ac_val'\'''"
    done | sort
    echo

```

```

        if test -n "$ac_subst_files"; then
            $as_echo "## ----- ##"
## File substitutions. ##
## ----- ##"
            echo
            for ac_var in $ac_subst_files
            do
                eval ac_val=\${$ac_var}
                case $ac_val in
                    *'\''*) ac_val=`$as_echo "$ac_val" | sed
"s/'\''/'\''\\\'\\\'\\\'\\\'\\\'\\\'\\\'\\\'\\\'\\\'/g"`;;
                esac
                $as_echo "$ac_var='\''$ac_val'\''"
                done | sort
                echo
            fi

            if test -s confdefs.h; then
                $as_echo "## ----- ##"
## confdefs.h. ##
## ----- ##"
                echo
                cat confdefs.h
                echo
            fi
            test "$ac_signal" != 0 &&
                $as_echo "$as_me: caught signal $ac_signal"
                $as_echo "$as_me: exit $exit_status"
        } >&5
        rm -f core *.core core.conftest.* &&
        rm -f -r conftest* confdefs* conf$$$* $ac_clean_files &&
        exit $exit_status
' 0
for ac_signal in 1 2 13 15; do
    trap 'ac_signal='$ac_signal'; as_fn_exit 1' $ac_signal
done
ac_signal=0

# confdefs.h avoids OS command line length limits that DEFS can
# exceed.
rm -f -r conftest* confdefs.h

$as_echo "/* confdefs.h */" > confdefs.h

# Predefined preprocessor variables.

cat >>confdefs.h <<_ACEOF
#define PACKAGE_NAME "$PACKAGE_NAME"
_ACEOF

cat >>confdefs.h <<_ACEOF

```

```

#define PACKAGE_TARNAME "$PACKAGE_TARNAME"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_VERSION "$PACKAGE_VERSION"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_STRING "$PACKAGE_STRING"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_BUGREPORT "$PACKAGE_BUGREPORT"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define PACKAGE_URL "$PACKAGE_URL"
_ACEOF

# Let the site file select an alternate cache file if it wants to.
# Prefer an explicitly selected file to automatically selected ones.
ac_site_file1=NONE
if test -n "$CONFIG_SITE"; then
  # We do not want a PATH search for config.site.
  case $CONFIG_SITE in #((
    -*)  ac_site_file1=./$CONFIG_SITE;;
    */*) ac_site_file1=$CONFIG_SITE;;
    *)   ac_site_file1=./$CONFIG_SITE;;
  esac
fi
for ac_site_file in $ac_site_file1
do
  test "x$ac_site_file" = xNONE && continue
  if test /dev/null != "$ac_site_file" && test -r "$ac_site_file";
then
  { $as_echo "$as_me:${as_lineno-$LINENO}: loading site script
$ac_site_file" >&5
$as_echo "$as_me: loading site script $ac_site_file" >&6;}
  sed 's/^/| /' "$ac_site_file" >&5
  . "$ac_site_file" \
  || { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in
\`$ac_pwd':" >&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error $? "failed to load site script $ac_site_file
See \`config.log' for more details" "$LINENO" 5; }
  fi
done

if test -r "$cache_file"; then
  # Some versions of bash will fail to source /dev/null (special files

```

```

# actually), so we avoid doing that. DJGPP emulates it as a regular
file.
if test /dev/null != "$cache_file" && test -f "$cache_file"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: loading cache
$cache_file" >&5
$as_echo "$as_me: loading cache $cache_file" >&6;}
  case $cache_file in
    [\\/]*) | ?:[\\/]*) . "$cache_file";;
    *) . "$cache_file";;
  esac
fi
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: creating cache $cache_file"
>&5
$as_echo "$as_me: creating cache $cache_file" >&6;}
  >$cache_file
fi

# Check that the precious variables saved in the cache have kept the
same
# value.
ac_cache_corrupted=false
for ac_var in $ac_precious_vars; do
  eval ac_old_set=\$ac_cv_env_${ac_var}_set
  eval ac_new_set=\$ac_env_${ac_var}_set
  eval ac_old_val=\$ac_cv_env_${ac_var}_value
  eval ac_new_val=\$ac_env_${ac_var}_value
  case $ac_old_set,$ac_new_set in
    set,)
      { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' was
set to \`${ac_old_val}' in the previous run" >&5
$as_echo "$as_me: error: \`${ac_var}' was set to \`${ac_old_val}' in the
previous run" >&2;}
      ac_cache_corrupted=: ;;
    ,set)
      { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' was
not set in the previous run" >&5
$as_echo "$as_me: error: \`${ac_var}' was not set in the previous run"
>&2;}
      ac_cache_corrupted=: ;;
    ,);)
      *)
        if test "x$ac_old_val" != "x$ac_new_val"; then
          # differences in whitespace do not lead to failure.
          ac_old_val_w=`echo x $ac_old_val`
          ac_new_val_w=`echo x $ac_new_val`
          if test "$ac_old_val_w" != "$ac_new_val_w"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: error: \`${ac_var}' has
changed since the previous run:" >&5
$as_echo "$as_me: error: \`${ac_var}' has changed since the previous
run:" >&2;}
            ac_cache_corrupted=:

```

```

else
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: warning: ignoring
whitespace changes in \`${ac_var}' since the previous run:" >&5
$zas_echo "$sas_me: warning: ignoring whitespace changes in \`${ac_var}'
since the previous run:" >&2;}
    eval $ac_var=\${ac_old_val}
fi
    { $zas_echo "$sas_me:${as_lineno-$LINENO}: former value:
\`${ac_old_val}'" >&5
$zas_echo "$sas_me: former value: \`${ac_old_val}'" >&2;}
    { $zas_echo "$sas_me:${as_lineno-$LINENO}: current value:
\`${ac_new_val}'" >&5
$zas_echo "$sas_me: current value: \`${ac_new_val}'" >&2;}
fi;;
esac
# Pass precious variables to config.status.
if test "$ac_new_set" = set; then
    case $ac_new_val in
    *'*) ac_arg=$ac_var=`$zas_echo "$ac_new_val" | sed
"s/'/'\\\\\\\\\\\\\\\\\\'/g"` ;;
    *) ac_arg=$ac_var=$ac_new_val ;;
    esac
    case " $ac_configure_args " in
    *' '$ac_arg' '*) ;; # Avoid dups. Use of quotes ensures
accuracy.
    *) as_fn_append ac_configure_args " '$ac_arg'" ;;
    esac
fi
done
if $ac_cache_corrupted; then
    { $zas_echo "$sas_me:${as_lineno-$LINENO}: error: in \`${ac_pwd}':" >&5
$zas_echo "$sas_me: error: in \`${ac_pwd}':" >&2;}
    { $zas_echo "$sas_me:${as_lineno-$LINENO}: error: changes in the
environment can compromise the build" >&5
$zas_echo "$sas_me: error: changes in the environment can compromise the
build" >&2;}
    as_fn_error $? "run `make distclean' and/or `rm $cache_file' and
start over" "$LINENO" 5
fi
## ----- ##
## Main body of script. ##
## ----- ##

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

```



```

ac_aux_dir=
for ac_dir in "$srcdir" "$srcdir/.." "$srcdir/../../.."; do
  if test -f "$ac_dir/install-sh"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/install-sh -c"
    break
  elif test -f "$ac_dir/install.sh"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/install.sh -c"
    break
  elif test -f "$ac_dir/shtool"; then
    ac_aux_dir=$ac_dir
    ac_install_sh="$ac_aux_dir/shtool install -c"
    break
  fi
done
if test -z "$ac_aux_dir"; then
  as_fn_error $? "cannot find install-sh, install.sh, or shtool in
\"$srcdir\" \"$srcdir/..\" \"$srcdir/../../..\"" "$LINENO" 5
fi

# These three variables are undocumented and unsupported,
# and are intended to be withdrawn in a future Autoconf release.
# They can cause serious problems if a builder's source tree is in a
directory
# whose full name contains unusual characters.
ac_config_guess="$SHELL $ac_aux_dir/config.guess" # Please don't use
this var.
ac_config_sub="$SHELL $ac_aux_dir/config.sub" # Please don't use this
var.
ac_configure="$SHELL $ac_aux_dir/configure" # Please don't use this
var.

# Make sure we can run config.sub.
$SHELL "$ac_aux_dir/config.sub" sun4 >/dev/null 2>&1 ||
  as_fn_error $? "cannot run $SHELL $ac_aux_dir/config.sub" "$LINENO"
5

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking build system type"
>&5
$as_echo_n "checking build system type... " >&6; }
if ${ac_cv_build+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_build_alias=$build_alias
test "x$ac_build_alias" = x &&
  ac_build_alias=`$SHELL "$ac_aux_dir/config.guess"`
test "x$ac_build_alias" = x &&
  as_fn_error $? "cannot guess build type; you must specify one"
"$LINENO" 5
ac_cv_build=`$SHELL "$ac_aux_dir/config.sub" $ac_build_alias` ||

```

```

    as_fn_error $? "$SHELL $ac_aux_dir/config.sub $ac_build_alias
failed" "$LINENO" 5

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_build" >&5
$as_echo "$ac_cv_build" >&6; }
case $ac_cv_build in
*-*-*) ;;
*) as_fn_error $? "invalid value of canonical build" "$LINENO" 5;;
esac
build=$ac_cv_build
ac_save_IFS=$IFS; IFS='- '
set x $ac_cv_build
shift
build_cpu=$1
build_vendor=$2
shift; shift
# Remember, the first character of IFS is used to create $*,
# except with old shells:
build_os=$*
IFS=$ac_save_IFS
case $build_os in *\ *) build_os=`echo "$build_os" | sed 's/ /-/g'`;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking host system type"
>&5
$as_echo_n "checking host system type... " >&6; }
if ${ac_cv_host+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "x$host_alias" = x; then
    ac_cv_host=$ac_cv_build
  else
    ac_cv_host=`$SHELL "$ac_aux_dir/config.sub" $host_alias` ||
      as_fn_error $? "$SHELL $ac_aux_dir/config.sub $host_alias failed"
"$LINENO" 5
  fi
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_host" >&5
$as_echo "$ac_cv_host" >&6; }
case $ac_cv_host in
*-*-*) ;;
*) as_fn_error $? "invalid value of canonical host" "$LINENO" 5;;
esac
host=$ac_cv_host
ac_save_IFS=$IFS; IFS='- '
set x $ac_cv_host
shift
host_cpu=$1
host_vendor=$2

```

```

shift; shift
# Remember, the first character of IFS is used to create $*,
# except with old shells:
host_os=$*
IFS=$_ac_save_IFS
case $host_os in *\ *) host_os=`echo "$host_os" | sed 's/ /-/g'`;
esac

ac_config_headers="$ac_config_headers config.h"

am__api_version='1.12'

# Find a good install program.  We prefer a C program (faster),
# so one script is as good as another.  But avoid the broken or
# incompatible versions:
# SysV /etc/install, /usr/sbin/install
# SunOS /usr/etc/install
# IRIX /sbin/install
# AIX /bin/install
# AmigaOS /C/install, which installs bootblocks on floppy discs
# AIX 4 /usr/bin/installbsd, which doesn't work without a -g flag
# AFS /usr/afsws/bin/install, which mishandles nonexistent args
# SVR4 /usr/ucb/install, which tries to use the nonexistent group
"staff"
# OS/2's system install, which has a completely different semantic
# ./install, which can be erroneously created by make from
./install.sh.
# Reject install programs that cannot install multiple files.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for a BSD-compatible
install" >&5
$as_echo_n "checking for a BSD-compatible install... " >&6; }
if test -z "$INSTALL"; then
if ${ac_cv_path_install+:} false; then :
  $as_echo_n "(cached) " >&6
else
  as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  # Account for people who put trailing slashes in PATH elements.
case $as_dir/ in #((
  ./ | ../ | /[cC]/* | \
  /etc/* | /usr/sbin/* | /usr/etc/* | /sbin/* | /usr/afsws/bin/* | \
  ?:[\\/]os2[\\/]install[\\/] * | ?:[\\/]OS2[\\/]INSTALL[\\/] * | \
  /usr/ucb/* ) ;;
*)
  # OSF1 and SCO ODT 3.0 have their own names for install.

```

```

# Don't use installbsd from OSF since it installs stuff as root
# by default.
for ac_prog in ginstall scoinst install; do
  for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_prog$ac_exec_ext"; then
      if test $ac_prog = install &&
        grep dspmsg "$as_dir/$ac_prog$ac_exec_ext" >/dev/null 2>&1;
then
      # AIX install.  It has an incompatible calling convention.
      :
    elif test $ac_prog = install &&
      grep pwplus "$as_dir/$ac_prog$ac_exec_ext" >/dev/null 2>&1;
then
      # program-specific install script used by HP pwplus--don't
      use.
      :
    else
      rm -rf conftest.one conftest.two conftest.dir
      echo one > conftest.one
      echo two > conftest.two
      mkdir conftest.dir
      if "$as_dir/$ac_prog$ac_exec_ext" -c conftest.one
conftest.two "`pwd`/conftest.dir" &&
        test -s conftest.one && test -s conftest.two &&
        test -s conftest.dir/conftest.one &&
        test -s conftest.dir/conftest.two
      then
        ac_cv_path_install="$as_dir/$ac_prog$ac_exec_ext -c"
        break 3
      fi
    fi
  fi
done
done
;;
esac

done
IFS=$as_save_IFS

rm -rf conftest.one conftest.two conftest.dir

fi
if test "${ac_cv_path_install+set}" = set; then
  INSTALL=$ac_cv_path_install
else
  # As a last resort, use the slow shell script.  Don't cache a
  # value for INSTALL within a source directory, because that will
  # break other packages using the cache if that directory is
  # removed, or if the value is a relative name.
  INSTALL=$ac_install_sh
fi

```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $INSTALL" >&5
$as_echo "$INSTALL" >&6; }

# Use test -z because SunOS4 sh mishandles braces in ${var-val}.
# It thinks the first close brace ends the variable substitution.
test -z "$INSTALL_PROGRAM" && INSTALL_PROGRAM='${INSTALL}'

test -z "$INSTALL_SCRIPT" && INSTALL_SCRIPT='${INSTALL}'

test -z "$INSTALL_DATA" && INSTALL_DATA='${INSTALL} -m 644'

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether build
environment is sane" >&5
$as_echo_n "checking whether build environment is sane... " >&6; }
# Reject unsafe characters in $srcdir or the absolute working
directory
# name.  Accept space and tab only in the latter.
am_lf='
'
case `pwd` in
  *[\ \ \#\ \&\ \'\ \`\$am_lf]*)
    as_fn_error $? "unsafe absolute working directory name" "$LINENO"
5;;
esac
case $srcdir in
  *[\ \ \#\ \&\ \'\ \`\$am_lf\ \]*)
    as_fn_error $? "unsafe srcdir value: '$srcdir'" "$LINENO" 5;;
esac

# Do 'set' in a subshell so we don't clobber the current shell's
# arguments.  Must try -L first in case configure is actually a
# symlink; some systems play weird games with the mod time of symlinks
# (eg FreeBSD returns the mod time of the symlink's containing
# directory).
if (
  am_has_slept=no
  for am_try in 1 2; do
    echo "timestamp, slept: $am_has_slept" > conftest.file
    set X `ls -Lt "$srcdir/configure" conftest.file 2> /dev/null`
    if test "$*" = "X"; then
      # -L didn't work.
      set X `ls -t "$srcdir/configure" conftest.file`
    fi
    if test "$*" != "X $srcdir/configure conftest.file" \
      && test "$*" != "X conftest.file $srcdir/configure"; then

      # If neither matched, then we have a broken ls.  This can happen
      # if, for instance, CONFIG_SHELL is bash and it inherits a
      # broken ls alias from the environment.  This has actually
      # happened.  Such a system could not be considered "sane".

```

```

        as_fn_error $? "ls -t appears to fail.  Make sure there is not a
broken
alias in your environment" "$LINENO" 5
    fi
    if test "$2" = confptest.file || test $am_try -eq 2; then
        break
    fi
    # Just in case.
    sleep 1
    am_has_slept=yes
done
test "$2" = confptest.file
)
then
    # Ok.
    :
else
    as_fn_error $? "newly created file is older than distributed files!
Check your system clock" "$LINENO" 5
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
# If we didn't sleep, we still need to ensure time stamps of
config.status and
# generated files are strictly newer.
am_sleep_pid=
if grep 'slept: no' confptest.file >/dev/null 2>&1; then
    ( sleep 1 ) &
    am_sleep_pid=$!
fi

rm -f confptest.file

test "$program_prefix" != NONE &&

program_transform_name="s^&$program_prefix&;$program_transform_name"
# Use a double $ so make ignores it.
test "$program_suffix" != NONE &&

program_transform_name="s\&$program_suffix&;$program_transform_name"
# Double any \ or $.
# By default was `s,x,x', remove it if useless.
ac_script='s/[\\\$]/&&/g;s;/s,x,x,$//'
program_transform_name=`$as_echo "$program_transform_name" | sed
"$ac_script"`

# expand $ac_aux_dir to an absolute path
am_aux_dir=`cd $ac_aux_dir && pwd`

if test x"${MISSING+set}" != xset; then
    case $am_aux_dir in
    *\ * | *\ *)

```

```

        MISSING="\${SHELL} \"\$am_aux_dir/missing\" " ;;
*)
        MISSING="\${SHELL} \$am_aux_dir/missing" ;;
esac
fi
# Use eval to expand $SHELL
if eval "$MISSING --run true"; then
    am_missing_run="$MISSING --run "
else
    am_missing_run=
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: 'missing' script
is too old or missing" >&5
$as_echo "$as_me: WARNING: 'missing' script is too old or missing"
>&2;}
fi

if test x"${install_sh}" != xset; then
    case $am_aux_dir in
*\ * | *\ * *)
        install_sh="\${SHELL} '$am_aux_dir/install-sh'" ;;
*)
        install_sh="\${SHELL} $am_aux_dir/install-sh"
    esac
fi

# Installed binaries are usually stripped using 'strip' when the user
# run "make install-strip". However 'strip' might not be the right
# tool to use in cross-compilation environments, therefore Automake
# will honor the 'STRIP' environment variable to overrule this
program.
if test "$cross_compiling" != no; then
    if test -n "$ac_tool_prefix"; then
        # Extract the first word of "${ac_tool_prefix}strip", so it can be a
        program name with args.
        set dummy ${ac_tool_prefix}strip; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_prog_STRIP+:} false; then :
            $as_echo_n "(cached) " >&6
        else
            if test -n "$STRIP"; then
                ac_cv_prog_STRIP="$STRIP" # Let the user override the test.
            else
                as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
                for as_dir in $PATH
                do
                    IFS=$as_save_IFS
                    test -z "$as_dir" && as_dir=.
                    for ac_exec_ext in ' ' $ac_executable_extensions; do
                        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                            ac_cv_prog_STRIP="${ac_tool_prefix}strip"
                        fi
                    done
                done
            fi
        fi
    fi
fi

```

```

        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
STRIP=$ac_cv_prog_STRIP
if test -n "$STRIP"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $STRIP" >&5
$as_echo "$STRIP" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_STRIP"; then
    ac_ct_STRIP=$STRIP
    # Extract the first word of "strip", so it can be a program name
    with args.
    set dummy strip; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_STRIP+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_STRIP"; then
            ac_cv_prog_ac_ct_STRIP="$ac_ct_STRIP" # Let the user override the
            test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '$ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_ac_ct_STRIP="strip"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

fi
fi

```



```

fi
ac_ct_STRIP=$ac_cv_prog_ac_ct_STRIP
if test -n "$ac_ct_STRIP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_STRIP" >&5
$as_echo "$ac_ct_STRIP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_STRIP" = x; then
    STRIP=":"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    STRIP=$ac_ct_STRIP
  fi
else
  STRIP="$ac_cv_prog_STRIP"
fi

fi
INSTALL_STRIP_PROGRAM="\$(install_sh) -c -s"

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for a thread-safe
mkdir -p" >&5
$as_echo_n "checking for a thread-safe mkdir -p... " >&6; }
if test -z "$MKDIR_P"; then
  if ${ac_cv_path_mkdir+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH$PATH_SEPARATOR/opt/sfw/bin
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_prog in mkdir gmkdir; do
    for ac_exec_ext in '' $ac_executable_extensions; do
      as_fn_executable_p "$as_dir/$ac_prog$ac_exec_ext" || continue
      case `"$as_dir/$ac_prog$ac_exec_ext" --version 2>&1` in #(
        'mkdir (GNU coreutils) '* | \
        'mkdir (coreutils) '* | \
        'mkdir (fileutils) '4.1*)
        ac_cv_path_mkdir=$as_dir/$ac_prog$ac_exec_ext
        break 3;;
      esac
    done
  done
fi

```

```

        done
        done
    done
IFS=$as_save_IFS

fi

test -d ./--version && rmdir ./--version
if test "${ac_cv_path_mkdir+set}" = set; then
    MKDIR_P="$ac_cv_path_mkdir -p"
else
    # As a last resort, use the slow shell script. Don't cache a
    # value for MKDIR_P within a source directory, because that will
    # break other packages using the cache if that directory is
    # removed, or if the value is a relative name.
    MKDIR_P="$ac_install_sh -d"
fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $MKDIR_P" >&5
$as_echo "$MKDIR_P" >&6; }

for ac_prog in gawk mawk nawk awk
do
    # Extract the first word of "$ac_prog", so it can be a program name
    with args.
    set dummy $ac_prog; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_AWK+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$AWK"; then
            ac_cv_prog_AWK="$AWK" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '$ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_AWK="$ac_prog"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS
        fi
    fi
fi

```

```

AWK=$ac_cv_prog_AWK
if test -n "$AWK"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $AWK" >&5
$as_echo "$AWK" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  test -n "$AWK" && break
done

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether ${MAKE-make}
sets \$(MAKE)" >&5
$as_echo_n "checking whether ${MAKE-make} sets \$(MAKE)... " >&6; }
set x ${MAKE-make}
ac_make=`$as_echo "$2" | sed 's/+/p/g; s/[^a-zA-Z0-9_]/_/g`
if eval \${ac_cv_prog_make_${ac_make}_set+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat >conftest.make <<\_ACEOF
SHELL = /bin/sh
all:
    @echo '@@@@%=%$(MAKE)=@@@@%'
\_ACEOF
# GNU make sometimes prints "make[1]: Entering ...", which would
confuse us.
case ` ${MAKE-make} -f conftest.make 2>/dev/null ` in
  *@@@@%=?*@@@@%*)
    eval ac_cv_prog_make_${ac_make}_set=yes;;
  *)
    eval ac_cv_prog_make_${ac_make}_set=no;;
esac
rm -f conftest.make
fi
if eval test \${ac_cv_prog_make_${ac_make}_set} = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
  SET_MAKE=
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  SET_MAKE="MAKE=${MAKE-make}"
fi

rm -rf .tst 2>/dev/null
mkdir .tst 2>/dev/null
if test -d .tst; then
  am__leading_dot=.
else
  am__leading_dot=_

```

```

fi
rmdir .tst 2>/dev/null

if test "`cd $srcdir && pwd`" != "`pwd`; then
  # Use -I$(srcdir) only when $(srcdir) != ., so that make's output
  # is not polluted with repeated "-I."
  am__isrc=' -I$(srcdir)'
  # test to see if srcdir already configured
  if test -f $srcdir/config.status; then
    as_fn_error $? "source directory already configured; run \"make
distclean\" there first" "$LINENO" 5
  fi
fi

# test whether we have cygpath
if test -z "$CYGPATH_W"; then
  if (cygpath --version) >/dev/null 2>/dev/null; then
    CYGPATH_W='cygpath -w'
  else
    CYGPATH_W=echo
  fi
fi

# Define the identity of the package.
PACKAGE='dbus'
VERSION='1.6.8'

cat >>confdefs.h <<_ACEOF
#define PACKAGE "$PACKAGE"
_ACEOF

cat >>confdefs.h <<_ACEOF
#define VERSION "$VERSION"
_ACEOF

# Some tools Automake needs.

ACLOCAL=${ACLOCAL-"${am_missing_run}aclocal-${am__api_version}"}

AUTOCONF=${AUTOCONF-"${am_missing_run}autoconf"}

AUTOMAKE=${AUTOMAKE-"${am_missing_run}automake-${am__api_version}"}

AUTOHEADER=${AUTOHEADER-"${am_missing_run}autoheader"}

```

```

MAKEINFO=${MAKEINFO-"${am_missing_run}makeinfo"}

# For better backward compatibility.  To be removed once Automake
1.9.x
# dies out for good.  For more background, see:
# <http://lists.gnu.org/archive/html/automake/2012-07/msg00001.html>
# <http://lists.gnu.org/archive/html/automake/2012-07/msg00014.html>
mkdir_p='${MKDIR_P}'

# We need awk for the "check" target.  The system "awk" is bad on
# some platforms.
# Always define AMTAR for backward compatibility.  Yes, it's still
used
# in the wild :-( We should find a proper way to deprecate it ...
AMTAR='${TAR-tar}'

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to create a
ustar tar archive" >&5
$as_echo_n "checking how to create a ustar tar archive... " >&6; }
# Loop over all known methods to create a tar archive until one works.
_am_tools='gnutar plaintar cpio pax none'
_am_tools=${am_cv_prog_tar_ustar-$_am_tools}
# Do not fold the above two line into one, because Tru64 sh and
# Solaris sh will not grok spaces in the rhs of '-'.
for _am_tool in $_am_tools
do
  case $_am_tool in
    gnutar)
      for _am_tar in tar gnutar gtar;
      do
        { echo "$as_me:$LINENO: $_am_tar --version" >&5
($ _am_tar --version) >&5 2>&5
ac_status=$?
echo "$as_me:$LINENO: \$_? = $ac_status" >&5
(exit $ac_status); } && break
        done
        am__tar="$_am_tar --format=ustar -chf - "'"$stardir"'"
        am__tar_="$_am_tar --format=ustar -chf - "'"$stardir"'"
        am__untar="$_am_tar -xf -"
        ;;
    plaintar)
      # Must skip GNU tar: if it does not support --format= it doesn't
create
      # ustar tarball either.
      (tar --version) >/dev/null 2>&1 && continue
      am__tar='tar chf - "'"$stardir"'"
      am__tar_='tar chf - "$stardir"'
      am__untar='tar xf -'
      ;;
    pax)
      am__tar='pax -L -x ustar -w "'"$stardir"'"

```

```

    am__tar='pax -L -x ustar -w "$stardir"'
    am__untar='pax -r'
    ;;
cpio)
    am__tar='find "$stardir" -print | cpio -o -H ustar -L'
    am__tar_='find "$stardir" -print | cpio -o -H ustar -L'
    am__untar='cpio -i -H ustar -d'
    ;;
none)
    am__tar=false
    am__tar_=false
    am__untar=false
    ;;
esac

# If the value was cached, stop now.  We just wanted to have am__tar
# and am__untar set.
test -n "${am_cv_prog_tar_ustar}" && break

# tar/untar a dummy directory, and stop if the command works
rm -rf confptest.dir
mkdir confptest.dir
echo GrepMe > confptest.dir/file
{ echo "$as_me:$LINENO: tardir=confptest.dir && eval $am__tar_
>confptest.tar" >&5
  (tardir=confptest.dir && eval $am__tar_ >confptest.tar) >&5 2>&5
  ac_status=$?
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  (exit $ac_status); }
rm -rf confptest.dir
if test -s confptest.tar; then
  { echo "$as_me:$LINENO: $am__untar <confptest.tar" >&5
    ($am__untar <confptest.tar) >&5 2>&5
    ac_status=$?
    echo "$as_me:$LINENO: \$? = $ac_status" >&5
    (exit $ac_status); }
  grep GrepMe confptest.dir/file >/dev/null 2>&1 && break
fi
done
rm -rf confptest.dir

if ${am_cv_prog_tar_ustar+:} false; then :
  $as_echo_n "(cached) " >&6
else
  am_cv_prog_tar_ustar=$_am_tool
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_prog_tar_ustar" >&5
  $as_echo "$am_cv_prog_tar_ustar" >&6; }

```

```
GETTEXT_PACKAGE=dbus-1
```

```
cat >>confdefs.h <<_ACEOF
```

```
#define GETTEXT_PACKAGE "$GETTEXT_PACKAGE"  
_ACEOF
```

```
# By default, rebuild autotools files on demand; only use ./missing if  
the  
# user says --disable-maintainer-mode (some distributions like to do  
this)
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to enable  
maintainer-specific portions of Makefiles" >&5  
$as_echo_n "checking whether to enable maintainer-specific portions of  
Makefiles... " >&6; }
```

```
  # Check whether --enable-maintainer-mode was given.  
if test "${enable_maintainer_mode+set}" = set; then :  
  enableval=$enable_maintainer_mode; USE_MAINTAINER_MODE=$enableval  
else  
  USE_MAINTAINER_MODE=yes  
fi
```

```
  { $as_echo "$as_me:${as_lineno-$LINENO}: result:  
$USE_MAINTAINER_MODE" >&5  
$as_echo "$USE_MAINTAINER_MODE" >&6; }  
  if test $USE_MAINTAINER_MODE = yes; then  
    MAINTAINER_MODE_TRUE=  
    MAINTAINER_MODE_FALSE='#'  
else  
    MAINTAINER_MODE_TRUE='#'  
    MAINTAINER_MODE_FALSE=  
fi
```

```
  MAINT=$MAINTAINER_MODE_TRUE
```

```
# Check whether --enable-silent-rules was given.  
if test "${enable_silent_rules+set}" = set; then :  
  enableval=$enable_silent_rules;  
fi
```

```
case $enable_silent_rules in # (((  
  yes) AM_DEFAULT_VERBOSITY=0;;  
  no) AM_DEFAULT_VERBOSITY=1;;  
  *) AM_DEFAULT_VERBOSITY=0;;
```

```

esac
am_make=${MAKE-make}
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking whether $am_make
supports nested variables" >&5
$sas_echo_n "checking whether $am_make supports nested variables... "
>&6; }
if ${am_cv_make_support_nested_variables+:} false; then :
    $sas_echo_n "(cached) " >&6
else
    if $sas_echo 'TRUE=$(BAR$(V))
BAR0=false
BAR1=true
V=1
am__doit:
    @$(TRUE)
.PHONY: am__doit' | $am_make -f - >/dev/null 2>&1; then
    am_cv_make_support_nested_variables=yes
else
    am_cv_make_support_nested_variables=no
fi
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$am_cv_make_support_nested_variables" >&5
$sas_echo "$am_cv_make_support_nested_variables" >&6; }
if test $am_cv_make_support_nested_variables = yes; then
    AM_V='$(V)'
    AM_DEFAULT_V='$(AM_DEFAULT_VERBOSITY)'
else
    AM_V=$AM_DEFAULT_VERBOSITY
    AM_DEFAULT_V=$AM_DEFAULT_VERBOSITY
fi
AM_BACKSLASH='\ '

```

```

cat >>confdefs.h <<_ACEOF
#define DBUS_DAEMON_NAME "dbus-daemon"
_ACEOF

```

```

# libtool versioning - this applies to libdbus
#
# See
http://sources.redhat.com/autobook/autobook/autobook\_91.html#SEC91 for
details
#

```

```

## increment if the interface has additions, changes, removals.
LT_CURRENT=10

```

```

## increment any time the source changes; set to
## 0 if you increment CURRENT

```



```
LT_REVISION=2
```

```
## increment if any interfaces have been added; set to 0  
## if any interfaces have been changed or removed. removal has  
## precedence over adding, so set to 0 if both happened.  
LT_AGE=7
```

```
DBUS_MAJOR_VERSION=1  
DBUS_MINOR_VERSION=6  
DBUS_MICRO_VERSION=8  
DBUS_VERSION=1.6.8
```

```
ac_ext=c  
ac_cpp='$CPP $CPPFLAGS'  
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'  
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS  
conftest.$ac_ext $LIBS >&5'  
ac_compiler_gnu=$ac_cv_c_compiler_gnu  
if test -n "$ac_tool_prefix"; then  
  # Extract the first word of "${ac_tool_prefix}gcc", so it can be a  
  program name with args.  
  set dummy ${ac_tool_prefix}gcc; ac_word=$2  
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5  
  $as_echo_n "checking for $ac_word... " >&6; }  
  if ${ac_cv_prog_CC+:} false; then :  
    $as_echo_n "(cached) " >&6  
  else  
    if test -n "$CC"; then  
      ac_cv_prog_CC="$CC" # Let the user override the test.  
    else  
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR  
      for as_dir in $PATH  
      do  
        IFS=$as_save_IFS  
        test -z "$as_dir" && as_dir=.  
        for ac_exec_ext in ' $ac_executable_extensions; do  
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then  
            ac_cv_prog_CC="${ac_tool_prefix}gcc"  
            $as_echo "$as_me:${as_lineno-$LINENO}: found  
$as_dir/$ac_word$ac_exec_ext" >&5  
            break 2  
          fi  
        done  
      done  
    fi  
  fi  
done
```

```

done
IFS=$as_save_IFS

fi
fi
CC=$ac_cv_prog_CC
if test -n "$CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
$as_echo "$CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_CC"; then
  ac_ct_CC=$CC
  # Extract the first word of "gcc", so it can be a program name with
  args.
  set dummy gcc; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_CC+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_CC"; then
      ac_cv_prog_ac_ct_CC="$ac_ct_CC" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_CC="gcc"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

fi
fi
ac_ct_CC=$ac_cv_prog_ac_ct_CC
if test -n "$ac_ct_CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_CC" >&5
$as_echo "$ac_ct_CC" >&6; }
else

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    if test "x$ac_ct_CC" = x; then
        CC=""
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        CC=$ac_ct_CC
    fi
else
    CC="$ac_cv_prog_CC"
fi

if test -z "$CC"; then
    if test -n "$ac_tool_prefix"; then
        # Extract the first word of "${ac_tool_prefix}cc", so it can be a
program name with args.
set dummy ${ac_tool_prefix}cc; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_CC+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -n "$CC"; then
        ac_cv_prog_CC="$CC" # Let the user override the test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '$ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_CC="${ac_tool_prefix}cc"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
    done
IFS=$as_save_IFS

fi
fi

```

```

CC=$ac_cv_prog_CC
if test -n "$CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
$as_echo "$CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
fi
if test -z "$CC"; then
  # Extract the first word of "cc", so it can be a program name with
  args.
  set dummy cc; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_CC+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$CC"; then
      ac_cv_prog_CC="$CC" # Let the user override the test.
    else
      ac_prog_rejected=no
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            if test "$as_dir/$ac_word$ac_exec_ext" = "/usr/ucb/cc"; then
              ac_prog_rejected=yes
              continue
            fi
            ac_cv_prog_CC="cc"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS
    fi

    if test $ac_prog_rejected = yes; then
      # We found a bogon in the path, so make sure we never use it.
      set dummy $ac_cv_prog_CC
      shift
      if test $# != 0; then
        # We chose a different compiler from the bogus one.
        # However, it has the same basename, so the bogon will be chosen

```

```

    # first if we set CC to just the basename; use the full file name.
    shift
    ac_cv_prog_CC="$sas_dir/$ac_word${1+' '}$@"
  fi
fi
fi
fi
CC=$ac_cv_prog_CC
if test -n "$CC"; then
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $CC" >&5
  $sas_echo "$CC" >&6; }
else
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
  $sas_echo "no" >&6; }
fi

fi
if test -z "$CC"; then
  if test -n "$ac_tool_prefix"; then
    for ac_prog in cl.exe
    do
      # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
      be a program name with args.
      set dummy $ac_tool_prefix$ac_prog; ac_word=$2
      { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $sas_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_CC+:} false; then :
        $sas_echo_n "(cached) " >&6
      else
        if test -n "$CC"; then
          ac_cv_prog_CC="$CC" # Let the user override the test.
        else
          as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
          for as_dir in $PATH
          do
            IFS=$as_save_IFS
            test -z "$as_dir" && as_dir=.
            for ac_exec_ext in '' $ac_executable_extensions; do
              if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                ac_cv_prog_CC="$ac_tool_prefix$ac_prog"
                $sas_echo "$sas_me:${as_lineno-$LINENO}: found
                $as_dir/$ac_word$ac_exec_ext" >&5
                break 2
              fi
            done
          done
          IFS=$as_save_IFS
        fi
      fi
    done
  fi
  CC=$ac_cv_prog_CC

```

```

if test -n "$CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CC" >&5
$as_echo "$CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    test -n "$CC" && break
done
fi
if test -z "$CC"; then
  ac_ct_CC=$CC
  for ac_prog in cl.exe
do
  # Extract the first word of "$ac_prog", so it can be a program name
with args.
set dummy $ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_CC+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_CC"; then
    ac_cv_prog_ac_ct_CC="$ac_ct_CC" # Let the user override the test.
else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_ac_ct_CC="$ac_prog"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_CC=$ac_cv_prog_ac_ct_CC
if test -n "$ac_ct_CC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_CC" >&5
$as_echo "$ac_ct_CC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

```

```

fi

    test -n "$ac_ct_CC" && break
done

    if test "x$ac_ct_CC" = x; then
        CC=""
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        CC=$ac_ct_CC
    fi
fi

fi

test -z "$CC" && { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in
`$ac_pwd':" >&5
$as_echo "$as_me: error: in `$ac_pwd':" >&2;}
as_fn_error $? "no acceptable C compiler found in $PATH
See `config.log' for more details" "$LINENO" 5; }

# Provide some information about the compiler.
$as_echo "$as_me:${as_lineno-$LINENO}: checking for C compiler
version" >&5
set X $ac_compile
ac_compiler=$2
for ac_option in --version -v -V -qversion; do
    { { ac_try="$ac_compiler $ac_option >&5"
case "($ac_try" in
*`* | *`* | *`*) ac_try_echo=\$ac_try;;
*) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="`"\$as_me:${as_lineno-$LINENO}: $ac_try_echo`"
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_compiler $ac_option >&5") 2>conftest.err
    ac_status=$?
    if test -s conftest.err; then
        sed '10a\
... rest of stderr output deleted ...
    10q' conftest.err >conftest.er1
        cat conftest.er1 >&5
    fi
    rm -f conftest.er1 conftest.err

```

```

    $sas_echo "$sas_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
    test $ac_status = 0; }
done

cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
ac_clean_files_save=$ac_clean_files
ac_clean_files="$ac_clean_files a.out a.out.dSYM a.exe b.out"
# Try to create an executable without -o first, disregard a.out.
# It will help us diagnose broken compilers, and finding out an
intuition
# of exeext.
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking whether the C
compiler works" >&5
$sas_echo_n "checking whether the C compiler works... " >&6; }
ac_link_default=`$sas_echo "$ac_link" | sed 's/ -o *conftest[^\ ]*//'\`

# The possible output files:
ac_files="a.out conftest.exe conftest a.exe a_out.exe b.out
conftest.*"

ac_rmfiles=
for ac_file in $ac_files
do
    case $ac_file in
        *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj ) ;;
        * ) ac_rmfiles="$ac_rmfiles $ac_file";;
    esac
done
rm -f $ac_rmfiles

if { { ac_try="$ac_link_default"
case "($ac_try" in
*\"* | *\`* | *\`*) ac_try_echo=\$ac_try;;
*) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\`"\$sas_me:${as_lineno-$LINENO}: $ac_try_echo\`"
$sas_echo "$ac_try_echo"; } >&5
(eval "$ac_link_default") 2>&5
ac_status=$?
$sas_echo "$sas_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }; then :

```



```

# Autoconf-2.13 could set the ac_cv_exeext variable to `no'.
# So ignore a value of `no', otherwise this would lead to `EXEEXT =
no'
# in a Makefile. We should not override ac_cv_exeext if it was
cached,
# so that the user can short-circuit this test for compilers unknown
to
# Autoconf.
for ac_file in $ac_files ''
do
  test -f "$ac_file" || continue
  case $ac_file in
    *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj )
      ;;
    [ab].out )
      # We found the default executable, but exeext='' is most
      # certainly right.
      break;;
    *.* )
      if test "${ac_cv_exeext+set}" = set && test "$ac_cv_exeext" !=
no;
      then ;; else
        ac_cv_exeext=`expr "$ac_file" : '[^.]*(\..*)'`
      fi
      # We set ac_cv_exeext here because the later test for it is not
      # safe: cross compilers may not add the suffix if given an `-o'
      # argument, so we may need to know it at that point already.
      # Even if this section looks crufty: it has the advantage of
      # actually working.
      break;;
    * )
      break;;
  esac
done
test "$ac_cv_exeext" = no && ac_cv_exeext=

else
  ac_file=''
fi
if test -z "$ac_file"; then :
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
$as_echo "$as_me: failed program was:" >&5
sed 's/^/| /' conftest.$ac_ext >&5

{ { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':" >&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error 77 "C compiler cannot create executables
See `config.log' for more details" "$LINENO" 5; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5

```

```

$as_echo "yes" >&6; }
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for C compiler
default output file name" >&5
$as_echo_n "checking for C compiler default output file name... " >&6;
}
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_file" >&5
$as_echo "$ac_file" >&6; }
ac_exeext=$ac_cv_exeext

rm -f -r a.out a.out.dSYM a.exe conftest$ac_cv_exeext b.out
ac_clean_files=$ac_clean_files_save
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for suffix of
executables" >&5
$as_echo_n "checking for suffix of executables... " >&6; }
if { { ac_try="$ac_link"
case "($ac_try" in
  *\"* | *\\* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"\`\\$as_me:${as_lineno-$LINENO}: \$ac_try_echo\`"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link" ) 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; }; then :
  # If both `conftest.exe' and `conftest' are `present' (well,
  observable)
  # catch `conftest.exe'. For instance with Cygwin, `ls conftest' will
  # work properly (i.e., refer to `conftest.exe'), while it won't with
  # `rm'.
  for ac_file in conftest.exe conftest conftest.*; do
    test -f "$ac_file" || continue
    case $ac_file in
      *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM | *.o | *.obj ) ;;
      *.* ) ac_cv_exeext=`expr "$ac_file" : '[^.]*(\..*)'`
        break;;
      * ) break;;
    esac
  done
else
  { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error $? "cannot compute suffix of executables: cannot compile
and link
See `config.log' for more details" "$LINENO" 5; }
fi
rm -f conftest conftest$ac_cv_exeext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_exeext" >&5
$as_echo "$ac_cv_exeext" >&6; }

```

```

rm -f conftest.$ac_ext
EXEEXT=$ac_cv_exeext
ac_exeext=$EXEEXT
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdio.h>
int
main ()
{
FILE *f = fopen ("conftest.out", "w");
return ferror (f) || fclose (f) != 0;

;
return 0;
}
_ACEOF
ac_clean_files="$ac_clean_files conftest.out"
# Check that the compiler produces executables we can run.  If not,
either
# the compiler is broken, or we cross compile.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether we are cross
compiling" >&5
$as_echo_n "checking whether we are cross compiling... " >&6; }
if test "$cross_compiling" != yes; then
  { { ac_try="$ac_link"
case "($ac_try" in
  *\"* | *\\`* | *\\`*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: \$ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_link") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; }
  if { ac_try='./conftest$ac_cv_exeext'
  { { case "($ac_try" in
  *\"* | *\\`* | *\\`*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo=\"`\$as_me:${as_lineno-$LINENO}: \$ac_try_echo\"
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_try") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; }; }; then
    cross_compiling=no
  else
    if test "$cross_compiling" = maybe; then
      cross_compiling=yes
    else

```

```

        { { $sas_echo "$sas_me:${as_lineno-$LINENO}: error: in \"\$ac_pwd':"
>&5
$sas_echo "$sas_me: error: in \"\$ac_pwd':" >&2;}
as_fn_error $? "cannot run C compiled programs.
If you meant to cross compile, use \"--host".
See \"config.log\" for more details" "$LINENO" 5; }
    fi
    fi
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $cross_compiling" >&5
$sas_echo "$cross_compiling" >&6; }

rm -f confctest.$ac_ext confctest$ac_cv_exeext confctest.out
ac_clean_files=$ac_clean_files_save
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for suffix of object
files" >&5
$sas_echo_n "checking for suffix of object files... " >&6; }
if ${ac_cv_objext+:} false; then :
  $sas_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >>confctest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

  ;
  return 0;
}
_ACEOF
rm -f confctest.o confctest.obj
if { { ac_try="$ac_compile"
case "($ac_try" in
  *\"* | *\\* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$sas_me:${as_lineno-$LINENO}: $ac_try_echo\""
$sas_echo "$ac_try_echo"; } >&5
  (eval "$ac_compile") 2>&5
  ac_status=$?
  $sas_echo "$sas_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; }; then :
  for ac_file in confctest.o confctest.obj confctest.*; do
  test -f "$ac_file" || continue;
  case $ac_file in
    *.$ac_ext | *.xcoff | *.tds | *.d | *.pdb | *.xSYM | *.bb | *.bbg
| *.map | *.inf | *.dSYM ) ;;
    *) ac_cv_objext=`expr "$ac_file" : '.*\\.\\(.*)'`
      break;;
  esac
done

```

```

else
  $as_echo "$as_me: failed program was:" >&5
sed 's/^\| /' conftest.$ac_ext >&5

{ { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`${ac_pwd}':" >&5
$as_echo "$as_me: error: in \`${ac_pwd}':" >&2;}
as_fn_error $? "cannot compute suffix of object files: cannot compile
See \`config.log' for more details" "$LINENO" 5; }
fi
rm -f conftest.$ac_cv_objext conftest.$ac_ext
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_objext" >&5
$as_echo "$ac_cv_objext" >&6; }
OBJEXT=$ac_cv_objext
ac_objext=$OBJEXT
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether we are using
the GNU C compiler" >&5
$as_echo_n "checking whether we are using the GNU C compiler... " >&6;
}
if ${ac_cv_c_compiler_gnu+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{
#ifdef __GNUC__
  choke me
#endif

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_compiler_gnu=yes
else
  ac_compiler_gnu=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
ac_cv_c_compiler_gnu=$ac_compiler_gnu

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_c_compiler_gnu" >&5
$as_echo "$ac_cv_c_compiler_gnu" >&6; }
if test $ac_compiler_gnu = yes; then
  GCC=yes
else
  GCC=

```

```

fi
ac_test_CFLAGS=${CFLAGS+set}
ac_save_CFLAGS=$CFLAGS
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CC accepts
-g" >&5
$as_echo_n "checking whether $CC accepts -g... " >&6; }
if ${ac_cv_prog_cc_g+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_save_c_werror_flag=$ac_c_werror_flag
  ac_c_werror_flag=yes
  ac_cv_prog_cc_g=no
  CFLAGS="-g"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```

```

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_cv_prog_cc_g=yes
else
  CFLAGS=""
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```

```

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :

else
  ac_c_werror_flag=$ac_save_c_werror_flag
  CFLAGS="-g"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```

```

int
main ()
{

  ;

```

```

    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_prog_cc_g=yes
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
    ac_c_werror_flag=$ac_save_c_werror_flag
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_prog_cc_g" >&5
$as_echo "$ac_cv_prog_cc_g" >&6; }
if test "$ac_test_CFLAGS" = set; then
    CFLAGS=$ac_save_CFLAGS
elif test $ac_cv_prog_cc_g = yes; then
    if test "$GCC" = yes; then
        CFLAGS="-g -O2"
    else
        CFLAGS="-g"
    fi
else
    if test "$GCC" = yes; then
        CFLAGS="-O2"
    else
        CFLAGS=
    fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $CC option to
accept ISO C89" >&5
$as_echo_n "checking for $CC option to accept ISO C89... " >&6; }
if ${ac_cv_prog_cc_c89+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_cv_prog_cc_c89=no
ac_save_CC=$CC
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdarg.h>
#include <stdio.h>
struct stat;
/* Most of the following tests are stolen from RCS 5.7's src/conf.sh.
*/
struct buf { int x; };
FILE * (*rcsopen) (struct buf *, struct stat *, int);
static char *e (p, i)
    char **p;
    int i;
{
    return p[i];

```

```

}
static char *f (char * (*g) (char **, int), char **p, ...)
{
    char *s;
    va_list v;
    va_start (v,p);
    s = g (p, va_arg (v,int));
    va_end (v);
    return s;
}

/* OSF 4.0 Compaq cc is some sort of almost-ANSI by default.  It has
function prototypes and stuff, but not '\xHH' hex character
constants.
These don't provoke an error unfortunately, instead are silently
treated
as 'x'.  The following induces an error, until -std is added to get
proper ANSI mode.  Curiously '\x00'!='x' always comes out true, for
an
array size at least.  It's necessary to write '\x00'==0 to get
something
that's true only with -std.  */
int osf4_cc_array ['\x00' == 0 ? 1 : -1];

/* IBM C 6 for AIX is almost-ANSI by default, but it replaces macro
parameters
inside strings and character constants.  */
#define FOO(x) 'x'
int xlc6_cc_array[FOO(a) == 'x' ? 1 : -1];

int test (int i, double x);
struct s1 {int (*f) (int a);};
struct s2 {int (*f) (double a);};
int pairnames (int, char **, FILE *(*)(struct buf *, struct stat *,
int), int, int);
int argc;
char **argv;
int
main ()
{
return f (e, argv, 0) != argv[0] || f (e, argv, 1) != argv[1];
;
return 0;
}
_ACEOF
for ac_arg in '' -qlanglvl=extc89 -qlanglvl=ansi -std \
    -Ae "-Aa -D_HPUX_SOURCE" "-Xc -D__EXTENSIONS__"
do
    CC="$ac_save_CC $ac_arg"
    if ac_fn_c_try_compile "$LINENO"; then :
        ac_cv_prog_cc_c89=$ac_arg
fi

```



```

rm -f core conftest.err conftest.$ac_objext
  test "x$ac_cv_prog_cc_c89" != "xno" && break
done
rm -f conftest.$ac_ext
CC=$ac_save_CC

fi
# AC_CACHE_VAL
case "x$ac_cv_prog_cc_c89" in
  x)
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: none needed" >&5
$as_echo "none needed" >&6; } ;;
  xno)
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: unsupported" >&5
$as_echo "unsupported" >&6; } ;;
  *)
    CC="$CC $ac_cv_prog_cc_c89"
    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_prog_cc_c89" >&5
$as_echo "$ac_cv_prog_cc_c89" >&6; } ;;
esac
if test "x$ac_cv_prog_cc_c89" != xno; then :

fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
DEPDIR="$am__leading_dot"deps"

ac_config_commands="$ac_config_commands depfiles"

am_make=${MAKE-make}
cat > confinc << 'END'
am__doit:
  @echo this is the am__doit target
.PHONY: am__doit
END
# If we don't find an include directive, just comment out the code.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for style of include
used by $am_make" >&5
$as_echo_n "checking for style of include used by $am_make... " >&6; }
am__include="#"
am__quote=
_am_result=none
# First try GNU make style include.
echo "include confinc" > confmf
# Ignore all kinds of additional output from 'make'.

```

```

case `\$am_make -s -f confmf 2> /dev/null` in #(
*the\ am__doit\ target*)
  am__include=include
  am__quote=
  _am_result=GNU
  ;;
esac
# Now try BSD make style include.
if test "\$am__include" = "#"; then
  echo '.include "confinc"' > confmf
  case `\$am_make -s -f confmf 2> /dev/null` in #(
*the\ am__doit\ target*)
    am__include=.include
    am__quote="\\"
    _am_result=BSD
    ;;
  esac
fi

{ \$as_echo "\$as_me:${as_lineno-$LINENO}: result: \$_am_result" >&5
\$as_echo "\$_am_result" >&6; }
rm -f confinc confmf

# Check whether --enable-dependency-tracking was given.
if test "${enable_dependency_tracking+set}" = set; then :
  enableval=\$enable_dependency_tracking;
fi

if test "x\$enable_dependency_tracking" != xno; then
  am_depcomp="\$ac_aux_dir/depcomp"
  AMDEPBACKSLASH='\'
  am__nodep='_no'
fi
if test "x\$enable_dependency_tracking" != xno; then
  AMDEP_TRUE=
  AMDEP_FALSE='#'
else
  AMDEP_TRUE='#'
  AMDEP_FALSE=
fi

depcc="\$CC"  am_compiler_list=

{ \$as_echo "\$as_me:${as_lineno-$LINENO}: checking dependency style of
\$depcc" >&5
\$as_echo_n "checking dependency style of \$depcc... " >&6; }
if {\$am_cv_CC_dependencies_compiler_type+:} false; then :
  \$as_echo_n "(cached) " >&6
else

```

```

if test -z "$SAMDEP_TRUE" && test -f "$am_depcomp"; then
# We make a subdir and do the tests there.  Otherwise we can end up
# making bogus files that we don't know about and never remove.  For
# instance it was reported that on HP-UX the gcc test will end up
# making a dummy file named 'D' -- because '-MD' means "put the
output
# in D".
rm -rf confptest.dir
mkdir confptest.dir
# Copy depcomp to subdir because otherwise we won't find it if we're
# using a relative directory.
cp "$am_depcomp" confptest.dir
cd confptest.dir
# We will build objects and dependencies in a subdirectory because
# it helps to detect inapplicable dependency modes.  For instance
# both Tru64's cc and ICC support -MD to output dependencies as a
# side effect of compilation, but ICC will put the dependencies in
# the current directory while Tru64 will put them in the object
# directory.
mkdir sub

am_cv_CC_dependencies_compiler_type=none
if test "$am_compiler_list" = ""; then
    am_compiler_list=`sed -n 's/^#*\([a-zA-Z0-9]*\))$/\1/p' <
./depcomp`
fi
am_universal=false
case " $depcc " in #(
    *\ -arch\ *\ -arch\ *) am_universal=true ;;
esac

for depmode in $am_compiler_list; do
# Setup a source with many dependencies, because some compilers
# like to wrap large dependency lists on column 80 (with \), and
# we should not choose a depcomp mode which is confused by this.
#
# We need to recreate these files for each test, as the compiler
may
# overwrite some of them when testing with obscure command lines.
# This happens at least with the AIX C compiler.
: > sub/confptest.c
for i in 1 2 3 4 5 6; do
    echo '#include "conftst'$i'.h"' >> sub/confptest.c
    # Using ": > sub/conftst$i.h" creates only sub/conftst1.h with
    # Solaris 10 /bin/sh.
    echo '/* dummy */' > sub/conftst$i.h
done
echo "${am__include} ${am__quote}sub/confptest.Po${am__quote}" >
confmf

# We check with '-c' and '-o' for the sake of the "dashmstdout"

```

```

# mode. It turns out that the SunPro C++ compiler does not
properly
# handle '-M -o', and we need to detect this. Also, some Intel
# versions had trouble with output in subdirs.
am__obj=sub/confptest.${OBJEXT}-o}
am__minus_obj="-o $am__obj"
case $depmode in
gcc)
# This depmode causes a compiler race in universal mode.
test "$am__universal" = false || continue
;;
nosideeffect)
# After this tag, mechanisms are not by side-effect, so they'll
# only be used when explicitly requested.
if test "x$enable_dependency_tracking" = xyes; then
continue
else
break
fi
;;
msvc7 | msvc7msys | msvisualcpp | msvcmsys)
# This compiler won't grok '-c -o', but also, the minuso test
has
# not run yet. These depmodes are late enough in the game, and
# so weak that their functioning should not be impacted.
am__obj=confptest.${OBJEXT}-o}
am__minus_obj=
;;
none) break ;;
esac
if depmode=$depmode \
source=sub/confptest.c object=$am__obj \
depfile=sub/confptest.Po tmpdepfile=sub/confptest.TPo \
$SHELL ./depcomp $depcc -c $am__minus_obj sub/confptest.c \
>/dev/null 2>confptest.err &&
grep sub/conftst1.h sub/confptest.Po > /dev/null 2>&1 &&
grep sub/conftst6.h sub/confptest.Po > /dev/null 2>&1 &&
grep $am__obj sub/confptest.Po > /dev/null 2>&1 &&
${MAKE-make} -s -f confmf > /dev/null 2>&1; then
# icc doesn't choke on unknown options, it will just issue
warnings
# or remarks (even with -Werror). So we grep stderr for any
message
# that says an option was ignored or not supported.
# When given -MP, icc 7.0 and 7.1 complain thusly:
# icc: Command line warning: ignoring option '-M'; no argument
required
# The diagnosis changed in icc 8.0:
# icc: Command line remark: option '-MP' not supported
if (grep 'ignoring option' confptest.err ||
grep 'not supported' confptest.err) >/dev/null 2>&1; then ;;
else

```

```

        am_cv_CC_dependencies_compiler_type=$depmode
        break
    fi
fi
done

cd ..
rm -rf confptest.dir
else
    am_cv_CC_dependencies_compiler_type=none
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_CC_dependencies_compiler_type" >&5
$as_echo "$am_cv_CC_dependencies_compiler_type" >&6; }
CCDEPMODE=depmode=$am_cv_CC_dependencies_compiler_type

if
    test "x$enable_dependency_tracking" != xno \
    && test "$am_cv_CC_dependencies_compiler_type" = gcc3; then
    am__fastdepCC_TRUE=
    am__fastdepCC_FALSE='#'
else
    am__fastdepCC_TRUE='#'
    am__fastdepCC_FALSE=
fi

if test "x$CC" != xcc; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CC and cc
understand -c and -o together" >&5
$as_echo_n "checking whether $CC and cc understand -c and -o
together... " >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether cc
understands -c and -o together" >&5
$as_echo_n "checking whether cc understands -c and -o together... "
>&6; }
fi
set dummy $CC; ac_cc=`$as_echo "$2" |
    sed 's/[^a-zA-Z0-9_]/_/g;s/^[0-9]/_/`
if eval \${ac_cv_prog_cc_${ac_cc}_c_o+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */

int
main ()
{

```

```

;
return 0;
}
_ACEOF
# Make sure it works both with $CC and with simple cc.
# We do the test twice because some compilers refuse to overwrite an
# existing .o file with -o, though they will create one.
ac_try='$CC -c conftest.$ac_ext -o conftest2.$ac_objext >&5'
rm -f conftest2.*
if { { case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_try") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; } &&
  test -f conftest2.$ac_objext && { { case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_try") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; };
then
  eval ac_cv_prog_cc_${ac_cc}_c_o=yes
  if test "x$CC" != xcc; then
    # Test first that cc exists at all.
    if { ac_try='cc -c conftest.$ac_ext >&5'
    { { case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
  (eval "$ac_try") 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \\$? = $ac_status" >&5
  test $ac_status = 0; }; }; then
    ac_try='cc -c conftest.$ac_ext -o conftest2.$ac_objext >&5'
    rm -f conftest2.*
    if { { case "($ac_try" in
  *\"* | *\\`* | *\\*) ac_try_echo=\\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5

```

```

(eval "$ac_try") 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; } &&
    test -f conftest2.$ac_objext && { { case "((($ac_try" in
*\"* | *\`* | *\`*) ac_try_echo=\$ac_try;;
*) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\` \$as_me:${as_lineno-$LINENO}: $ac_try_echo\`"
$as_echo "$ac_try_echo"; } >&5
(eval "$ac_try") 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; };
    then
        # cc works too.
        :
        else
            # cc exists but doesn't like -o.
            eval ac_cv_prog_cc_${ac_cc}_c_o=no
            fi
        fi
    fi
else
    eval ac_cv_prog_cc_${ac_cc}_c_o=no
fi
rm -f core conftest*

fi
if eval test \$ac_cv_prog_cc_${ac_cc}_c_o = yes; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

$as_echo "#define NO_MINUS_C_MINUS_O 1" >>confdefs.h

fi

# FIXME: we rely on the cache variable name because
# there is no other way.
set dummy $CC
am_cc=`echo $2 | sed 's/[^a-zA-Z0-9_]/_/g;s/^[0-9]/_/`
eval am_t=\$ac_cv_prog_cc_${am_cc}_c_o
if test "$am_t" != yes; then
    # Losing compiler, so override with the script.
    # FIXME: It is wrong to rewrite CC.
    # But if we don't then we get into trouble of one sort or another.
    # A longer-term fix would be to have automake use am__CC in this
case,
    # and then we could set am__CC="\$(top_srcdir)/compile \$(CC)"

```

```

    CC="$am_aux_dir/compile $CC"
fi

ac_ext=cpp
ac_cpp='$CXXCPP $CPPFLAGS'
ac_compile='$CXX -c $CXXFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CXX -o conftest$ac_exeext $CXXFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_cxx_compiler_gnu
if test -z "$CXX"; then
  if test -n "$CCC"; then
    CXX=$CCC
  else
    if test -n "$ac_tool_prefix"; then
      for ac_prog in g++ c++ gpp aCC CC cxx cc++ cl.exe FCC KCC RCC x1C_r
x1C
      do
        # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
        be a program name with args.
        set dummy $ac_tool_prefix$ac_prog; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_prog_CXX+:} false; then :
          $as_echo_n "(cached) " >&6
        else
          if test -n "$CXX"; then
            ac_cv_prog_CXX="$CXX" # Let the user override the test.
          else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
              IFS=$as_save_IFS
              test -z "$as_dir" && as_dir=.
              for ac_exec_ext in '' $ac_executable_extensions; do
                if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                  ac_cv_prog_CXX="$ac_tool_prefix$ac_prog"
                  $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                  break 2
                fi
              done
            done
            IFS=$as_save_IFS
          fi
        fi
      done
    fi
  fi
  CXX=$ac_cv_prog_CXX
  if test -n "$CXX"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $CXX" >&5
$as_echo "$CXX" >&6; }
  else

```



```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    test -n "$CXX" && break
done
fi
if test -z "$CXX"; then
    ac_ct_CXX=$CXX
    for ac_prog in g++ c++ gpp aCC CC cxx cc++ cl.exe FCC KCC RCC x1C_r
x1C
do
    # Extract the first word of "$ac_prog", so it can be a program name
with args.
    set dummy $ac_prog; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_CXX+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_CXX"; then
            ac_cv_prog_ac_ct_CXX="$ac_ct_CXX" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in ' $ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_ac_ct_CXX="$ac_prog"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

        fi
    fi
    ac_ct_CXX=$ac_cv_prog_ac_ct_CXX
    if test -n "$ac_ct_CXX"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_CXX" >&5
$as_echo "$ac_ct_CXX" >&6; }
    else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi
fi

```

```

    test -n "$ac_ct_CXX" && break
done

    if test "x$ac_ct_CXX" = x; then
        CXX="g++"
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        CXX=$ac_ct_CXX
    fi
fi

    fi
fi
# Provide some information about the compiler.
$as_echo "$as_me:${as_lineno-$LINENO}: checking for C++ compiler
version" >&5
set X $ac_compile
ac_compiler=$2
for ac_option in --version -v -V -qversion; do
    { { ac_try="$ac_compiler $ac_option >&5"
case "($ac_try" in
  *\"* | *\`* | *\\*) ac_try_echo=\$ac_try;;
  *) ac_try_echo=$ac_try;;
esac
eval ac_try_echo="\"\$as_me:${as_lineno-$LINENO}: $ac_try_echo\""
$as_echo "$ac_try_echo"; } >&5
    (eval "$ac_compiler $ac_option >&5") 2>conftest.err
    ac_status=$?
    if test -s conftest.err; then
        sed '10a\
... rest of stderr output deleted ...
        10q' conftest.err >conftest.er1
        cat conftest.er1 >&5
    fi
    rm -f conftest.er1 conftest.err
    $as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
    test $ac_status = 0; }
done

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether we are using
the GNU C++ compiler" >&5
$as_echo_n "checking whether we are using the GNU C++ compiler... "
>&6; }
if ${ac_cv_cxx_compiler_gnu+:} false; then :
    $as_echo_n "(cached) " >&6

```

```

else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{
#ifdef __GNUC__
  choke me
#endif

  ;
  return 0;
}
_ACEOF
if ac_fn_cxx_try_compile "$LINENO"; then :
  ac_compiler_gnu=yes
else
  ac_compiler_gnu=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
ac_cv_cxx_compiler_gnu=$ac_compiler_gnu

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_cxx_compiler_gnu" >&5
$as_echo "$ac_cv_cxx_compiler_gnu" >&6; }
if test $ac_compiler_gnu = yes; then
  GXX=yes
else
  GXX=
fi
ac_test_CXXFLAGS=${CXXFLAGS+set}
ac_save_CXXFLAGS=$CXXFLAGS
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CXX accepts
-g" >&5
$as_echo_n "checking whether $CXX accepts -g... " >&6; }
if ${ac_cv_prog_cxx_g+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_save_cxx_werror_flag=$ac_cxx_werror_flag
  ac_cxx_werror_flag=yes
  ac_cv_prog_cxx_g=no
  CXXFLAGS="-g"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

  ;

```

```

    return 0;
}
_ACEOF
if ac_fn_cxx_try_compile "$LINENO"; then :
    ac_cv_prog_cxx_g=yes
else
    CXXFLAGS=""
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_cxx_try_compile "$LINENO"; then :

else
    ac_cxx_werror_flag=$ac_save_cxx_werror_flag
    CXXFLAGS="-g"
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_cxx_try_compile "$LINENO"; then :
    ac_cv_prog_cxx_g=yes
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
    ac_cxx_werror_flag=$ac_save_cxx_werror_flag
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_prog_cxx_g"
>&5
$as_echo "$ac_cv_prog_cxx_g" >&6; }
if test "$ac_test_CXXFLAGS" = set; then
    CXXFLAGS=$ac_save_CXXFLAGS
elif test $ac_cv_prog_cxx_g = yes; then
    if test "$GXX" = yes; then
        CXXFLAGS="-g -O2"

```

```

else
    CXXFLAGS="-g"
fi
else
    if test "$GXX" = yes; then
        CXXFLAGS="-O2"
    else
        CXXFLAGS=
    fi
fi
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

depcc="$CXX"  am_compiler_list=

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking dependency style of
$depcc" >&5
$as_echo_n "checking dependency style of $depcc... " >&6; }
if ${am_cv_CXX_dependencies_compiler_type+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -z "$AMDEP_TRUE" && test -f "$am_depcomp"; then
        # We make a subdir and do the tests there.  Otherwise we can end up
        # making bogus files that we don't know about and never remove.  For
        # instance it was reported that on HP-UX the gcc test will end up
        # making a dummy file named 'D' -- because '-MD' means "put the
output
        # in D".
        rm -rf conftest.dir
        mkdir conftest.dir
        # Copy depcomp to subdir because otherwise we won't find it if we're
        # using a relative directory.
        cp "$am_depcomp" conftest.dir
        cd conftest.dir
        # We will build objects and dependencies in a subdirectory because
        # it helps to detect inapplicable dependency modes.  For instance
        # both Tru64's cc and ICC support -MD to output dependencies as a
        # side effect of compilation, but ICC will put the dependencies in
        # the current directory while Tru64 will put them in the object
        # directory.
        mkdir sub

        am_cv_CXX_dependencies_compiler_type=none
        if test "$am_compiler_list" = ""; then
            am_compiler_list=`sed -n 's/^#\([a-zA-Z0-9]*\))$/\1/p' <
./depcomp`
        fi
        am__universal=false

```

```

case " $depcc " in #(
    *\ -arch\ *\ -arch\ *) am__universal=true ;;
esac

for depmode in $am_compiler_list; do
    # Setup a source with many dependencies, because some compilers
    # like to wrap large dependency lists on column 80 (with \), and
    # we should not choose a depcomp mode which is confused by this.
    #
    # We need to recreate these files for each test, as the compiler
may # overwrite some of them when testing with obscure command lines.
    # This happens at least with the AIX C compiler.
    : > sub/confctest.c
    for i in 1 2 3 4 5 6; do
        echo '#include "confstt'$i'.h"' >> sub/confctest.c
        # Using ": > sub/confstt'$i'.h" creates only sub/confstt1.h with
        # Solaris 10 /bin/sh.
        echo '/* dummy */' > sub/confstt'$i'.h
    done
    echo "${am__include} ${am__quote}sub/confctest.Po${am__quote}" >
confmf

    # We check with '-c' and '-o' for the sake of the "dashmstdout"
    # mode. It turns out that the SunPro C++ compiler does not
properly # handle '-M -o', and we need to detect this. Also, some Intel
    # versions had trouble with output in subdirs.
    am__obj=sub/confctest.${OBJEXT-o}
    am__minus_obj="-o $am__obj"
    case $depmode in
gcc)
        # This depmode causes a compiler race in universal mode.
        test "$am__universal" = false || continue
        ;;
nosideeffect)
        # After this tag, mechanisms are not by side-effect, so they'll
        # only be used when explicitly requested.
        if test "x$enable_dependency_tracking" = xyes; then
            continue
        else
            break
        fi
        ;;
msvc7 | msvc7msys | msvisualcpp | msvcmsys)
        # This compiler won't grok '-c -o', but also, the minuso test
has # not run yet. These depmodes are late enough in the game, and
        # so weak that their functioning should not be impacted.
        am__obj=confctest.${OBJEXT-o}
        am__minus_obj=
        ;;

```

```

none) break ;;
esac
if depmode=$depmode \
  source=sub/confctest.c object=$am__obj \
  depfile=sub/confctest.Po tmpdepfile=sub/confctest.TPo \
  $SHELL ./depcomp $depcc -c $am__minus_obj sub/confctest.c \
  >/dev/null 2>confctest.err &&
  grep sub/confctst1.h sub/confctest.Po > /dev/null 2>&1 &&
  grep sub/confctst6.h sub/confctest.Po > /dev/null 2>&1 &&
  grep $am__obj sub/confctest.Po > /dev/null 2>&1 &&
  ${MAKE-make} -s -f confmf > /dev/null 2>&1; then
  # icc doesn't choke on unknown options, it will just issue
warnings
  # or remarks (even with -Werror). So we grep stderr for any
message
  # that says an option was ignored or not supported.
  # When given -MP, icc 7.0 and 7.1 complain thusly:
  #   icc: Command line warning: ignoring option '-M'; no argument
required
  # The diagnosis changed in icc 8.0:
  #   icc: Command line remark: option '-MP' not supported
  if (grep 'ignoring option' confctest.err ||
      grep 'not supported' confctest.err) >/dev/null 2>&1; then ;;
else
  am_cv_CXX_dependencies_compiler_type=$depmode
  break
fi
fi
done

cd ..
rm -rf confctest.dir
else
  am_cv_CXX_dependencies_compiler_type=none
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_CXX_dependencies_compiler_type" >&5
$as_echo "$am_cv_CXX_dependencies_compiler_type" >&6; }
CXXDEPMODE=depmode=$am_cv_CXX_dependencies_compiler_type

if
  test "x$enable_dependency_tracking" != xno \
  && test "$am_cv_CXX_dependencies_compiler_type" = gcc3; then
  am__fastdepCXX_TRUE=
  am__fastdepCXX_FALSE='#'
else
  am__fastdepCXX_TRUE='#'
  am__fastdepCXX_FALSE=
fi

```

```

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to run the C
preprocessor" >&5
$as_echo_n "checking how to run the C preprocessor... " >&6; }
# On Suns, sometimes $CPP names a directory.
if test -n "$CPP" && test -d "$CPP"; then
  CPP=
fi
if test -z "$CPP"; then
  if ${ac_cv_prog_CPP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    # Double quotes because CPP needs to be expanded
    for CPP in "$CC -E" "$CC -E -traditional-cpp" "/lib/cpp"
    do
      ac_preproc_ok=false
      for ac_c_preproc_warn_flag in ' yes
do
  # Use a header file that comes with gcc, so configuring glibc
  # with a fresh cross-compiler works.
  # Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
  # <limits.h> exists even on freestanding compilers.
  # On the NeXT, cc -E runs the code through the compiler's parser,
  # not just through cpp. "Syntax error" is here to catch this case.
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#ifdef __STDC__
# include <limits.h>
#else
# include <assert.h>
#endif
                Syntax error
      _ACEOF
      if ac_fn_c_try_cpp "$LINENO"; then :

else
  # Broken: fails on valid input.
  continue
fi
rm -f conftest.err conftest.i conftest.$ac_ext

  # OK, works on sane cases. Now check whether nonexistent headers
  # can be detected and how.
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```



```

#include <ac_nonexistent.h>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
    # Broken: success on invalid input.
    continue
else
    # Passes both tests.
    ac_preproc_ok=:
    break
fi
rm -f confptest.err confptest.i confptest.$ac_ext

done
# Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
rm -f confptest.i confptest.err confptest.$ac_ext
if $ac_preproc_ok; then :
    break
fi

    done
    ac_cv_prog_CPP=$CPP

fi
    CPP=$ac_cv_prog_CPP
else
    ac_cv_prog_CPP=$CPP
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $CPP" >&5
$as_echo "$CPP" >&6; }
ac_preproc_ok=false
for ac_c_preproc_warn_flag in ' yes
do
    # Use a header file that comes with gcc, so configuring glibc
    # with a fresh cross-compiler works.
    # Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
    # <limits.h> exists even on freestanding compilers.
    # On the NeXT, cc -E runs the code through the compiler's parser,
    # not just through cpp. "Syntax error" is here to catch this case.
    cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */
#ifdef __STDC__
# include <limits.h>
#else
# include <assert.h>
#endif

        Syntax error

_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :

else
    # Broken: fails on valid input.
    continue

```

```

fi
rm -f confptest.err confptest.i confptest.$ac_ext

# OK, works on sane cases. Now check whether nonexistent headers
# can be detected and how.
cat confdefs.h - <<_ACEOF >confptest.$ac_ext
/* end confdefs.h. */
#include <ac_nonexistent.h>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
# Broken: success on invalid input.
continue
else
# Passes both tests.
ac_preproc_ok=:
break
fi
rm -f confptest.err confptest.i confptest.$ac_ext

done
# Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
rm -f confptest.i confptest.err confptest.$ac_ext
if $ac_preproc_ok; then :

else
{ { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `'$ac_pwd':"
>&5
$as_echo "$as_me: error: in `'$ac_pwd':" >&2;}
as_fn_error $? "C preprocessor `'$CPP'" fails sanity check
See `config.log' for more details" "$LINENO" 5; }
fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS confptest.$ac_ext >&5'
ac_link='$CC -o confptest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
confptest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for grep that
handles long lines and -e" >&5
$as_echo_n "checking for grep that handles long lines and -e... " >&6;
}
if ${ac_cv_path_GREP+:} false; then :
$as_echo_n "(cached) " >&6
else
if test -z "$GREP"; then
ac_path_GREP_found=false
# Loop through the user's path and test for each of PROGRAMME-LIST
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH$PATH_SEPARATOR/usr/xpg4/bin

```

```

do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_prog in grep ggrep; do
    for ac_exec_ext in ' ' $ac_executable_extensions; do
      ac_path_GREP="$as_dir/$ac_prog$ac_exec_ext"
      as_fn_executable_p "$ac_path_GREP" || continue
    # Check for GNU ac_path_GREP and select it if it is found.
    # Check for GNU $ac_path_GREP
    case `"$ac_path_GREP" --version 2>&1` in
    *GNU*)
      ac_cv_path_GREP="$ac_path_GREP" ac_path_GREP_found=;;;
    *)
      ac_count=0
      $as_echo_n 0123456789 >"confptest.in"
      while :
      do
        cat "confptest.in" "confptest.in" >"confptest.tmp"
        mv "confptest.tmp" "confptest.in"
        cp "confptest.in" "confptest.nl"
        $as_echo 'GREP' >> "confptest.nl"
        "$ac_path_GREP" -e 'GREP$' -e '-(cannot match)-' < "confptest.nl"
      >"confptest.out" 2>/dev/null || break
        diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
        as_fn_arith $ac_count + 1 && ac_count=$as_val
        if test $ac_count -gt ${ac_path_GREP_max-0}; then
          # Best one so far, save it but keep looking for a better one
          ac_cv_path_GREP="$ac_path_GREP"
          ac_path_GREP_max=$ac_count
        fi
        # 10*(2^10) chars as input seems more than enough
        test $ac_count -gt 10 && break
      done
      rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
    esac

    $ac_path_GREP_found && break 3
  done
done
done
IFS=$as_save_IFS
if test -z "$ac_cv_path_GREP"; then
  as_fn_error $? "no acceptable grep could be found in
$PATH$PATH_SEPARATOR/usr/xpg4/bin" "$LINENO" 5
fi
else
  ac_cv_path_GREP=$GREP
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_GREP" >&5
$as_echo "$ac_cv_path_GREP" >&6; }

```

```

GREP="$ac_cv_path_GREP"

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for egrep" >&5
$as_echo_n "checking for egrep... " >&6; }
if ${ac_cv_path_EGREP+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if echo a | $GREP -E '(a|b)' >/dev/null 2>&1
  then ac_cv_path_EGREP="$GREP -E"
  else
    if test -z "$EGREP"; then
      ac_path_EGREP_found=false
      # Loop through the user's path and test for each of PROGNAMES_LIST
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH$PATH_SEPARATOR/usr/xpg4/bin
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_prog in egrep; do
          for ac_exec_ext in ' ' $ac_executable_extensions; do
            ac_path_EGREP="$as_dir/$ac_prog$ac_exec_ext"
            as_fn_executable_p "$ac_path_EGREP" || continue
          # Check for GNU ac_path_EGREP and select it if it is found.
          # Check for GNU $ac_path_EGREP
          case `"$ac_path_EGREP" --version 2>&1` in
          *GNU*)
            ac_cv_path_EGREP="$ac_path_EGREP" ac_path_EGREP_found=::;
          *)
            ac_count=0
            $as_echo_n 0123456789 >"confptest.in"
            while :
            do
              cat "confptest.in" "confptest.in" >"confptest.tmp"
              mv "confptest.tmp" "confptest.in"
              cp "confptest.in" "confptest.nl"
              $as_echo 'EGREP' >> "confptest.nl"
              "$ac_path_EGREP" 'EGREP$' < "confptest.nl" >"confptest.out"
            2>/dev/null || break
              diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
              as_fn_arith $ac_count + 1 && ac_count=$as_val
              if test $ac_count -gt ${ac_path_EGREP_max-0}; then
                # Best one so far, save it but keep looking for a better one
                ac_cv_path_EGREP="$ac_path_EGREP"
                ac_path_EGREP_max=$ac_count
              fi
              # 10*(2^10) chars as input seems more than enough
              test $ac_count -gt 10 && break
            done
            rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
          esac

```

```

        $ac_path_EGREP_found && break 3
    done
done
done
IFS=$as_save_IFS
    if test -z "$ac_cv_path_EGREP"; then
        as_fn_error $? "no acceptable egrep could be found in
$PATH$PATH_SEPARATOR/usr/xpg4/bin" "$LINENO" 5
    fi
else
    ac_cv_path_EGREP=$EGREP
fi

    fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_EGREP"
>&5
$as_echo "$ac_cv_path_EGREP" >&6; }
EGREP="$ac_cv_path_EGREP"

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for ANSI C header
files" >&5
$as_echo_n "checking for ANSI C header files... " >&6; }
if ${ac_cv_header_stdctype} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <float.h>

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_header_stdctype=yes
else
    ac_cv_header_stdctype=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

if test $ac_cv_header_stdctype = yes; then
    # SunOS 4.x string.h does not declare mem*, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext

```

```

/* end confdefs.h.  */
#include <string.h>

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "memchr" >/dev/null 2>&1; then :

else
    ac_cv_header_stdc=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdc = yes; then
    # ISC 2.0.2 stdlib.h does not declare free, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
#include <stdlib.h>

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "free" >/dev/null 2>&1; then :

else
    ac_cv_header_stdc=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdc = yes; then
    # /bin/cc in Irix-4.0.5 gets non-ANSI ctype macros unless using -
ansi.
    if test "$cross_compiling" = yes; then :
    :
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */
#include <ctype.h>
#include <stdlib.h>
#if ((' ' & 0xFF) == 0x020)
# define ISLOWER(c) ('a' <= (c) && (c) <= 'z')
# define TOUPPER(c) (ISLOWER(c) ? 'A' + ((c) - 'a') : (c))
#else
# define ISLOWER(c) \
    (('a' <= (c) && (c) <= 'i') \
    || ('j' <= (c) && (c) <= 'r') \
    || ('s' <= (c) && (c) <= 'z'))
# define TOUPPER(c) (ISLOWER(c) ? ((c) | 0x40) : (c))
#endif

```

```

#define XOR(e, f) (((e) && !(f)) || (!(e) && (f)))
int
main ()
{
    int i;
    for (i = 0; i < 256; i++)
        if (XOR (islower (i), ISLOWER (i))
            || toupper (i) != TOUPPER (i))
            return 2;
    return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :

else
    ac_cv_header_stdc=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$sac_exeext \
    conftest.$sac_objext conftest.beam conftest.$sac_ext
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_header_stdc"
>&5
$as_echo "$ac_cv_header_stdc" >&6; }
if test $ac_cv_header_stdc = yes; then

$as_echo "#define STDC_HEADERS 1" >>confdefs.h

fi

# On IRIX 5.3, sys/types and inttypes.h are conflicting.
for ac_header in sys/types.h sys/stat.h stdlib.h string.h memory.h
strings.h \
    inttypes.h stdint.h unistd.h
do :
    as_ac_Header=`$as_echo "ac_cv_header_$ac_header" | $as_tr_sh`
    ac_fn_c_check_header_compile "$LINENO" "$ac_header" "$as_ac_Header"
"$ac_includes_default"
"
    if eval test `x`"$as_ac_Header" = x"yes"; then :
        cat >>confdefs.h <<_ACEOF
#define ` $as_echo "HAVE_$ac_header" | $as_tr_cpp` 1
_ACEOF

    fi
done

```

```

    ac_fn_c_check_header_mongrel "$LINENO" "minix/config.h"
"ac_cv_header_minix_config_h" "$ac_includes_default"
if test "x$ac_cv_header_minix_config_h" = xyes; then :
    MINIX=yes
else
    MINIX=
fi

    if test "$MINIX" = yes; then

$as_echo "#define _POSIX_SOURCE 1" >>confdefs.h

$as_echo "#define _POSIX_1_SOURCE 2" >>confdefs.h

$as_echo "#define _MINIX 1" >>confdefs.h

    fi

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether it is safe
to define __EXTENSIONS__" >&5
$as_echo_n "checking whether it is safe to define __EXTENSIONS__... "
>&6; }
if ${ac_cv_safe_to_define__extensions__+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

#       define __EXTENSIONS__ 1
       $ac_includes_default

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_safe_to_define__extensions__=yes
else
    ac_cv_safe_to_define__extensions__=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_safe_to_define__extensions__" >&5
$as_echo "$ac_cv_safe_to_define__extensions__" >&6; }

```



```

test $ac_cv_safe_to_define__extensions__ = yes &&
  $as_echo "#define __EXTENSIONS__ 1" >>confdefs.h

$as_echo "#define _ALL_SOURCE 1" >>confdefs.h

$as_echo "#define _GNU_SOURCE 1" >>confdefs.h

$as_echo "#define _POSIX_PTHREAD_SEMANTICS 1" >>confdefs.h

$as_echo "#define _TANDEM_SOURCE 1" >>confdefs.h

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for library
containing strerror" >&5
$as_echo_n "checking for library containing strerror... " >&6; }
if ${ac_cv_search_strerror+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_func_search_save_LIBS=$LIBS
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char strerror ();
int
main ()
{
return strerror ();
;
return 0;
}
_ACEOF
for ac_lib in ' cposix; do
  if test -z "$ac_lib"; then
    ac_res="none required"
  else
    ac_res=-l$ac_lib
    LIBS="-l$ac_lib $ac_func_search_save_LIBS"
  fi
  if ac_fn_c_try_link "$LINENO"; then :
ac_cv_search_strerror=$ac_res
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext
  if ${ac_cv_search_strerror+:} false; then :
break
fi

```

```

done
if ${ac_cv_search_strerror+:} false; then :

else
    ac_cv_search_strerror=no
fi
rm conftest.$ac_ext
LIBS=$ac_func_search_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_search_strerror" >&5
$as_echo "$ac_cv_search_strerror" >&6; }
ac_res=$ac_cv_search_strerror
if test "$ac_res" != no; then :
    test "$ac_res" = "none required" || LIBS="$ac_res $LIBS"

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for ANSI C header
files" >&5
$as_echo_n "checking for ANSI C header files... " >&6; }
if ${ac_cv_header_stdcl+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <float.h>

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_header_stdcl=yes
else
    ac_cv_header_stdcl=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

if test $ac_cv_header_stdcl = yes; then
    # SunOS 4.x string.h does not declare mem*, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <string.h>

```

```

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "memchr" >/dev/null 2>&1; then :

else
    ac_cv_header_stdcl=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdcl = yes; then
    # ISC 2.0.2 stdlib.h does not declare free, contrary to ANSI.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdlib.h>

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext") 2>&5 |
    $EGREP "free" >/dev/null 2>&1; then :

else
    ac_cv_header_stdcl=no
fi
rm -f conftest*

fi

if test $ac_cv_header_stdcl = yes; then
    # /bin/cc in Irix-4.0.5 gets non-ANSI ctype macros unless using -
ansi.
    if test "$cross_compiling" = yes; then :
    :
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <ctype.h>
#include <stdlib.h>
#if ((' ' & 0xFF) == 0x020)
# define ISLOWER(c) ('a' <= (c) && (c) <= 'z')
# define TOUPPER(c) (ISLOWER(c) ? 'A' + ((c) - 'a') : (c))
#else
# define ISLOWER(c) \
        (('a' <= (c) && (c) <= 'i') \
        || ('j' <= (c) && (c) <= 'r') \
        || ('s' <= (c) && (c) <= 'z'))
# define TOUPPER(c) (ISLOWER(c) ? ((c) | 0x40) : (c))
#endif

#define XOR(e, f) (((e) && !(f)) || (!(e) && (f)))
int
main ()

```

```

{
  int i;
  for (i = 0; i < 256; i++)
    if (XOR (islower (i), ISLOWER (i))
        || toupper (i) != TOUPPER (i))
      return 2;
  return 0;
}
__ACEOF
if ac_fn_c_try_run "$LINENO"; then :

else
  ac_cv_header_stdcl=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$ac_exeext \
  conftest.$ac_objext conftest.beam conftest.$ac_ext
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_header_stdcl"
>&5
$as_echo "$ac_cv_header_stdcl" >&6; }
if test $ac_cv_header_stdcl = yes; then

$as_echo "#define STDC_HEADERS 1" >>confdefs.h

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for inline" >&5
$as_echo_n "checking for inline... " >&6; }
if ${ac_cv_c_inline+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_cv_c_inline=no
for ac_kw in inline __inline__ inline; do
  cat confdefs.h - <<__ACEOF >>conftest.$ac_ext
/* end confdefs.h. */
#ifdef __cplusplus
typedef int foo_t;
static $ac_kw foo_t static_foo () {return 0; }
$ac_kw foo_t foo () {return 0; }
#endif

__ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_cv_c_inline=$ac_kw
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
test "$ac_cv_c_inline" != no && break
done

```

```

fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $ac_cv_c_inline" >&5
$sas_echo "$ac_cv_c_inline" >&6; }

case $ac_cv_c_inline in
  inline | yes) ;;
  *)
    case $ac_cv_c_inline in
      no) ac_val=;;
      *) ac_val=$ac_cv_c_inline;;
    esac
    cat >>confdefs.h <<_ACEOF
#ifndef __cplusplus
#define inline $ac_val
#endif
_ACEOF
    ;;
esac

case `pwd` in
  *\ * | *\ *)
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: WARNING: Libtool does not
cope well with whitespace in ``pwd``" >&5
$sas_echo "$sas_me: WARNING: Libtool does not cope well with whitespace
in ``pwd``" >&2;} ;;
esac

macro_version='2.4.2'
macro_revision='1.3337'

```

```
ltmain="$ac_aux_dir/ltmain.sh"
```

```
# Backslashify metacharacters that are still active within
# double-quoted strings.
sed_quote_subst='s/\([\"`$\\]\)/\\\1/g'
```

```
# Same as above, but do not quote variable references.
double_quote_subst='s/\([\"`\\]\)/\\\1/g'
```



```

do
    cat "confptest.in" "confptest.in" >"confptest.tmp"
    mv "confptest.tmp" "confptest.in"
    cp "confptest.in" "confptest.nl"
    $as_echo ' ' >> "confptest.nl"
    "$ac_path_SED" -f confptest.sed < "confptest.nl" >"confptest.out"
2>/dev/null || break
    diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
    as_fn_arith $ac_count + 1 && ac_count=$as_val
    if test $ac_count -gt ${ac_path_SED_max-0}; then
        # Best one so far, save it but keep looking for a better one
        ac_cv_path_SED="$ac_path_SED"
        ac_path_SED_max=$ac_count
    fi
    # 10*(2^10) chars as input seems more than enough
    test $ac_count -gt 10 && break
done
rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
esac

    $ac_path_SED_found && break 3
done
done
done
IFS=$as_save_IFS
if test -z "$ac_cv_path_SED"; then
    as_fn_error $? "no acceptable sed could be found in \$PATH"
"$LINENO" 5
fi
else
    ac_cv_path_SED=$SED
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_SED" >&5
$as_echo "$ac_cv_path_SED" >&6; }
SED="$ac_cv_path_SED"
rm -f confptest.sed

test -z "$SED" && SED=sed
Xsed="$SED -e 1s/^X//"

```



```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for fgrep" >&5
$as_echo_n "checking for fgrep... " >&6; }
if ${ac_cv_path_FGREP+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if echo 'ab*c' | $GREP -F 'ab*c' >/dev/null 2>&1
  then ac_cv_path_FGREP="$GREP -F"
  else
    if test -z "$FGREP"; then
ac_path_FGREP_found=false
# Loop through the user's path and test for each of PROGNAMES_PATH
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH$PATH_SEPARATOR/usr/xpg4/bin
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_prog in fgrep; do
    for ac_exec_ext in '' $ac_executable_extensions; do
      ac_path_FGREP="$as_dir/$ac_prog$ac_exec_ext"
      as_fn_executable_p "$ac_path_FGREP" || continue
# Check for GNU ac_path_FGREP and select it if it is found.
# Check for GNU $ac_path_FGREP
case `"$ac_path_FGREP" --version 2>&1` in
*GNU*)
  ac_cv_path_FGREP="$ac_path_FGREP" ac_path_FGREP_found=:;;
*)
  ac_count=0
  $as_echo_n 0123456789 >"confptest.in"
  while :
  do
    cat "confptest.in" "confptest.in" >"confptest.tmp"
    mv "confptest.tmp" "confptest.in"
    cp "confptest.in" "confptest.nl"
    $as_echo 'FGREP' >> "confptest.nl"
    "$ac_path_FGREP" FGREP < "confptest.nl" >"confptest.out" 2>/dev/null
  || break
  diff "confptest.out" "confptest.nl" >/dev/null 2>&1 || break
  as_fn_arith $ac_count + 1 && ac_count=$as_val
  if test $ac_count -gt ${ac_path_FGREP_max-0}; then
    # Best one so far, save it but keep looking for a better one
    ac_cv_path_FGREP="$ac_path_FGREP"
    ac_path_FGREP_max=$ac_count
  fi
  # 10*(2^10) chars as input seems more than enough
  test $ac_count -gt 10 && break
done
  rm -f confptest.in confptest.tmp confptest.nl confptest.out;;
esac

  $ac_path_FGREP_found && break 3
done
done

```

```

done
IFS=$as_save_IFS
if test -z "$ac_cv_path_FGREP"; then
  as_fn_error $? "no acceptable fgrep could be found in
$PATH$PATH_SEPARATOR/usr/xpg4/bin" "$LINENO" 5
fi
else
  ac_cv_path_FGREP=$FGREP
fi

fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_path_FGREP"
>&5
$as_echo "$ac_cv_path_FGREP" >&6; }
FGREP="$ac_cv_path_FGREP"

```

```

test -z "$GREP" && GREP=grep

```

```

# Check whether --with-gnu-ld was given.
if test "${with_gnu_ld+set}" = set; then :
  withval=$with_gnu_ld; test "$withval" = no || with_gnu_ld=yes
else
  with_gnu_ld=no
fi

ac_prog=ld
if test "$GCC" = yes; then
  # Check if gcc -print-prog-name=ld gives a path.
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for ld used by
$CC" >&5
$as_echo_n "checking for ld used by $CC... " >&6; }
  case $host in

```

```

*-*-mingw*)
  # gcc leaves a trailing carriage return which upsets mingw
  ac_prog=`($CC -print-prog-name=ld) 2>&5 | tr -d '\015'` ;;
*)
  ac_prog=`($CC -print-prog-name=ld) 2>&5` ;;
esac
case $ac_prog in
  # Accept absolute paths.
  [\\/] * | ?:[\\/] *)
    re_direlt='/[^/][^/]*/\.\./'
    # Canonicalize the pathname of ld
    ac_prog=`$ECHO "$ac_prog"| $SED 's%\\\\\%/g'`
    while $ECHO "$ac_prog" | $GREP "$re_direlt" > /dev/null 2>&1; do
      ac_prog=`$ECHO $ac_prog| $SED "s%$re_direlt%/"`
    done
    test -z "$LD" && LD="$ac_prog"
    ;;
  "")
    # If it fails, then pretend we aren't using GCC.
    ac_prog=ld
    ;;
  *)
    # If it is relative, then search for the first ld in PATH.
    with_gnu_ld=unknown
    ;;
esac
elif test "$with_gnu_ld" = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for GNU ld" >&5
$as_echo_n "checking for GNU ld... " >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for non-GNU ld"
>&5
$as_echo_n "checking for non-GNU ld... " >&6; }
fi
if ${lt_cv_path_LD+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -z "$LD"; then
    lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
    for ac_dir in $PATH; do
      IFS="$lt_save_ifs"
      test -z "$ac_dir" && ac_dir=.
      if test -f "$ac_dir/$ac_prog" || test -f
"$ac_dir/$ac_prog$ac_exeext"; then
        lt_cv_path_LD="$ac_dir/$ac_prog"
        # Check to see if the program is GNU ld.  I'd rather use --
version,
        # but apparently some variants of GNU ld only accept -v.
        # Break only if it was the GNU/non-GNU ld that we prefer.
        case `"$lt_cv_path_LD" -v 2>&1 </dev/null` in
          *GNU* | *'with BFD'*)
            test "$with_gnu_ld" != no && break

```

```

        ;;
        *)
        test "$with_gnu_ld" != yes && break
        ;;
    esac
fi
done
IFS="$lt_save_ifs"
else
    lt_cv_path_LD="$LD" # Let the user override the test with a path.
fi
fi

LD="$lt_cv_path_LD"
if test -n "$LD"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LD" >&5
$as_echo "$LD" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi
test -z "$LD" && as_fn_error $? "no acceptable ld found in \$PATH"
"$LINENO" 5
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if the linker ($LD)
is GNU ld" >&5
$as_echo_n "checking if the linker ($LD) is GNU ld... " >&6; }
if ${lt_cv_prog_gnu_ld+:} false; then :
    $as_echo_n "(cached) " >&6
else
    # I'd rather use --version here, but apparently some GNU lds only
    accept -v.
    case ` $LD -v 2>&1 </dev/null` in
    *GNU* | *'with BFD'*)
        lt_cv_prog_gnu_ld=yes
        ;;
    *)
        lt_cv_prog_gnu_ld=no
        ;;
    esac
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_prog_gnu_ld"
>&5
$as_echo "$lt_cv_prog_gnu_ld" >&6; }
with_gnu_ld=$lt_cv_prog_gnu_ld

```

```

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for BSD- or MS-
compatible name lister (nm)" >&5
$sas_echo_n "checking for BSD- or MS-compatible name lister (nm)... "
>&6; }
if ${lt_cv_path_NM+:} false; then :
  $sas_echo_n "(cached) " >&6
else
  if test -n "$NM"; then
    # Let the user override the test.
    lt_cv_path_NM="$NM"
  else
    lt_nm_to_check="${ac_tool_prefix}nm"
    if test -n "$ac_tool_prefix" && test "$build" = "$host"; then
      lt_nm_to_check="$lt_nm_to_check nm"
    fi
    for lt_tmp_nm in $lt_nm_to_check; do
      lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
      for ac_dir in $PATH /usr/ccs/bin/elf /usr/ccs/bin /usr/ucb /bin;
do
      IFS="$lt_save_ifs"
      test -z "$ac_dir" && ac_dir=.
      tmp_nm="$ac_dir/$lt_tmp_nm"
      if test -f "$tmp_nm" || test -f "$tmp_nm$ac_exeext" ; then
        # Check to see if the nm accepts a BSD-compat flag.
        # Adding the `sed 1q' prevents false positives on HP-UX, which
says:
        # nm: unknown option "B" ignored
        # Tru64's nm complains that /dev/null is an invalid object file
        case `"$tmp_nm" -B /dev/null 2>&1 | sed '1q'` in
        */dev/null* | *'Invalid file or object type'*)
          lt_cv_path_NM="$tmp_nm -B"
          break
          ;;
        *)
          case `"$tmp_nm" -p /dev/null 2>&1 | sed '1q'` in
          */dev/null*)
            lt_cv_path_NM="$tmp_nm -p"
            break
            ;;
          *)
            lt_cv_path_NM=${lt_cv_path_NM="$tmp_nm"} # keep the first
match, but
            continue # so that we can try to find one that supports BSD
flags
            ;;
          esac
          ;;
        esac
      fi
    done
    IFS="$lt_save_ifs"

```

```

done
: ${lt_cv_path_NM=no}
fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_path_NM" >&5
$as_echo "$lt_cv_path_NM" >&6; }
if test "$lt_cv_path_NM" != "no"; then
  NM="$lt_cv_path_NM"
else
  # Didn't find any BSD compatible name lister, look for dumpbin.
  if test -n "$DUMPBIN"; then :
    # Let the user override the test.
  else
    if test -n "$ac_tool_prefix"; then
      for ac_prog in dumpbin "link -dump"
      do
        # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
        be a program name with args.
        set dummy $ac_tool_prefix$ac_prog; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_prog_DUMPBIN+:} false; then :
          $as_echo_n "(cached) " >&6
        else
          if test -n "$DUMPBIN"; then
            ac_cv_prog_DUMPBIN="$DUMPBIN" # Let the user override the test.
          else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
              IFS=$as_save_IFS
              test -z "$as_dir" && as_dir=.
              for ac_exec_ext in ' $ac_executable_extensions; do
                if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                  ac_cv_prog_DUMPBIN="$ac_tool_prefix$ac_prog"
                  $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                  break 2
                fi
              done
            done
            IFS=$as_save_IFS
          fi
        fi
      done
    fi
  fi
  DUMPBIN=$ac_cv_prog_DUMPBIN
  if test -n "$DUMPBIN"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DUMPBIN" >&5
$as_echo "$DUMPBIN" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  fi
fi

```

```

fi

    test -n "$DUMPBIN" && break
done
fi
if test -z "$DUMPBIN"; then
    ac_ct_DUMPBIN=$DUMPBIN
    for ac_prog in dumpbin "link -dump"
do
    # Extract the first word of "$ac_prog", so it can be a program name
    with args.
    set dummy $ac_prog; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_DUMPBIN+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_DUMPBIN"; then
            ac_cv_prog_ac_ct_DUMPBIN="$ac_ct_DUMPBIN" # Let the user override
            the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '$ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_ac_ct_DUMPBIN="$ac_prog"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

        fi
        fi
        ac_ct_DUMPBIN=$ac_cv_prog_ac_ct_DUMPBIN
        if test -n "$ac_ct_DUMPBIN"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DUMPBIN" >&5
$as_echo "$ac_ct_DUMPBIN" >&6; }
        else
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        fi
    fi

    test -n "$ac_ct_DUMPBIN" && break
done

```

```

if test "x$ac_ct_DUMPBIN" = x; then
  DUMPBIN=":"
else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  DUMPBIN=$ac_ct_DUMPBIN
fi
fi

  case ` $DUMPBIN -symbols /dev/null 2>&1 | sed '1q'` in
*COFF*)
  DUMPBIN="$DUMPBIN -symbols"
  ;;
*)
  DUMPBIN=:
  ;;
esac
fi

if test "$DUMPBIN" != ":"; then
  NM="$DUMPBIN"
fi
fi
test -z "$NM" && NM=nm

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking the name lister
($NM) interface" >&5
$as_echo_n "checking the name lister ($NM) interface... " >&6; }
if ${lt_cv_nm_interface+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_nm_interface="BSD nm"
  echo "int some_variable = 0;" > conftest.$ac_ext
  (eval echo "\"$as_me:$LINENO: $ac_compile\"" >&5)
  (eval "$ac_compile" 2>conftest.err)
  cat conftest.err >&5
  (eval echo "\"$as_me:$LINENO: $NM \\\"conftest.$ac_objext\\\"\""
>&5)
  (eval "$NM \"conftest.$ac_objext\" 2>conftest.err > conftest.out)
  cat conftest.err >&5

```



```

    (eval echo "\"\$as_me:$LINENO: output\" " >&5)
    cat confptest.out >&5
    if $GREP 'External.*some_variable' confptest.out > /dev/null; then
        lt_cv_nm_interface="MS dumpbin"
    fi
    rm -f confptest*
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_nm_interface"
>&5
$as_echo "$lt_cv_nm_interface" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether ln -s works"
>&5
$as_echo_n "checking whether ln -s works... " >&6; }
LN_S=$as_ln_s
if test "$LN_S" = "ln -s"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no, using $LN_S"
>&5
$as_echo "no, using $LN_S" >&6; }
fi

# find the maximum length of command line arguments
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking the maximum length
of command line arguments" >&5
$as_echo_n "checking the maximum length of command line arguments... "
>&6; }
if ${lt_cv_sys_max_cmd_len+:} false; then :
    $as_echo_n "(cached) " >&6
else
    i=0
    teststring="ABCD"

    case $build_os in
    msdosdjgpp*)
        # On DJGPP, this test can blow up pretty badly due to problems in
libc
        # (any single argument exceeding 2000 bytes causes a buffer
overrun
        # during glob expansion). Even if it were fixed, the result of
this
        # check would be larger than it should be.
        lt_cv_sys_max_cmd_len=12288;    # 12K is about right
        ;;

gnu*)
        # Under GNU Hurd, this test is not required because there is
        # no limit to the length of command line arguments.
        # Libtool will interpret -1 as no limit whatsoever
        lt_cv_sys_max_cmd_len=-1;

```

```

;;

cygwin* | mingw* | cegcc*)
# On Win9x/ME, this test blows up -- it succeeds, but takes
# about 5 minutes as the teststring grows exponentially.
# Worse, since 9x/ME are not pre-emptively multitasking,
# you end up with a "frozen" computer, even though with patience
# the test eventually succeeds (with a max line length of 256k).
# Instead, let's just punt: use the minimum linelength reported by
# all of the supported platforms: 8192 (on NT/2K/XP).
lt_cv_sys_max_cmd_len=8192;
;;

mint*)
# On MiNT this can take a long time and run out of memory.
lt_cv_sys_max_cmd_len=8192;
;;

amigaos*)
# On AmigaOS with pdksh, this test takes hours, literally.
# So we just punt and use a minimum line length of 8192.
lt_cv_sys_max_cmd_len=8192;
;;

netbsd* | freebsd* | openbsd* | darwin* | dragonfly*)
# This has been around since 386BSD, at least. Likely further.
if test -x /sbin/sysctl; then
  lt_cv_sys_max_cmd_len=`/sbin/sysctl -n kern.argmax`
elif test -x /usr/sbin/sysctl; then
  lt_cv_sys_max_cmd_len=`/usr/sbin/sysctl -n kern.argmax`
else
  lt_cv_sys_max_cmd_len=65536      # usable default for all BSDs
fi
# And add a safety zone
lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
;;

interix*)
# We know the value 262144 and hardcode it with a safety zone
# (like BSD)
lt_cv_sys_max_cmd_len=196608
;;

os2*)
# The test takes a long time on OS/2.
lt_cv_sys_max_cmd_len=8192
;;

osf*)
# Dr. Hans Ekkehard Plesser reports seeing a kernel panic running
configure

```

```

# due to this test when exec_disable_arg_limit is 1 on Tru64. It
is not
# nice to cause kernel panics so lets avoid the loop below.
# First set a reasonable default.
lt_cv_sys_max_cmd_len=16384
#
if test -x /sbin/sysconfig; then
  case `/sbin/sysconfig -q proc exec_disable_arg_limit` in
    *1*) lt_cv_sys_max_cmd_len=-1 ;;
  esac
fi
;;
sco3.2v5*)
  lt_cv_sys_max_cmd_len=102400
  ;;
sysv5* | sco5v6* | sysv4.2uw2*)
  kargmax=`grep ARG_MAX /etc/conf/cf.d/stune 2>/dev/null`
  if test -n "$kargmax"; then
    lt_cv_sys_max_cmd_len=`echo $kargmax | sed 's/.*[ ]//`
  else
    lt_cv_sys_max_cmd_len=32768
  fi
  ;;
*)
  lt_cv_sys_max_cmd_len=`(getconf ARG_MAX) 2> /dev/null`
  if test -n "$lt_cv_sys_max_cmd_len"; then
    lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
    lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
  else
    # Make teststring a little bigger before we do anything with it.
    # a 1K string should be a reasonable start.
    for i in 1 2 3 4 5 6 7 8 ; do
      teststring=$teststring$teststring
    done
    SHELL=${SHELL-${CONFIG_SHELL-/bin/sh}}
    # If test is not a shell built-in, we'll probably end up
computing a
# maximum length that is only half of the actual maximum length,
but
# we can't tell.
while { test "X"$teststring`env echo "$teststring$teststring" 2>/dev/null`
\
      = "X$teststring$teststring"; } >/dev/null 2>&1 &&
  test $i != 17 # 1/2 MB should be enough
do
  i=`expr $i + 1`
  teststring=$teststring$teststring
done
# Only check the string length outside the loop.
lt_cv_sys_max_cmd_len=`expr "X$teststring" : ".*" 2>&1`
teststring=

```

```

        # Add a significant safety factor because C++ compilers can tack
on
        # massive amounts of additional arguments before passing them to
the
        # linker. It appears as though 1/2 is a usable value.
        lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 2`
    fi
    ;;
esac

```

```
fi
```

```

if test -n $lt_cv_sys_max_cmd_len ; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_sys_max_cmd_len" >&5
$as_echo "$lt_cv_sys_max_cmd_len" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: none" >&5
$as_echo "none" >&6; }
fi
max_cmd_len=$lt_cv_sys_max_cmd_len

```

```

: ${CP="cp -f"}
: ${MV="mv -f"}
: ${RM="rm -f"}

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the shell
understands some XSI constructs" >&5
$as_echo_n "checking whether the shell understands some XSI
constructs... " >&6; }
# Try some XSI features
xsi_shell=no
( _lt_dummy="a/b/c"
  test
"$_{_lt_dummy##*/},${_lt_dummy%/*},${_lt_dummy#??}"${_lt_dummy%$_lt_du
mmy"}, \
    = c,a/b,b/c, \
    && eval 'test $(( 1 + 1 )) -eq 2 \
    && test "#_lt_dummy" -eq 5' ) >/dev/null 2>&1 \
    && xsi_shell=yes
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $xsi_shell" >&5
$as_echo "$xsi_shell" >&6; }

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the shell
understands \"+=\\" >&5
$as_echo_n "checking whether the shell understands \"+=\\"... " >&6; }

```

```

lt_shell_append=no
( foo=bar; set foo baz; eval "$1+=\$2" && test "$foo" = barbaz ) \
  >/dev/null 2>&1 \
  && lt_shell_append=yes
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_shell_append" >&5
$as_echo "$lt_shell_append" >&6; }

```

```

if ( (MAIL=60; unset MAIL) || exit) >/dev/null 2>&1; then
  lt_unset=unset
else
  lt_unset=false
fi

```

```

# test EBCDIC or ASCII
case `echo X|tr X '\101'` in
  A) # ASCII based system
    # \n is not interpreted correctly by Solaris 8 /usr/ucb/tr
    lt_SP2NL='tr \040 \012'
    lt_NL2SP='tr \015\012 \040\040'
    ;;
  *) # EBCDIC based system
    lt_SP2NL='tr \100 \n'
    lt_NL2SP='tr \r\n \100\100'
    ;;
esac

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to convert
$build file names to $host format" >&5
$as_echo_n "checking how to convert $build file names to $host
format... " >&6; }
if ${lt_cv_to_host_file_cmd+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $host in
    *-*-mingw* )
      case $build in
        *-*-mingw* ) # actually msys
          lt_cv_to_host_file_cmd=func_convert_file_msys_to_w32
          ;;

```

```

    *-*-cygwin* )
        lt_cv_to_host_file_cmd=func_convert_file_cygwin_to_w32
        ;;
    * ) # otherwise, assume *nix
        lt_cv_to_host_file_cmd=func_convert_file_nix_to_w32
        ;;
esac
;;
*-*-cygwin* )
case $build in
    *-*-mingw* ) # actually msys
        lt_cv_to_host_file_cmd=func_convert_file_msys_to_cygwin
        ;;
    *-*-cygwin* )
        lt_cv_to_host_file_cmd=func_convert_file_noop
        ;;
    * ) # otherwise, assume *nix
        lt_cv_to_host_file_cmd=func_convert_file_nix_to_cygwin
        ;;
esac
;;
* ) # unhandled hosts (and "normal" native builds)
    lt_cv_to_host_file_cmd=func_convert_file_noop
    ;;
esac

fi

to_host_file_cmd=$lt_cv_to_host_file_cmd
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_to_host_file_cmd" >&5
$as_echo "$lt_cv_to_host_file_cmd" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to convert
$build file names to toolchain format" >&5
$as_echo_n "checking how to convert $build file names to toolchain
format... " >&6; }
if ${lt_cv_to_tool_file_cmd+:} false; then :
    $as_echo_n "(cached) " >&6
else
    #assume ordinary cross tools, or native build.
    lt_cv_to_tool_file_cmd=func_convert_file_noop
    case $host in
        *-*-mingw* )
            case $build in
                *-*-mingw* ) # actually msys
                    lt_cv_to_tool_file_cmd=func_convert_file_msys_to_w32
                    ;;
            esac
        ;;
    esac

```

```

        esac
        ;;
    esac

fi

to_tool_file_cmd=${lt_cv_to_tool_file_cmd}
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_to_tool_file_cmd" >&5
$as_echo "$lt_cv_to_tool_file_cmd" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $LD option to
reload object files" >&5
$as_echo_n "checking for $LD option to reload object files... " >&6; }
if ${lt_cv_ld_reload_flag+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_ld_reload_flag='-r'
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_ld_reload_flag" >&5
$as_echo "$lt_cv_ld_reload_flag" >&6; }
reload_flag=${lt_cv_ld_reload_flag}
case $reload_flag in
"" | " *") ;;
*) reload_flag=" $reload_flag" ;;
esac
reload_cmds='$LD$reload_flag -o $output$reload_objs'
case $host_os in
  cygwin* | mingw* | pw32* | cegcc*)
    if test "$GCC" != yes; then
      reload_cmds=false
    fi
    ;;
  darwin*)
    if test "$GCC" = yes; then
      reload_cmds='$LTCC $LTCFLAGS -nostdlib ${wl}-r -o
$output$reload_objs'
    else
      reload_cmds='$LD$reload_flag -o $output$reload_objs'
    fi
    ;;
esac

```

```

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}objdump", so it can be
  a program name with args.
  set dummy ${ac_tool_prefix}objdump; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_OBJDUMP+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$OBJDUMP"; then
      ac_cv_prog_OBJDUMP="$OBJDUMP" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_OBJDUMP="${ac_tool_prefix}objdump"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      OBJDUMP=$ac_cv_prog_OBJDUMP
      if test -n "$OBJDUMP"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OBJDUMP" >&5
        $as_echo "$OBJDUMP" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi

      fi

      if test -z "$ac_cv_prog_OBJDUMP"; then
        ac_ct_OBJDUMP=$OBJDUMP
        # Extract the first word of "objdump", so it can be a program name
        with args.
        set dummy objdump; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
        $as_echo_n "checking for $ac_word... " >&6; }

```



```

if ${ac_cv_prog_ac_ct_OBJDUMP+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_OBJDUMP"; then
    ac_cv_prog_ac_ct_OBJDUMP="$ac_ct_OBJDUMP" # Let the user override
the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_ac_ct_OBJDUMP="objdump"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
ac_ct_OBJDUMP=$ac_cv_prog_ac_ct_OBJDUMP
if test -n "$ac_ct_OBJDUMP"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OBJDUMP" >&5
$as_echo "$ac_ct_OBJDUMP" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_OBJDUMP" = x; then
    OBJDUMP="false"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    OBJDUMP=$ac_ct_OBJDUMP
  fi
else
  OBJDUMP="$ac_cv_prog_OBJDUMP"
fi

test -z "$OBJDUMP" && OBJDUMP=objdump

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to recognize
dependent libraries" >&5
$as_echo_n "checking how to recognize dependent libraries... " >&6; }
if ${lt_cv_deplibs_check_method+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_file_magic_cmd='$MAGIC_CMD'
lt_cv_file_magic_test_file=
lt_cv_deplibs_check_method='unknown'
# Need to set the preceding variable on all platforms that support
# interlibrary dependencies.
# 'none' -- dependencies not supported.
# `unknown' -- same as none, but documents that we really don't know.
# 'pass_all' -- all dependencies passed with no checks.
# 'test_compile' -- check by making test program.
# 'file_magic [[regex]]' -- check by looking for files in library path
# which responds to the $file_magic_cmd with a given extended regex.
# If you have `file' or equivalent on your system and you're not sure
# whether `pass_all' will *always* work, you probably want this one.

case $host_os in
aix[4-9]*)
  lt_cv_deplibs_check_method=pass_all
  ;;

beos*)
  lt_cv_deplibs_check_method=pass_all
  ;;

bsdi[45]*)
  lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(shared object|dynamic lib)'
  lt_cv_file_magic_cmd='/usr/bin/file -L'
  lt_cv_file_magic_test_file=/shlib/libc.so
  ;;

cygwin*)
  # func_win32_libid is a shell function defined in ltmain.sh
  lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86 DLL'
  lt_cv_file_magic_cmd='func_win32_libid'
  ;;

mingw* | pw32*)

```

```

# Base MSYS/MinGW do not provide the 'file' command needed by
# func_win32_libid shell function, so use a weaker test based on
'objdump',
# unless we find 'file', for example because we are cross-compiling.
# func_win32_libid assumes BSD nm, so disallow it if using MS
dumpbin.
if ( test "$lt_cv_nm_interface" = "BSD nm" && file / ) >/dev/null
2>&1; then
    lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86
DLL'
    lt_cv_file_magic_cmd='func_win32_libid'
else
    # Keep this pattern in sync with the one in func_win32_libid.
    lt_cv_deplibs_check_method='file_magic file format (pei*-
i386(.*architecture: i386)?|pe-arm-wince|pe-x86-64)'
    lt_cv_file_magic_cmd='$OBJDUMP -f'
fi
;;

cegcc*)
# use the weaker test based on 'objdump'. See mingw*.
lt_cv_deplibs_check_method='file_magic file format pe-arm-
.*little(.*architecture: arm)?'
lt_cv_file_magic_cmd='$OBJDUMP -f'
;;

darwin* | rhapsody*)
lt_cv_deplibs_check_method=pass_all
;;

freebsd* | dragonfly*)
if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
    case $host_cpu in
        i*86 )
            # Not sure whether the presence of OpenBSD here was a mistake.
            # Let's accept both of them until this is cleared up.
            lt_cv_deplibs_check_method='file_magic
(FreeBSD|OpenBSD|DragonFly)/i[3-9]86 (compact )?demand paged shared
library'
            lt_cv_file_magic_cmd=/usr/bin/file
            lt_cv_file_magic_test_file=`echo /usr/lib/libc.so.*`
            ;;
        esac
    else
        lt_cv_deplibs_check_method=pass_all
    fi
    ;;

gnu*)
lt_cv_deplibs_check_method=pass_all
;;

```

```

haiku*)
    lt_cv_deplibs_check_method=pass_all
    ;;

hpux10.20* | hpux11*)
    lt_cv_file_magic_cmd=/usr/bin/file
    case $host_cpu in
    ia64*)
        lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|ELF-[0-9][0-9]) shared object file - IA64'
        lt_cv_file_magic_test_file=/usr/lib/hpux32/libc.so
        ;;
    hppa*64*)
        lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|ELF[ -][0-9][0-9])(-bit)?( [LM]SB)? shared object( file)?[, -]* PA-RISC [0-9]\.[0-9]'
        lt_cv_file_magic_test_file=/usr/lib/pa20_64/libc.sl
        ;;
    *)
        lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|PA-RISC[0-9]\.[0-9]) shared library'
        lt_cv_file_magic_test_file=/usr/lib/libc.sl
        ;;
    esac
    ;;

interix[3-9]*)
    # PIC code is broken on Interix 3.x, that's why |\a not |\_pic\.a
    here
    lt_cv_deplibs_check_method='match_pattern /lib[^\.]+(\.so|\a)$'
    ;;

irix5* | irix6* | nonstopux*)
    case $LD in
    *-32|*" -32 ") libmagic=32-bit;;
    *-n32|*" -n32 ") libmagic=N32;;
    *-64|*" -64 ") libmagic=64-bit;;
    *) libmagic=never-match;;
    esac
    lt_cv_deplibs_check_method=pass_all
    ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu)
    lt_cv_deplibs_check_method=pass_all
    ;;

netbsd*)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
        lt_cv_deplibs_check_method='match_pattern /lib[^\.]+(\.so\.[0-9]+\.[0-9]+|\_pic\a)$'
    else

```

```

    lt_cv_deplibs_check_method='match_pattern
/lib[^/]+(\.so|_pic\.a)$'
    fi
    ;;

newos6*)
    lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(executable|dynamic lib)'
    lt_cv_file_magic_cmd=/usr/bin/file
    lt_cv_file_magic_test_file=/usr/lib/libnls.so
    ;;

*nto* | *qnx*)
    lt_cv_deplibs_check_method=pass_all
    ;;

openbsd*)
    if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
        lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so\.[0-
9]+\.[0-9]+|_pic\.a)$'
    else
        lt_cv_deplibs_check_method='match_pattern /lib[^/]+(\.so\.[0-
9]+\.[0-9]+|_pic\.a)$'
    fi
    ;;

osf3* | osf4* | osf5*)
    lt_cv_deplibs_check_method=pass_all
    ;;

rdos*)
    lt_cv_deplibs_check_method=pass_all
    ;;

solaris*)
    lt_cv_deplibs_check_method=pass_all
    ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
    lt_cv_deplibs_check_method=pass_all
    ;;

sysv4 | sysv4.3*)
    case $host_vendor in
    motorola)
        lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [ML]SB
(shared object|dynamic lib) M[0-9][0-9]* Version [0-9]'
        lt_cv_file_magic_test_file=`echo /usr/lib/libc.so*`
        ;;
    ncr)
        lt_cv_deplibs_check_method=pass_all
    ;;

```

```

    ;;
sequent)
    lt_cv_file_magic_cmd='/bin/file'
    lt_cv_deplibs_check_method='file_magic ELF [0-9][0-9]*-bit [LM]SB
(shared object|dynamic lib )'
    ;;
sni)
    lt_cv_file_magic_cmd='/bin/file'
    lt_cv_deplibs_check_method="file_magic ELF [0-9][0-9]*-bit [LM]SB
dynamic lib"
    lt_cv_file_magic_test_file=/lib/libc.so
    ;;
siemens)
    lt_cv_deplibs_check_method=pass_all
    ;;
pc)
    lt_cv_deplibs_check_method=pass_all
    ;;
esac
;;

tpf*)
    lt_cv_deplibs_check_method=pass_all
    ;;
esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_deplibs_check_method" >&5
$as_echo "$lt_cv_deplibs_check_method" >&6; }

file_magic_glob=
want_nocaseglob=no
if test "$build" = "$host"; then
  case $host_os in
mingw* | pw32*)
    if ( shopt | grep nocaseglob ) >/dev/null 2>&1; then
      want_nocaseglob=yes
    else
      file_magic_glob=`echo
aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ | $SED -e
"s/\(..\)/s\/[\1]\\/[\1]\\/g;/g"`
    fi
    ;;
  esac
fi

file_magic_cmd=$lt_cv_file_magic_cmd
deplibs_check_method=$lt_cv_deplibs_check_method
test -z "$deplibs_check_method" && deplibs_check_method=unknown

```

```

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}dlltool", so it can be
  a program name with args.
  set dummy ${ac_tool_prefix}dlltool; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_DLLTOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$DLLTOOL"; then
      ac_cv_prog_DLLTOOL="$DLLTOOL" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_DLLTOOL="${ac_tool_prefix}dlltool"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
            $as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      DLLTOOL=$ac_cv_prog_DLLTOOL
      if test -n "$DLLTOOL"; then

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DLLTOOL" >&5
$as_echo "$DLLTOOL" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi

if test -z "$ac_cv_prog_DLLTOOL"; then
    ac_ct_DLLTOOL=$DLLTOOL
    # Extract the first word of "dlltool", so it can be a program name
    with args.
    set dummy dlltool; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_DLLTOOL+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_DLLTOOL"; then
            ac_cv_prog_ac_ct_DLLTOOL="$ac_ct_DLLTOOL" # Let the user override
            the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in ' $ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_ac_ct_DLLTOOL="dlltool"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

            fi
            fi
            ac_ct_DLLTOOL=$ac_cv_prog_ac_ct_DLLTOOL
            if test -n "$ac_ct_DLLTOOL"; then
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DLLTOOL" >&5
$as_echo "$ac_ct_DLLTOOL" >&6; }
            else
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
            fi

            if test "x$ac_ct_DLLTOOL" = x; then
                DLLTOOL="false"

```



```

else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  DLLTOOL=$ac_ct_DLLTOOL
  fi
else
  DLLTOOL="$ac_cv_prog_DLLTOOL"
fi

test -z "$DLLTOOL" && DLLTOOL=dlltool

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking how to associate
runtime and link libraries" >&5
$as_echo_n "checking how to associate runtime and link libraries... "
>&6; }
if ${lt_cv_sharedlib_from_linklib_cmd+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_sharedlib_from_linklib_cmd='unknown'

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
  # two different shell functions defined in ltmain.sh
  # decide which to use based on capabilities of $DLLTOOL
  case ` $DLLTOOL --help 2>&1 ` in
*--identify-strict*)
  lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib
  ;;
*)
lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib_fallback
  ;;
esac
  ;;
*)
  # fallback: assume linklib IS sharedlib
  lt_cv_sharedlib_from_linklib_cmd="$ECHO"

```

```

    ;;
esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_sharedlib_from_linklib_cmd" >&5
$as_echo "$lt_cv_sharedlib_from_linklib_cmd" >&6; }
sharedlib_from_linklib_cmd=$lt_cv_sharedlib_from_linklib_cmd
test -z "$sharedlib_from_linklib_cmd" &&
sharedlib_from_linklib_cmd=$ECHO

if test -n "$ac_tool_prefix"; then
  for ac_prog in ar
  do
    # Extract the first word of "$ac_tool_prefix$ac_prog", so it can
    be a program name with args.
    set dummy $ac_tool_prefix$ac_prog; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_AR+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      if test -n "$AR"; then
        ac_cv_prog_AR="$AR" # Let the user override the test.
      else
        as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
        for as_dir in $PATH
        do
          IFS=$as_save_IFS
          test -z "$as_dir" && as_dir=.
          for ac_exec_ext in '' $ac_executable_extensions; do
            if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
              ac_cv_prog_AR="$ac_tool_prefix$ac_prog"
              $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
              break 2
            fi
          done
        done
        IFS=$as_save_IFS
      fi
    fi
  done
  AR=$ac_cv_prog_AR
  if test -n "$AR"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $AR" >&5

```

```

$as_echo "$AR" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    test -n "$AR" && break
done
fi
if test -z "$AR"; then
  ac_ct_AR=$AR
  for ac_prog in ar
do
  # Extract the first word of "$ac_prog", so it can be a program name
with args.
set dummy $ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_AR+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_AR"; then
    ac_cv_prog_ac_ct_AR="$ac_ct_AR" # Let the user override the test.
else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_ac_ct_AR="$ac_prog"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_AR=$ac_cv_prog_ac_ct_AR
if test -n "$ac_ct_AR"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_AR" >&5
$as_echo "$ac_ct_AR" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

```

```

    test -n "$ac_ct_AR" && break
done

if test "x$ac_ct_AR" = x; then
    AR="false"
else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    AR=$ac_ct_AR
    fi
fi

: ${AR=ar}
: ${AR_FLAGS=cru}

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for archiver @FILE
support" >&5
$as_echo_n "checking for archiver @FILE support... " >&6; }
if ${lt_cv_ar_at_file+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_ar_at_file=no
    cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    echo conftest.$ac_objext > conftest.lst

```

```

        lt_ar_try='$AR $AR_FLAGS libconfptest.a @confptest.lst >&5'
        { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}:
\"$lt_ar_try\""; } >&5
        (eval $lt_ar_try) 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }
            if test "$ac_status" -eq 0; then
                # Ensure the archiver fails upon bogus file names.
                rm -f confptest.$ac_objext libconfptest.a
                { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}: \"$lt_ar_try\"";
} >&5
                (eval $lt_ar_try) 2>&5
                ac_status=$?
                $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
                test $ac_status = 0; }
                    if test "$ac_status" -ne 0; then
                        lt_cv_ar_at_file=@
                    fi
                fi
                rm -f confptest.* libconfptest.a

fi
rm -f core confptest.err confptest.$ac_objext confptest.$ac_ext

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_ar_at_file"
>&5
$as_echo "$lt_cv_ar_at_file" >&6; }

if test "x$lt_cv_ar_at_file" = xno; then
    archiver_list_spec=
else
    archiver_list_spec=$lt_cv_ar_at_file
fi

if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}strip", so it can be a
    program name with args.
    set dummy ${ac_tool_prefix}strip; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_STRIP+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$STRIP"; then

```

```

    ac_cv_prog_STRIP="$STRIP" # Let the user override the test.
else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_STRIP="{ac_tool_prefix}strip"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
STRIP=$ac_cv_prog_STRIP
if test -n "$STRIP"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $STRIP" >&5
$as_echo "$STRIP" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_STRIP"; then
    ac_ct_STRIP=$STRIP
    # Extract the first word of "strip", so it can be a program name
    with args.
    set dummy strip; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_STRIP+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_STRIP"; then
            ac_cv_prog_ac_ct_STRIP="$ac_ct_STRIP" # Let the user override the
            test.
        else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then

```

```

        ac_cv_prog_ac_ct_STRIP="strip"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
ac_ct_STRIP=$ac_cv_prog_ac_ct_STRIP
if test -n "$ac_ct_STRIP"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_STRIP" >&5
$as_echo "$ac_ct_STRIP" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    if test "x$ac_ct_STRIP" = x; then
        STRIP=":"
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        STRIP=$ac_ct_STRIP
    fi
else
    STRIP="$ac_cv_prog_STRIP"
fi

test -z "$STRIP" && STRIP=:

if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}ranlib", so it can be
    a program name with args.
    set dummy ${ac_tool_prefix}ranlib; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_RANLIB+:} false; then :
        $as_echo_n "(cached) " >&6

```

```

else
  if test -n "$RANLIB"; then
    ac_cv_prog_RANLIB="$RANLIB" # Let the user override the test.
  else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_prog_RANLIB="{ac_tool_prefix}ranlib"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

fi
fi
RANLIB=$ac_cv_prog_RANLIB
if test -n "$RANLIB"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $RANLIB" >&5
$as_echo "$RANLIB" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_RANLIB"; then
  ac_ct_RANLIB=$RANLIB
  # Extract the first word of "ranlib", so it can be a program name
  with args.
  set dummy ranlib; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_RANLIB+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_RANLIB"; then
      ac_cv_prog_ac_ct_RANLIB="$ac_ct_RANLIB" # Let the user override the
      test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.

```



```

        for ac_exec_ext in ' ' $ac_executable_extensions; do
        if as_fn_executable_p "$sas_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_RANLIB="ranlib"
            $sas_echo "$sas_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
        fi
    done
    done
IFS=$sas_save_IFS

fi
fi
ac_ct_RANLIB=$ac_cv_prog_ac_ct_RANLIB
if test -n "$ac_ct_RANLIB"; then
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $ac_ct_RANLIB" >&5
$as_echo "$ac_ct_RANLIB" >&6; }
else
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    if test "x$ac_ct_RANLIB" = x; then
        RANLIB=":"
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$sas_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        RANLIB=$ac_ct_RANLIB
    fi
else
    RANLIB="$ac_cv_prog_RANLIB"
fi

test -z "$RANLIB" && RANLIB=:

# Determine commands to create old-style static archives.
old_archive_cmds='$AR $AR_FLAGS $oldlib$oldobjs'
old_postinstall_cmds='chmod 644 $oldlib'
old_postuninstall_cmds=

if test -n "$RANLIB"; then

```

```
case $host_os in
openbsd*)
    old_postinstall_cmds="$old_postinstall_cmds~\${RANLIB} -t
\${tool_oldlib}"
    ;;
*)
    old_postinstall_cmds="$old_postinstall_cmds~\${RANLIB}
\${tool_oldlib}"
    ;;
esac
old_archive_cmds="$old_archive_cmds~\${RANLIB} \${tool_oldlib}"
fi
```

```
case $host_os in
darwin*)
    lock_old_archive_extraction=yes ;;
*)
    lock_old_archive_extraction=no ;;
esac
```

```

# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC

# Check for command to grab the raw symbol name followed by C symbol
from nm.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking command to parse $NM
output from $compiler object" >&5
$as_echo_n "checking command to parse $NM output from $compiler
object... " >&6; }
if ${lt_cv_sys_global_symbol_pipe+:} false; then :
  $as_echo_n "(cached) " >&6
else
# These are sane defaults that work on at least a few old systems.
# [They come from Ultrix.  What could be older than Ultrix?!! ;)]

# Character class describing NM global symbol codes.
symcode='[BCDEGRST]'

# Regexp to match symbols that can be accessed directly from C.
sympat='\([^_A-Za-z][_A-Za-z0-9]*\)'

# Define system-specific variables.
case $host_os in
aix*)
  symcode='[BCDT]'
  ;;
cygwin* | mingw* | pw32* | cegcc*)
  symcode='[ABCDGISTW]'
  ;;
hpux*)
  if test "$host_cpu" = ia64; then
    symcode='[BCDEGRST]'
  fi
  ;;
irix* | nonstopux*)
  symcode='[BCDEGRST]'
  ;;
osf*)

```

```

    symcode=' [BCDEGQRST] '
    ;;
solaris*)
    symcode=' [BDRT] '
    ;;
sco3.2v5*)
    symcode=' [DT] '
    ;;
sysv4.2uw2*)
    symcode=' [DT] '
    ;;
sysv5* | sco5v6* | unixware* | OpenUNIX*)
    symcode=' [ABDT] '
    ;;
sysv4)
    symcode=' [DFNSTU] '
    ;;
esac

# If we're using GNU nm, then use its standard symbol codes.
case ` $NM -V 2>&1 ` in
*GNU* | *'with BFD'*)
    symcode=' [ABCDGIRSTW] ' ;;
esac

# Transform an extracted symbol line into a proper C declaration.
# Some systems (esp. on ia64) link data and code symbols differently,
# so use this general approach.
lt_cv_sys_global_symbol_to_cdecl="sed -n -e 's/^T .* \\.(\.*)$/extern
int \1();/p' -e 's/^$symcode* .* \\.(\.*)$/extern char \1;/p'"

# Transform an extracted symbol line into symbol name and symbol
address
lt_cv_sys_global_symbol_to_c_name_address="sed -n -e 's/^: \.([\^ ]*)[
]*$/ {\\\"\\1\\\"}, (void *) 0},/p' -e 's/^$symcode* \.([\^ ]*) \.([\^
]*)\$/ {\\\"\\2\\\"}, (void *) \&2},/p'"
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix="sed -n -e 's/^:
\([\^ ]*)[ ]*$/ {\\\"\\1\\\"}, (void *) 0},/p' -e 's/^$symcode* \.([\^
]*)\ \.(\lib[\^ ]*)\$/ {\\\"\\2\\\"}, (void *) \&2},/p' -e 's/^$symcode* \([\^
]*)\ \.([\^ ]*)\$/ {\\\"lib\\2\\\"}, (void *) \&2},/p'"

# Handle CRLF in mingw tool chain
opt_cr=
case $build_os in
mingw*)
    opt_cr=` $ECHO 'x\{0,1\}' | tr x '\015' ` # option cr in regexp
    ;;
esac

# Try without a prefix underscore, then with it.
for ac_symprfx in "" "_"; do

```

```

# Transform symcode, sympat, and symprfx into a raw symbol and a C
symbol.
symxfrm="\1 $ac_symprfx\2 \2"

# Write the raw and C identifiers.
if test "$lt_cv_nm_interface" = "MS dumpbin"; then
# Fake it for dumpbin and say T for any non-static function
# and D for any global variable.
# Also find C++ and __fastcall symbols from MSVC++,
# which start with @ or ?.
lt_cv_sys_global_symbol_pipe="$AWK '\
{last_section=section; section=\$ 3};\
/^COFF SYMBOL TABLE/{for(i in hide) delete hide[i]};\
/Section length .*#relocs.*(pick any)/{hide[last_section]=1};\
\$ 0!~/External *\\|/{next};\
/ 0+ UNDEF /{next}; / UNDEF \([^|\\)*()/next};\
{if(hide[section]) next};\
{f=0}; \$ 0~/\\(\\).*\\|/{f=1}; {printf f ? \"T \" : \"D \"};\
{split(\$ 0, a, /\\|\\r/); split(a[2], s)};\
s[1]~/^[@?]/{print s[1], s[1]; next};\
s[1]~prfx {split(s[1],t,\"@\\"); print t[1],\
substr(t[1],length(prfx))}"\
' prfx=^$ac_symprfx"
else
lt_cv_sys_global_symbol_pipe="sed -n -e 's/^[^*[\
\\]($symcode$symcode*\\)[ \t]*\
]*$ac_symprfx$sympat$opt_cr$/$symxfrm/p'"
fi
lt_cv_sys_global_symbol_pipe="$lt_cv_sys_global_symbol_pipe | sed '/\
__gnu_lto/d'"

# Check to see that the pipe works correctly.
pipe_works=no

rm -f conftest*
cat > conftest.$ac_ext <<_LT_EOF
#ifdef __cplusplus
extern "C" {
#endif
char nm_test_var;
void nm_test_func(void);
void nm_test_func(void){}
#ifdef __cplusplus
}
#endif
int main(){nm_test_var='a';nm_test_func();return(0);}
_LT_EOF

if { { eval echo "\"\$as_me\"":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?

```

```

$as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
test $ac_status = 0; }]; then
  # Now try to grab the symbols.
  nlist=confptest.nm
  if { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}: \"\$NM
confptest.$ac_objext \\\| \"$lt_cv_sys_global_symbol_pipe" \> $nlist\\""; }
>&5
  (eval $NM confptest.$ac_objext \\\| \"$lt_cv_sys_global_symbol_pipe" \>
$nlist) 2>&5
  ac_status=${?}
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; } && test -s "$nlist"; then
    # Try sorting and uniquifying the output.
    if sort "$nlist" | uniq > "$nlist.T"; then
      mv -f "$nlist.T" "$nlist"
    else
      rm -f "$nlist.T"
    fi

    # Make sure that we snagged all the symbols we need.
    if $GREP ' nm_test_var$' "$nlist" >/dev/null; then
      if $GREP ' nm_test_func$' "$nlist" >/dev/null; then
        cat <<_LT_EOF > confptest.$ac_ext
/* Keep this code in sync between libtool.m4, ltmain, lt_system.h, and
tests. */
#if defined(_WIN32) || defined(__CYGWIN__) || defined(_WIN32_WCE)
/* DATA imports from DLLs on WIN32 can't be const, because runtime
relocations are performed -- see ld's documentation on pseudo-
relocs. */
# define LT_DLSYM_CONST
#elif defined(__osf__)
/* This system does not cope well with relocations in const data. */
# define LT_DLSYM_CONST
#else
# define LT_DLSYM_CONST const
#endif

#ifdef __cplusplus
extern "C" {
#endif

_LT_EOF
  # Now generate the symbol file.
  eval "$lt_cv_sys_global_symbol_to_cdecl" < "$nlist" | $GREP -v
main >> confptest.$ac_ext'

  cat <<_LT_EOF >> confptest.$ac_ext

/* The mapping between symbol names and symbols. */
LT_DLSYM_CONST struct {
  const char *name;
  void *address;

```

```

}
lt__PROGRAM__LTX_preloaded_symbols[] =
{
  { "@PROGRAM@", (void *) 0 },
_LT_EOF
  $SED "s/^\$symcode$symcode* \(.*\)\ \(.*)$/ {\\"2\", (void *)
\&\2},/" < "$nlist" | $GREP -v main >> confptest.$ac_ext
  cat <<\_LT_EOF >> confptest.$ac_ext
  {0, (void *) 0}
};

/* This works around a problem in FreeBSD linker */
#ifdef FREEBSD_WORKAROUND
static const void *lt_preloaded_setup() {
  return lt__PROGRAM__LTX_preloaded_symbols;
}
#endif

#ifdef __cplusplus
}
#endif
_LT_EOF
  # Now try linking the two files.
  mv confptest.$ac_objext confftstm.$ac_objext
  lt_globsym_save_LIBS=$LIBS
  lt_globsym_save_CFLAGS=$CFLAGS
  LIBS="confftstm.$ac_objext"
  CFLAGS="$CFLAGS$lt_prog_compiler_no_built_in_flag"
  if { { eval echo "\\"$as_me\"":${as_lineno-$LINENO}:
\"$ac_link\>"; } >&5
  (eval $ac_link) 2>&5
  ac_status=$?
  $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
  test $ac_status = 0; } && test -s confptest${ac_exeext}; then
    pipe_works=yes
  fi
  LIBS=$lt_globsym_save_LIBS
  CFLAGS=$lt_globsym_save_CFLAGS
  else
    echo "cannot find nm_test_func in $nlist" >&5
  fi
  else
    echo "cannot find nm_test_var in $nlist" >&5
  fi
  else
    echo "cannot run $lt_cv_sys_global_symbol_pipe" >&5
  fi
  else
    echo "$progname: failed program was:" >&5
    cat confptest.$ac_ext >&5
  fi
  rm -rf confptest* confftst*

```

```

# Do not use the global_symbol_pipe unless it works.
if test "$pipe_works" = yes; then
    break
else
    lt_cv_sys_global_symbol_pipe=
fi
done

fi

if test -z "$lt_cv_sys_global_symbol_pipe"; then
    lt_cv_sys_global_symbol_to_cdecl=
fi
if test -z
"$lt_cv_sys_global_symbol_pipe$lt_cv_sys_global_symbol_to_cdecl"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: failed" >&5
$as_echo "failed" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: ok" >&5
$as_echo "ok" >&6; }
fi

# Response file support.
if test "$lt_cv_nm_interface" = "MS dumpbin"; then
    nm_file_list_spec='@'
elif $NM --help 2>/dev/null | grep '[@]FILE' >/dev/null; then
    nm_file_list_spec='@'
fi

```



```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for sysroot" >&5
$as_echo_n "checking for sysroot... " >&6; }

# Check whether --with-libtool-sysroot was given.
if test "${with_libtool_sysroot+set}" = set; then :
  withval=$with_libtool_sysroot;
else
  with_libtool_sysroot=no
fi

lt_sysroot=
case ${with_libtool_sysroot} in #(
  yes)
  if test "$GCC" = yes; then
    lt_sysroot=`$CC --print-sysroot 2>/dev/null`
  fi
  ;; #(
/*)
  lt_sysroot=`echo "$with_libtool_sysroot" | sed -e
"$sed_quote_subst"`
  ;; #(
no|'')
  ;; #(
*)
  { $as_echo "$as_me:${as_lineno-$LINENO}: result:
${with_libtool_sysroot}" >&5
$as_echo "${with_libtool_sysroot}" >&6; }
  as_fn_error $? "The sysroot must be an absolute path." "$LINENO" 5
  ;;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: ${lt_sysroot:-no}"
>&5
$as_echo "${lt_sysroot:-no}" >&6; }

# Check whether --enable-libtool-lock was given.
if test "${enable_libtool_lock+set}" = set; then :
  enableval=$enable_libtool_lock;
fi

test "x$enable_libtool_lock" != xno && enable_libtool_lock=yes

# Some flags need to be propagated to the compiler or linker for good

```

```

# libtool support.
case $host in
ia64-*-hpux*)
  # Find out which ABI we are using.
  echo 'int i;' > conftest.$ac_ext
  if { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
    (eval $ac_compile) 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    case ` /usr/bin/file conftest.$ac_objext` in
      *ELF-32*)
        HPUX_IA64_MODE="32"
        ;;
      *ELF-64*)
        HPUX_IA64_MODE="64"
        ;;
    esac
  fi
  rm -rf conftest*
  ;;
*-*-irix6*)
  # Find out which ABI we are using.
  echo '#line '$LINENO' "configure"' > conftest.$ac_ext
  if { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
    (eval $ac_compile) 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    if test "$lt_cv_prog_gnu_ld" = yes; then
      case ` /usr/bin/file conftest.$ac_objext` in
        *32-bit*)
          LD="${LD-ld} -melf32basmip"
          ;;
        *N32*)
          LD="${LD-ld} -melf32bmipn32"
          ;;
        *64-bit*)
          LD="${LD-ld} -melf64bmip"
          ;;
      esac
    else
      case ` /usr/bin/file conftest.$ac_objext` in
        *32-bit*)
          LD="${LD-ld} -32"
          ;;
        *N32*)
          LD="${LD-ld} -n32"
          ;;
        *64-bit*)

```

```

        LD="{LD-ld} -64"
        ;;
    esac
fi
fi
rm -rf conftest*
;;

x86_64-*kfreebsd*-gnu|x86_64-*linux*|ppc*-*linux*|powerpc*-*linux*| \
s390*-*linux*|s390*-*tpf*|sparc*-*linux*)
# Find out which ABI we are using.
echo 'int i;' > conftest.$ac_ext
if { { eval echo "\"\${as_me}\"":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }; then
    case `usr/bin/file conftest.o` in
        *32-bit*)
            case $host in
                x86_64-*kfreebsd*-gnu)
                    LD="{LD-ld} -m elf_i386_fbsd"
                    ;;
                x86_64-*linux*)
                    LD="{LD-ld} -m elf_i386"
                    ;;
                ppc64-*linux*|powerpc64-*linux*)
                    LD="{LD-ld} -m elf32ppclinux"
                    ;;
                s390x-*linux*)
                    LD="{LD-ld} -m elf_s390"
                    ;;
                sparc64-*linux*)
                    LD="{LD-ld} -m elf32_sparc"
                    ;;
            esac
        ;;
        *64-bit*)
            case $host in
                x86_64-*kfreebsd*-gnu)
                    LD="{LD-ld} -m elf_x86_64_fbsd"
                    ;;
                x86_64-*linux*)
                    LD="{LD-ld} -m elf_x86_64"
                    ;;
                ppc*-*linux*|powerpc*-*linux*)
                    LD="{LD-ld} -m elf64ppc"
                    ;;
                s390*-*linux*|s390*-*tpf*)
                    LD="{LD-ld} -m elf64_s390"
                    ;;
            esac
        ;;
    esac
fi

```

```

        sparc*-*linux*)
            LD="${LD-ld} -m elf64_sparc"
            ;;
        esac
    ;;
esac
fi
rm -rf conftest*
;;

*-*sco3.2v5*)
# On SCO OpenServer 5, we need -belf to get full-featured binaries.
SAVE_CFLAGS="$CFLAGS"
CFLAGS="$CFLAGS -belf"
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the C
compiler needs -belf" >&5
$as_echo_n "checking whether the C compiler needs -belf... " >&6; }
if ${lt_cv_cc_needs_belf+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    lt_cv_cc_needs_belf=yes
else
    lt_cv_cc_needs_belf=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
    ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_cc_needs_belf"
>&5
$as_echo "$lt_cv_cc_needs_belf" >&6; }
if test x"$lt_cv_cc_needs_belf" != x"yes"; then
  # this is probably gcc 2.8.0, egcs 1.0 or newer; no need for -belf
  CFLAGS="$SAVE_CFLAGS"
fi
;;
*-*solaris*)
# Find out which ABI we are using.
echo 'int i;' > conftest.$ac_ext
if { { eval echo "\"\${as_me}\":${as_lineno-$LINENO}:"
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }; then
  case `usr/bin/file conftest.o` in
    *64-bit*)
      case $lt_cv_prog_gnu_ld in
        yes*)
          case $host in
            i?86-*-*solaris*)
              LD="{LD-ld} -m elf_x86_64"
              ;;
            sparc*-*-*solaris*)
              LD="{LD-ld} -m elf64_sparc"
              ;;
          esac
          # GNU ld 2.21 introduced _sol2 emulations. Use them if
          available.
          if {LD-ld} -V | grep _sol2 >/dev/null 2>&1; then
            LD="{LD-ld}_sol2"
          fi
          ;;
        *)
          if {LD-ld} -64 -r -o conftest2.o conftest.o >/dev/null 2>&1;
then
            LD="{LD-ld} -64"
          fi
          ;;
        esac
      ;;
    esac
  fi
  rm -rf conftest*
  ;;
esac

need_locks="$enable_libtool_lock"

```

```

if test -n "$ac_tool_prefix"; then
  # Extract the first word of "${ac_tool_prefix}mt", so it can be a
  program name with args.
  set dummy ${ac_tool_prefix}mt; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_MANIFEST_TOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$MANIFEST_TOOL"; then
      ac_cv_prog_MANIFEST_TOOL="$MANIFEST_TOOL" # Let the user override
      the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_MANIFEST_TOOL="${ac_tool_prefix}mt"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      MANIFEST_TOOL=$ac_cv_prog_MANIFEST_TOOL
      if test -n "$MANIFEST_TOOL"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $MANIFEST_TOOL" >&5
        $as_echo "$MANIFEST_TOOL" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi
    fi

    fi
    if test -z "$ac_cv_prog_MANIFEST_TOOL"; then
      ac_ct_MANIFEST_TOOL=$MANIFEST_TOOL
      # Extract the first word of "mt", so it can be a program name with
      args.
      set dummy mt; ac_word=$2
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $as_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_ac_ct_MANIFEST_TOOL+:} false; then :
        $as_echo_n "(cached) " >&6
      else

```

```

    if test -n "$ac_ct_MANIFEST_TOOL"; then
      ac_cv_prog_ac_ct_MANIFEST_TOOL="$ac_ct_MANIFEST_TOOL" # Let the user
      override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_MANIFEST_TOOL="mt"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      ac_ct_MANIFEST_TOOL=$ac_cv_prog_ac_ct_MANIFEST_TOOL
      if test -n "$ac_ct_MANIFEST_TOOL"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_ct_MANIFEST_TOOL" >&5
$as_echo "$ac_ct_MANIFEST_TOOL" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
      fi

      if test "x$ac_ct_MANIFEST_TOOL" = x; then
        MANIFEST_TOOL=":"
      else
        case $cross_compiling:$ac_tool_warned in
        yes:)
          { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
          ac_tool_warned=yes ;;
        esac
        MANIFEST_TOOL=$ac_ct_MANIFEST_TOOL
      fi
    else
      MANIFEST_TOOL="$ac_cv_prog_MANIFEST_TOOL"
    fi

    test -z "$MANIFEST_TOOL" && MANIFEST_TOOL=mt
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $MANIFEST_TOOL is
a manifest tool" >&5

```

```

$as_echo_n "checking if $MANIFEST_TOOL is a manifest tool... " >&6; }
if ${lt_cv_path_manifest_tool+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_path_manifest_tool=no
  echo "$as_me:$LINENO: $MANIFEST_TOOL '-?'" >&5
  $MANIFEST_TOOL '-?' 2>confptest.err > confptest.out
  cat confptest.err >&5
  if $GREP 'Manifest Tool' confptest.out > /dev/null; then
    lt_cv_path_manifest_tool=yes
  fi
  rm -f confptest*
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_path_manifest_tool" >&5
$as_echo "$lt_cv_path_manifest_tool" >&6; }
if test "x$lt_cv_path_manifest_tool" != xyes; then
  MANIFEST_TOOL=:
fi

```

```

case $host_os in
  rhapsody* | darwin*)
    if test -n "$ac_tool_prefix"; then
      # Extract the first word of "${ac_tool_prefix}dsymutil", so it can
      be a program name with args.
      set dummy ${ac_tool_prefix}dsymutil; ac_word=$2
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
      $as_echo_n "checking for $ac_word... " >&6; }
      if ${ac_cv_prog_DSYMUTIL+:} false; then :
        $as_echo_n "(cached) " >&6
      else
        if test -n "$DSYMUTIL"; then
          ac_cv_prog_DSYMUTIL="$DSYMUTIL" # Let the user override the test.
        else
          as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
          for as_dir in $PATH
          do
            IFS=$as_save_IFS
            test -z "$as_dir" && as_dir=.
            for ac_exec_ext in ' ' $ac_executable_extensions; do
              if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                ac_cv_prog_DSYMUTIL="${ac_tool_prefix}dsymutil"
                $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                break 2
              fi
            done
          done
        fi
      fi
    fi
  done

```



```

done
IFS=$as_save_IFS

fi
fi
DSYMUTIL=$ac_cv_prog_DSYMUTIL
if test -n "$DSYMUTIL"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DSYMUTIL" >&5
$as_echo "$DSYMUTIL" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_DSYMUTIL"; then
  ac_ct_DSYMUTIL=$DSYMUTIL
  # Extract the first word of "dsymutil", so it can be a program name
  with args.
  set dummy dsymutil; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_DSYMUTIL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_DSYMUTIL"; then
      ac_cv_prog_ac_ct_DSYMUTIL="$ac_ct_DSYMUTIL" # Let the user override
      the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_DSYMUTIL="dsymutil"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

fi
fi
ac_ct_DSYMUTIL=$ac_cv_prog_ac_ct_DSYMUTIL
if test -n "$ac_ct_DSYMUTIL"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_DSYMUTIL"
>&5

```

```

$as_echo "$ac_ct_DSYMUTIL" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_DSYMUTIL" = x; then
    DSYMUTIL=":"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    DSYMUTIL=$ac_ct_DSYMUTIL
  fi
else
  DSYMUTIL="$ac_cv_prog_DSYMUTIL"
fi

  if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}nmedit", so it can be
    a program name with args.
    set dummy ${ac_tool_prefix}nmedit; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_NMEDIT+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      if test -n "$NMEDIT"; then
        ac_cv_prog_NMEDIT="$NMEDIT" # Let the user override the test.
      else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_NMEDIT="${ac_tool_prefix}nmedit"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi

```

```

fi
NMEDIT=$ac_cv_prog_NMEDIT
if test -n "$NMEDIT"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $NMEDIT" >&5
$as_echo "$NMEDIT" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_NMEDIT"; then
  ac_ct_NMEDIT=$NMEDIT
  # Extract the first word of "nmedit", so it can be a program name
  with args.
  set dummy nmedit; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_NMEDIT+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_NMEDIT"; then
      ac_cv_prog_ac_ct_NMEDIT="$ac_ct_NMEDIT" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_NMEDIT="nmedit"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS
    fi
  fi
  ac_ct_NMEDIT=$ac_cv_prog_ac_ct_NMEDIT
  if test -n "$ac_ct_NMEDIT"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_NMEDIT" >&5
$as_echo "$ac_ct_NMEDIT" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  fi
fi

```

```

if test "x$ac_ct_NMEDIT" = x; then
  NMEDIT=":"
else
  case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
  NMEDIT=$ac_ct_NMEDIT
  fi
else
  NMEDIT="$ac_cv_prog_NMEDIT"
  fi

  if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}lipo", so it can be a
    program name with args.
    set dummy ${ac_tool_prefix}lipo; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_LIPO+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      if test -n "$LIPO"; then
        ac_cv_prog_LIPO="$LIPO" # Let the user override the test.
      else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
      ac_cv_prog_LIPO="${ac_tool_prefix}lipo"
      $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
      break 2
    fi
  done
done
IFS=$as_save_IFS

fi
fi
LIPO=$ac_cv_prog_LIPO
if test -n "$LIPO"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LIPO" >&5
$as_echo "$LIPO" >&6; }

```

```

else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_LIPO"; then
  ac_ct_LIPO=$LIPO
  # Extract the first word of "lipo", so it can be a program name with
  args.
  set dummy lipo; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_LIPO+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_LIPO"; then
      ac_cv_prog_ac_ct_LIPO="$ac_ct_LIPO" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in '$ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_LIPO="lipo"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

    fi
  fi
  ac_ct_LIPO=$ac_cv_prog_ac_ct_LIPO
  if test -n "$ac_ct_LIPO"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_LIPO" >&5
$as_echo "$ac_ct_LIPO" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  fi

  if test "x$ac_ct_LIPO" = x; then
    LIPO=":"
  else
    case $cross_compiling:$ac_tool_warned in

```

```

yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    LIPO=$ac_ct_LIPO
    fi
else
    LIPO="$ac_cv_prog_LIPO"
fi

    if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}otool", so it can be a
program name with args.
set dummy ${ac_tool_prefix}otool; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_OTOOL+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -n "$OTOOL"; then
        ac_cv_prog_OTOOL="$OTOOL" # Let the user override the test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_OTOOL="${ac_tool_prefix}otool"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi
OTOOL=$ac_cv_prog_OTOOL
if test -n "$OTOOL"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OTOOL" >&5
$as_echo "$OTOOL" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

```

```

fi
if test -z "$ac_cv_prog_OTOOL"; then
  ac_ct_OTOOL=$OTOOL
  # Extract the first word of "otool", so it can be a program name
  with args.
  set dummy otool; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
  $as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_OTOOL+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_OTOOL"; then
      ac_cv_prog_ac_ct_OTOOL="$ac_ct_OTOOL" # Let the user override the
      test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_OTOOL="otool"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
            $as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      fi
      fi
      ac_ct_OTOOL=$ac_cv_prog_ac_ct_OTOOL
      if test -n "$ac_ct_OTOOL"; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OTOOL" >&5
        $as_echo "$ac_ct_OTOOL" >&6; }
      else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
        $as_echo "no" >&6; }
      fi

      if test "x$ac_ct_OTOOL" = x; then
        OTOOL=":"
      else
        case $cross_compiling:$ac_tool_warned in
        yes:)
          { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
          not prefixed with host triplet" >&5
          $as_echo "$as_me: WARNING: using cross tools not prefixed with host
          triplet" >&2;}
        }
      }
    }
  }
}

```

```

ac_tool_warned=yes ;;
esac
    OTOOL=$ac_ct_OTOOL
    fi
else
    OTOOL="$ac_cv_prog_OTOOL"
fi

    if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}otool64", so it can be
    a program name with args.
    set dummy ${ac_tool_prefix}otool64; ac_word=$2
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $as_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_OTOOL64+:} false; then :
        $as_echo_n "(cached) " >&6
    else
        if test -n "$OTOOL64"; then
            ac_cv_prog_OTOOL64="$OTOOL64" # Let the user override the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '' $ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_prog_OTOOL64="${ac_tool_prefix}otool64"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

            fi
            fi
            OTOOL64=$ac_cv_prog_OTOOL64
            if test -n "$OTOOL64"; then
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: $OTOOL64" >&5
                $as_echo "$OTOOL64" >&6; }
            else
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
                $as_echo "no" >&6; }
            fi

            fi
            if test -z "$ac_cv_prog_OTOOL64"; then
                ac_ct_OTOOL64=$OTOOL64

```



```

# Extract the first word of "otool64", so it can be a program name
with args.
set dummy otool64; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_ac_ct_OTOOL64+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -n "$ac_ct_OTOOL64"; then
    ac_cv_prog_ac_ct_OTOOL64="$ac_ct_OTOOL64" # Let the user override
the test.
  else
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
    for as_dir in $PATH
    do
      IFS=$as_save_IFS
      test -z "$as_dir" && as_dir=.
      for ac_exec_ext in ' $ac_executable_extensions; do
        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
          ac_cv_prog_ac_ct_OTOOL64="otool64"
          $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
          break 2
        fi
      done
    done
    IFS=$as_save_IFS

    fi
    fi
    ac_ct_OTOOL64=$ac_cv_prog_ac_ct_OTOOL64
    if test -n "$ac_ct_OTOOL64"; then
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_OTOOL64" >&5
$as_echo "$ac_ct_OTOOL64" >&6; }
    else
      { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi

    if test "x$ac_ct_OTOOL64" = x; then
      OTOOL64=":"
    else
      case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
      OTOOL64=$ac_ct_OTOOL64
    fi

```

```
else
  OTOOL64="$ac_cv_prog_OTOOL64"
fi
```

```
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for -
single_module linker flag" >&5
$as_echo_n "checking for -single_module linker flag... " >&6; }
if ${lt_cv_apple_cc_single_mod+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_apple_cc_single_mod=no
  if test -z "${LT_MULTI_MODULE}"; then
    # By default we will add the -single_module flag. You can
override
    # by either setting the environment variable LT_MULTI_MODULE
    # non-empty at configure time, or by adding -multi_module to the
    # link flags.
    rm -rf libconfptest.dylib*
    echo "int foo(void){return 1;}" > confptest.c
    echo "$LTCC $LTCFLAGS $LDFLAGS -o libconfptest.dylib \
-dynamiclib -Wl,-single_module confptest.c" >&5
    $LTCC $LTCFLAGS $LDFLAGS -o libconfptest.dylib \
    -dynamiclib -Wl,-single_module confptest.c 2>confptest.err
    _lt_result=$?
    # If there is a non-empty error log, and "single_module"
    # appears in it, assume the flag caused a linker warning
```

```

        if test -s confptest.err && $GREP single_module confptest.err;
then
    cat confptest.err >&5
    # Otherwise, if the output was created with a 0 exit code from
    # the compiler, it worked.
    elif test -f libconfptest.dylib && test $_lt_result -eq 0; then
        lt_cv_apple_cc_single_mod=yes
    else
        cat confptest.err >&5
    fi
    rm -rf libconfptest.dylib*
    rm -f confptest.*
    fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_apple_cc_single_mod" >&5
$as_echo "$lt_cv_apple_cc_single_mod" >&6; }

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for -
exported_symbols_list linker flag" >&5
$as_echo_n "checking for -exported_symbols_list linker flag... " >&6;
}
if ${lt_cv_ld_exported_symbols_list+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_ld_exported_symbols_list=no
    save_LDFLAGS=$LDFLAGS
    echo "_main" > confptest.sym
    LDFLAGS="$LDFLAGS -Wl,-exported_symbols_list,confptest.sym"
    cat confdefs.h - <<_ACEOF >>confptest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    lt_cv_ld_exported_symbols_list=yes
else
    lt_cv_ld_exported_symbols_list=no
fi
rm -f core confptest.err confptest.$ac_objext \
    confptest.$ac_exeext confptest.$ac_ext
    LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_ld_exported_symbols_list" >&5

```

```

$as_echo "$lt_cv_ld_exported_symbols_list" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for -force_load
linker flag" >&5
$as_echo_n "checking for -force_load linker flag... " >&6; }
if ${lt_cv_ld_force_load+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_ld_force_load=no
  cat > conftest.c << _LT_EOF
int forced_loaded() { return 2;}
_LT_EOF
  echo "$LTCC $LTCFLAGS -c -o conftest.o conftest.c" >&5
  $LTCC $LTCFLAGS -c -o conftest.o conftest.c 2>&5
  echo "$AR cru libconftest.a conftest.o" >&5
  $AR cru libconftest.a conftest.o 2>&5
  echo "$RANLIB libconftest.a" >&5
  $RANLIB libconftest.a 2>&5
  cat > conftest.c << _LT_EOF
int main() { return 0;}
_LT_EOF
  echo "$LTCC $LTCFLAGS $LDFLAGS -o conftest conftest.c -Wl,-
force_load,./libconftest.a" >&5
  $LTCC $LTCFLAGS $LDFLAGS -o conftest conftest.c -Wl,-
force_load,./libconftest.a 2>conftest.err
  _lt_result=$?
  if test -s conftest.err && $GREP force_load conftest.err; then
    cat conftest.err >&5
  elif test -f conftest && test $_lt_result -eq 0 && $GREP
forced_load conftest >/dev/null 2>&1 ; then
    lt_cv_ld_force_load=yes
  else
    cat conftest.err >&5
  fi
  rm -f conftest.err libconftest.a conftest conftest.c
  rm -rf conftest.dSYM

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_ld_force_load"
>&5
$as_echo "$lt_cv_ld_force_load" >&6; }
case $host_os in
rhapsody* | darwin1.[012])
  _lt_dar_allow_undefined='${wl}-undefined ${wl}suppress' ;;
darwin1.*)
  _lt_dar_allow_undefined='${wl}-flat_namespace ${wl}-undefined
${wl}suppress' ;;
darwin*) # darwin 5.x on
  # if running on 10.5 or later, the deployment target defaults
  # to the OS version, if on x86, and 10.4, the deployment
  # target defaults to 10.4. Don't you love it?
  case ${MACOSX_DEPLOYMENT_TARGET-10.0},$host in

```

```

    10.0,*86*-darwin8*|10.0,*-darwin[91]*)
        _lt_dar_allow_undefined='${wl}-undefined ${wl}dynamic_lookup'
;;
    10.[012]*)
        _lt_dar_allow_undefined='${wl}-flat_namespace ${wl}-undefined
${wl}suppress' ;;
    10.*)
        _lt_dar_allow_undefined='${wl}-undefined ${wl}dynamic_lookup'
;;
    esac
;;
esac
if test "$lt_cv_apple_cc_single_mod" = "yes"; then
    _lt_dar_single_mod='single_module'
fi
if test "$lt_cv_ld_exported_symbols_list" = "yes"; then
    _lt_dar_export_syms=' ${wl}-
exported_symbols_list,$output_objdir/${libname}-symbols.expsym'
else
    _lt_dar_export_syms='~$NMEDIT -s $output_objdir/${libname}-
symbols.expsym ${lib}'
fi
if test "$DSYMUTIL" != ":" && test "$lt_cv_ld_force_load" = "no";
then
    _lt_dsymutil='~$DSYMUTIL $lib || :'
else
    _lt_dsymutil=
fi
;;
esac

for ac_header in dlfcn.h
do :
    ac_fn_c_check_header_compile "$LINENO" "dlfcn.h"
"ac_cv_header_dlfcn_h" "$ac_includes_default
"
if test "x$ac_cv_header_dlfcn_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_DLFCN_H 1
_ACEOF

fi

done

func_stripname_cnf ()
{
    case ${2} in

```

```

    *) func_stripname_result=`$ECHO "${3}" | $SED "s%^${1}%%;
s%\\\\\\${2}\\$%%" `;;
    *) func_stripname_result=`$ECHO "${3}" | $SED "s%^${1}%%;
s%${2}\\$%%" `;;
    esac
} # func_stripname_cnf

```

```

# Set options

```

```

    enable_dlopen=no

```

```

    enable_win32_dll=no

```

```

        # Check whether --enable-shared was given.
if test "${enable_shared+set}" = set; then :
    enableval=$enable_shared; p=${PACKAGE-default}
    case $enableval in
    yes) enable_shared=yes ;;
    no) enable_shared=no ;;
    *)
        enable_shared=no
        # Look at the argument we got.  We use all the common list
separators.
        lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
        for pkg in $enableval; do
            IFS="$lt_save_ifs"
            if test "X$pkg" = "X$p"; then
                enable_shared=yes
            fi
        done
        IFS="$lt_save_ifs"
        ;;
    esac
else
    enable_shared=yes
fi

```

```

# Check whether --enable-static was given.
if test "${enable_static+set}" = set; then :
  enableval=$enable_static; p=${PACKAGE-default}
  case $enableval in
    yes) enable_static=yes ;;
    no) enable_static=no ;;
    *)
      enable_static=no
      # Look at the argument we got. We use all the common list
separators.
      lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
      for pkg in $enableval; do
        IFS="$lt_save_ifs"
        if test "X$pkg" = "X$p"; then
          enable_static=yes
        fi
      done
      IFS="$lt_save_ifs"
      ;;
  esac
else
  enable_static=yes
fi

```

```

# Check whether --with-pic was given.
if test "${with_pic+set}" = set; then :
  withval=$with_pic; lt_p=${PACKAGE-default}
  case $withval in
    yes|no) pic_mode=$withval ;;
    *)
      pic_mode=default
      # Look at the argument we got. We use all the common list
separators.
      lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
      for lt_pkg in $withval; do
        IFS="$lt_save_ifs"
        if test "X$lt_pkg" = "X$lt_p"; then
          pic_mode=yes
        fi
      done
      IFS="$lt_save_ifs"
      ;;
  esac

```

```

        esac
    else
        pic_mode=default
    fi

test -z "$pic_mode" && pic_mode=default

# Check whether --enable-fast-install was given.
if test "${enable_fast_install+set}" = set; then :
    enableval=$enable_fast_install; p=${PACKAGE-default}
    case $enableval in
    yes) enable_fast_install=yes ;;
    no) enable_fast_install=no ;;
    *)
        enable_fast_install=no
        # Look at the argument we got.  We use all the common list
        separators.
        lt_save_ifs="$IFS"; IFS="${IFS}$PATH_SEPARATOR,"
        for pkg in $enableval; do
            IFS="$lt_save_ifs"
            if test "X$pkg" = "X$p"; then
                enable_fast_install=yes
            fi
        done
        IFS="$lt_save_ifs"
        ;;
    esac
else
    enable_fast_install=yes
fi

```

```

# This can be used to rebuild libtool when needed
LIBTOOL_DEPS="$ltmain"

```

```

# Always use our own libtool.

```



```
LIBTOOL='$(top_builddir)'  
LIBTOOL="$LIBTOOL/${host_alias}-libtool"
```

```
test -z "$LN_S" && LN_S="ln -s"
```

```
if test -n "${ZSH_VERSION+set}" ; then  
  setopt NO_GLOB_SUBST  
fi
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for objdir" >&5
```

```

$as_echo_n "checking for objdir... " >&6; }
if ${lt_cv_objdir+:} false; then :
  $as_echo_n "(cached) " >&6
else
  rm -f .libs 2>/dev/null
  mkdir .libs 2>/dev/null
  if test -d .libs; then
    lt_cv_objdir=.libs
  else
    # MS-DOS does not allow filenames that begin with a dot.
    lt_cv_objdir=_libs
  fi
  rmdir .libs 2>/dev/null
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_objdir" >&5
$as_echo "$lt_cv_objdir" >&6; }
objdir=$lt_cv_objdir

```

```

cat >>confdefs.h <<_ACEOF
#define LT_OBJDIR "$lt_cv_objdir/"
_ACEOF

```

```

case $host_os in
aix3*)
  # AIX sometimes has problems with the GCC collect2 program.  For
  some
  # reason, if we set the COLLECT_NAMES environment variable, the
  problems
  # vanish in a puff of smoke.
  if test "X${COLLECT_NAMES+set}" != Xset; then
    COLLECT_NAMES=
    export COLLECT_NAMES
  fi
  ;;
esac

```

```

# Global variables:
ofile=${host_alias}-libtool
can_build_shared=yes

```

```

# All known linkers require a `.a' archive for static linking (except
MSVC,
# which needs '.lib').
libext=a

```

```

with_gnu_ld="$lt_cv_prog_gnu_ld"

old_CC="$CC"
old_CFLAGS="$CFLAGS"

# Set sane defaults for various variables
test -z "$CC" && CC=cc
test -z "$LTCC" && LTCC=$CC
test -z "$LTCFLAGS" && LTCFLAGS=$CFLAGS
test -z "$LD" && LD=ld
test -z "$ac_objext" && ac_objext=o

for cc_temp in $compiler""; do
  case $cc_temp in
    compile | *[\//]compile | ccache | *[\//]ccache ) ;;
    distcc | *[\//]distcc | purify | *[\//]purify ) ;;
    \-*) ;;
    *) break;;
  esac
done
cc_basename=`$ECHO "$cc_temp" | $SED "s%.*/%%; s%^$host_alias-%%"`

# Only perform the check for file, if the check method requires it
test -z "$MAGIC_CMD" && MAGIC_CMD=file
case $deplibs_check_method in
file_magic*)
  if test "$file_magic_cmd" = '$MAGIC_CMD'; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
${ac_tool_prefix}file" >&5
$as_echo_n "checking for ${ac_tool_prefix}file... " >&6; }
if ${lt_cv_path_MAGIC_CMD+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $MAGIC_CMD in
[\//]* | ?:[\//]*)
    lt_cv_path_MAGIC_CMD="$MAGIC_CMD" # Let the user override the test
with a path.
    ;;
*)
    lt_save_MAGIC_CMD="$MAGIC_CMD"
    lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
    ac_dummy="/usr/bin$PATH_SEPARATOR$PATH"
    for ac_dir in $ac_dummy; do
      IFS="$lt_save_ifs"
      test -z "$ac_dir" && ac_dir=.
      if test -f $ac_dir/${ac_tool_prefix}file; then
        lt_cv_path_MAGIC_CMD="$ac_dir/${ac_tool_prefix}file"
        if test -n "$file_magic_test_file"; then
          case $deplibs_check_method in
"file_magic" *)

```

```

        file_magic_regex=`expr "$deplibs_check_method" : "file_magic
\(.*\)"`
        MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
        if eval $file_magic_cmd \$file_magic_test_file 2> /dev/null |
            $EGREP "$file_magic_regex" > /dev/null; then
            :
        else
            cat <<_LT_EOF 1>&2

*** Warning: the command libtool uses to detect shared libraries,
*** $file_magic_cmd, produces output that libtool cannot recognize.
*** The result is that libtool may fail to recognize shared libraries
*** as such. This will affect the creation of libtool libraries that
*** depend on shared libraries, but programs linked with such libtool
*** libraries will work regardless of this problem. Nevertheless, you
*** may want to report the problem to your system manager and/or to
*** bug-libtool@gnu.org

_LT_EOF
        fi ;;
    esac
    fi
    break
    fi
done
IFS="$lt_save_ifs"
MAGIC_CMD="$lt_save_MAGIC_CMD"
;;
esac
fi

MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
if test -n "$MAGIC_CMD"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $MAGIC_CMD" >&5
$as_echo "$MAGIC_CMD" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

if test -z "$lt_cv_path_MAGIC_CMD"; then
  if test -n "$ac_tool_prefix"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for file" >&5
$as_echo_n "checking for file... " >&6; }
    if ${lt_cv_path_MAGIC_CMD+:} false; then :
      $as_echo_n "(cached) " >&6
    else
      case $MAGIC_CMD in

```

```

[\\/*] | ?:[\\/*]*)
  lt_cv_path_MAGIC_CMD="$MAGIC_CMD" # Let the user override the test
with a path.
  ;;
*)
  lt_save_MAGIC_CMD="$MAGIC_CMD"
  lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR
  ac_dummy="/usr/bin$PATH_SEPARATOR$PATH"
  for ac_dir in $ac_dummy; do
    IFS="$lt_save_ifs"
    test -z "$ac_dir" && ac_dir=.
    if test -f $ac_dir/file; then
      lt_cv_path_MAGIC_CMD="$ac_dir/file"
      if test -n "$file_magic_test_file"; then
        case $deplibs_check_method in
          "file_magic "*)
            file_magic_regex=`expr "$deplibs_check_method" : "file_magic
\(.*\)"`
            MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
            if eval $file_magic_cmd \$file_magic_test_file 2> /dev/null |
              $EGREP "$file_magic_regex" > /dev/null; then
              :
            else
              cat <<_LT_EOF 1>&2

```

```

*** Warning: the command libtool uses to detect shared libraries,
*** $file_magic_cmd, produces output that libtool cannot recognize.
*** The result is that libtool may fail to recognize shared libraries
*** as such. This will affect the creation of libtool libraries that
*** depend on shared libraries, but programs linked with such libtool
*** libraries will work regardless of this problem. Nevertheless, you
*** may want to report the problem to your system manager and/or to
*** bug-libtool@gnu.org

```

```

_LT_EOF
  fi ;;
  esac
  fi
  break
  fi
done
IFS="$lt_save_ifs"
MAGIC_CMD="$lt_save_MAGIC_CMD"
;;
esac
fi

```

```

MAGIC_CMD="$lt_cv_path_MAGIC_CMD"
if test -n "$MAGIC_CMD"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $MAGIC_CMD" >&5
$as_echo "$MAGIC_CMD" >&6; }
else

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

    else
        MAGIC_CMD=:
    fi
fi

    fi
;;
esac

# Use C for the default configuration in the libtool script

lt_save_CC="$CC"
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

# Source file extension for C test sources.
ac_ext=c

# Object file extension for compiled C test sources.
objext=o
objext=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code="int some_variable = 0;"

# Code to be used in simple link tests
lt_simple_link_test_code='int main(){return(0);}'

# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC

```

```

# Save the default compiler, since it gets overwritten when the other
# tags are being tested, and _LT_TAGVAR(compiler, []) is a NOP.
compiler_DEFAULT=$CC

# save warnings/boilerplate of simple test code
ac_outfile=conftest.$ac_objext
echo "$lt_simple_compile_test_code" >conftest.$ac_ext
eval "$ac_compile" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d'
>conftest.err
_lt_compiler_boilerplate=`cat conftest.err`
$RM conftest*

ac_outfile=conftest.$ac_objext
echo "$lt_simple_link_test_code" >conftest.$ac_ext
eval "$ac_link" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d' >conftest.err
_lt_linker_boilerplate=`cat conftest.err`
$RM -r conftest*

if test -n "$compiler"; then

lt_prog_compiler_no_built_in_flag=

if test "$GCC" = yes; then
  case $cc_basename in
  nvcc*)
    lt_prog_compiler_no_built_in_flag=' -Xcompiler -fno-builtin' ;;
  *)
    lt_prog_compiler_no_built_in_flag=' -fno-builtin' ;;
  esac

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -fno-rtti -fno-exceptions" >&5
$as_echo_n "checking if $compiler supports -fno-rtti -fno-
exceptions... " >&6; }
if ${lt_cv_prog_compiler_rtti_exceptions+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_rtti_exceptions=no
  ac_outfile=conftest.$ac_objext
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext
  lt_compiler_flag="-fno-rtti -fno-exceptions"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "conftest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag :; t' \

```

```

-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
(eval echo "\"\${as_me:$LINENO: $lt_compile}\"" >&5)
(eval "$lt_compile" 2>confptest.err)
ac_status=$?
cat confptest.err >&5
echo "$as_me:$LINENO: \${?} = $ac_status" >&5
if (exit $ac_status) && test -s "$ac_outfile"; then
  # The compiler can only warn and ignore the option if not
recognized
  # So say no if there are warnings other than the usual output.
  $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >confptest.exp
  $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2
  if test ! -s confptest.er2 || diff confptest.exp confptest.er2
>/dev/null; then
    lt_cv_prog_compiler_rtti_exceptions=yes
  fi
fi
$RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_rtti_exceptions" >&5
$as_echo "$lt_cv_prog_compiler_rtti_exceptions" >&6; }

if test x"$lt_cv_prog_compiler_rtti_exceptions" = xyes; then

lt_prog_compiler_no_builtin_flag="$lt_prog_compiler_no_builtin_flag -
fno-rtti -fno-exceptions"
else
  :
fi

fi

lt_prog_compiler_wl=
lt_prog_compiler_pic=
lt_prog_compiler_static=

if test "$GCC" = yes; then
  lt_prog_compiler_wl='-Wl,'
  lt_prog_compiler_static='-static'

  case $host_os in
    aix*)
      # All AIX code is PIC.

```



```

    if test "$host_cpu" = ia64; then
    # AIX 5 now supports IA64 processor
    lt_prog_compiler_static='-Bstatic'
    fi
    ;;

amigaos*)
    case $host_cpu in
    powerpc)
        # see comment about AmigaOS4 .so support
        lt_prog_compiler_pic='-fPIC'
        ;;
    m68k)
        # FIXME: we need at least 68020 code to build shared
libraries, but
        # adding the '-m68020' flag to GCC prevents building
anything better,
        # like '-m68040'.
        lt_prog_compiler_pic='-m68020 -resident32 -malways-
restore-a4'
        ;;
    esac
    ;;

beos* | irix5* | irix6* | nonstopux* | osf3* | osf4* | osf5*)
    # PIC is the default for these OSes.
    ;;

mingw* | cygwin* | pw32* | os2* | cegcc*)
    # This hack is so that the source file can tell whether it is
being
    # built for inclusion in a dll (and should export symbols for
example).
    # Although the cygwin gcc ignores -fPIC, still need this for
old-style
    # (--disable-auto-import) libraries
    lt_prog_compiler_pic='-DLL_EXPORT'
    ;;

darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    lt_prog_compiler_pic='-fno-common'
    ;;

haiku*)
    # PIC is the default for Haiku.
    # The "-static" flag exists, but is broken.
    lt_prog_compiler_static=
    ;;

hpux*)

```

```

# PIC is the default for 64-bit PA HP-UX, but not for 32-bit
# PA HP-UX. On IA64 HP-UX, PIC is the default but the pic flag
# sets the default TLS model and affects inlining.
case $host_cpu in
hppa*64*)
# +Z the default
;;
*)
lt_prog_compiler_pic='-fPIC'
;;
esac
;;

interix[3-9]*)
# Interix 3.x gcc -fpic/-fPIC options generate broken code.
# Instead, we relocate shared libraries at runtime.
;;

msdosdjgpp*)
# Just because we use GCC doesn't mean we suddenly get shared
libraries
# on systems that don't support them.
lt_prog_compiler_can_build_shared=no
enable_shared=no
;;

*nto* | *qnx*)
# QNX uses GNU C++, but need to define -shared option too,
otherwise
# it will coredump.
lt_prog_compiler_pic='-fPIC -shared'
;;

sysv4*MP*)
if test -d /usr/nec; then
lt_prog_compiler_pic=-Kconform_pic
fi
;;

*)
lt_prog_compiler_pic='-fPIC'
;;
esac

case $cc_basename in
nvcc*) # Cuda Compiler Driver 2.2
lt_prog_compiler_wl='-Xlinker '
if test -n "$lt_prog_compiler_pic"; then
lt_prog_compiler_pic="-Xcompiler $lt_prog_compiler_pic"
fi
;;
esac

```

```

else
  # PORTME Check for flag to pass linker flags through the system
  compiler.
  case $host_os in
    aix*)
      lt_prog_compiler_wl='-Wl,'
      if test "$host_cpu" = ia64; then
        # AIX 5 now supports IA64 processor
        lt_prog_compiler_static='-Bstatic'
      else
        lt_prog_compiler_static='-bnso -bI:/lib/syscalls.exp'
      fi
      ;;

    mingw* | cygwin* | pw32* | os2* | cegcc*)
      # This hack is so that the source file can tell whether it is
      being
      # built for inclusion in a dll (and should export symbols for
      example).
      lt_prog_compiler_pic='-DDLL_EXPORT'
      ;;

    hpux9* | hpux10* | hpux11*)
      lt_prog_compiler_wl='-Wl,'
      # PIC is the default for IA64 HP-UX and 64-bit HP-UX, but
      # not for PA HP-UX.
      case $host_cpu in
        hppa*64*|ia64*)
          # +Z the default
          ;;
        *)
          lt_prog_compiler_pic='+Z'
          ;;
      esac
      # Is there a better lt_prog_compiler_static that works with the
      bundled CC?
      lt_prog_compiler_static='${wl}-a ${wl}archive'
      ;;

    irix5* | irix6* | nonstopux*)
      lt_prog_compiler_wl='-Wl,'
      # PIC (with -KPIC) is the default.
      lt_prog_compiler_static='-non_shared'
      ;;

    linux* | k*bsd*-gnu | kopensolaris*-gnu)
      case $cc_basename in
        # old Intel for x86_64 which still supported -KPIC.
        ecc*)
          lt_prog_compiler_wl='-Wl,'
          lt_prog_compiler_pic='-KPIC'
          lt_prog_compiler_static='-static'

```

```

    ;;
    # icc used to be incompatible with GCC.
    # ICC 10 doesn't accept -KPIC any more.
    icc* | ifort*)
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-fPIC'
lt_prog_compiler_static='-static'
    ;;
    # Lahey Fortran 8.1.
    lf95*)
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='--shared'
lt_prog_compiler_static='--static'
    ;;
    nagfor*)
    # NAG Fortran compiler
lt_prog_compiler_wl='-Wl,-Wl,,'
lt_prog_compiler_pic='-PIC'
lt_prog_compiler_static='-Bstatic'
    ;;
    pgcc* | pgf77* | pgf90* | pgf95* | pgfortran*)
    # Portland Group compilers (*not* the Pentium gcc compiler,
    # which looks to be a dead project)
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-fpic'
lt_prog_compiler_static='-Bstatic'
    ;;
    ccc*)
    lt_prog_compiler_wl='-Wl,'
    # All Alpha code is PIC.
    lt_prog_compiler_static='-non_shared'
    ;;
    xl* | bgxl* | bgf* | mpixl*)
    # IBM XL C 8.0/Fortran 10.1, 11.1 on PPC and BlueGene
lt_prog_compiler_wl='-Wl,'
lt_prog_compiler_pic='-qpik'
lt_prog_compiler_static='-qstaticlink'
    ;;
    *)
case `\$CC -V 2>&1 | sed 5q` in
  *Sun\ Ceres\ Fortran* | *Sun*Fortran*\ [1-7].* | *Sun*Fortran*\
8.[0-3]*)
    # Sun Fortran 8.3 passes all unrecognized flags to the linker
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    lt_prog_compiler_wl=''
    ;;
  *Sun\ F* | *Sun*Fortran*)
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    lt_prog_compiler_wl='-Qoption ld '
    ;;

```

```

*Sun\ C*)
  # Sun C 5.9
  lt_prog_compiler_pic='-KPIC'
  lt_prog_compiler_static='-Bstatic'
  lt_prog_compiler_wl='-Wl,'
  ;;
  *Intel*\ [CF]*Compiler*)
  lt_prog_compiler_wl='-Wl,'
  lt_prog_compiler_pic='-fPIC'
  lt_prog_compiler_static='-static'
  ;;
*Portland\ Group*)
  lt_prog_compiler_wl='-Wl,'
  lt_prog_compiler_pic='-fpic'
  lt_prog_compiler_static='-Bstatic'
  ;;
esac
;;
esac
;;

newsos6)
  lt_prog_compiler_pic='-KPIC'
  lt_prog_compiler_static='-Bstatic'
  ;;

*nto* | *qnx*)
  # QNX uses GNU C++, but need to define -shared option too,
otherwise
  # it will coredump.
  lt_prog_compiler_pic='-fPIC -shared'
  ;;

osf3* | osf4* | osf5*)
  lt_prog_compiler_wl='-Wl,'
  # All OSF/1 code is PIC.
  lt_prog_compiler_static='-non_shared'
  ;;

rdos*)
  lt_prog_compiler_static='-non_shared'
  ;;

solaris*)
  lt_prog_compiler_pic='-KPIC'
  lt_prog_compiler_static='-Bstatic'
  case $cc_basename in
  f77* | f90* | f95* | sunf77* | sunf90* | sunf95*)
  lt_prog_compiler_wl='-Qoption ld ';;
  *)
  lt_prog_compiler_wl='-Wl, ';;
  esac

```

```

;;

sunos4*)
    lt_prog_compiler_wl='-Qoption ld '
    lt_prog_compiler_pic='-PIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

sysv4 | sysv4.2uw2* | sysv4.3*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

sysv4*MP*)
    if test -d /usr/nec ;then
    lt_prog_compiler_pic='-Kconform_pic'
    lt_prog_compiler_static='-Bstatic'
    fi
    ;;

sysv5* | unixware* | sco3.2v5* | sco5v6* | OpenUNIX*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_pic='-KPIC'
    lt_prog_compiler_static='-Bstatic'
    ;;

unicos*)
    lt_prog_compiler_wl='-Wl,'
    lt_prog_compiler_can_build_shared=no
    ;;

uts4*)
    lt_prog_compiler_pic='-pic'
    lt_prog_compiler_static='-Bstatic'
    ;;

*)
    lt_prog_compiler_can_build_shared=no
    ;;
esac
fi

case $host_os in
# For platforms which do not support PIC, -DPIC is meaningless:
*djgpp*)
    lt_prog_compiler_pic=
    ;;
*)
    lt_prog_compiler_pic="$lt_prog_compiler_pic -DPIC"
    ;;
esac

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $compiler option
to produce PIC" >&5
$as_echo_n "checking for $compiler option to produce PIC... " >&6; }
if ${lt_cv_prog_compiler_pic+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_pic=$lt_prog_compiler_pic
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic" >&5
$as_echo "$lt_cv_prog_compiler_pic" >&6; }
lt_prog_compiler_pic=$lt_cv_prog_compiler_pic

#
# Check to make sure the PIC flag actually works.
#
if test -n "$lt_prog_compiler_pic"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler PIC
flag $lt_prog_compiler_pic works" >&5
$as_echo_n "checking if $compiler PIC flag $lt_prog_compiler_pic
works... " >&6; }
if ${lt_cv_prog_compiler_pic_works+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_pic_works=no
  ac_outfile=confptest.$ac_objext
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext
  lt_compiler_flag="$lt_prog_compiler_pic -DPIC"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag :; t' \
-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "`echo \"$as_me:$LINENO: $lt_compile\"`" >&5)
  (eval "$lt_compile" 2>confptest.err)
  ac_status=$?
  cat confptest.err >&5
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  if (exit $ac_status) && test -s "$ac_outfile"; then
    # The compiler can only warn and ignore the option if not
recognized
    # So say no if there are warnings other than the usual output.
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >confptest.exp
    $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2

```

```

        if test ! -s confptest.er2 || diff confptest.exp confptest.er2
>/dev/null; then
            lt_cv_prog_compiler_pic_works=yes
        fi
    fi
    $RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic_works" >&5
$as_echo "$lt_cv_prog_compiler_pic_works" >&6; }

if test x"$lt_cv_prog_compiler_pic_works" = xyes; then
    case $lt_prog_compiler_pic in
        "" | " "*) ;;
        *) lt_prog_compiler_pic="$lt_prog_compiler_pic" ;;
    esac
else
    lt_prog_compiler_pic=
    lt_prog_compiler_can_build_shared=no
fi

fi

#
# Check to make sure the static flag actually works.
#
wl=$lt_prog_compiler_wl eval
lt_tmp_static_flag="\$lt_prog_compiler_static\"
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler static
flag $lt_tmp_static_flag works" >&5
$as_echo_n "checking if $compiler static flag $lt_tmp_static_flag
works... " >&6; }
if ${lt_cv_prog_compiler_static_works+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_prog_compiler_static_works=no
    save_LDFLAGS="$LDFLAGS"
    LDFLAGS="$LDFLAGS $lt_tmp_static_flag"
    echo "$lt_simple_link_test_code" > confptest.$ac_ext
    if (eval $ac_link 2>confptest.err) && test -s confptest$ac_exeext;
then

```



```

# The linker can only warn and ignore the option if not
recognized
# So say no if there are warnings
if test -s confptest.err; then
  # Append any errors to the config.log.
  cat confptest.err 1>&5
  $ECHO "$_lt_linker_boilerplate" | $SED '/^$/d' > confptest.exp
  $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2
  if diff confptest.exp confptest.er2 >/dev/null; then
    lt_cv_prog_compiler_static_works=yes
  fi
else
  lt_cv_prog_compiler_static_works=yes
fi
fi
$RM -r confptest*
LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$_lt_cv_prog_compiler_static_works" >&5
$as_echo "$_lt_cv_prog_compiler_static_works" >&6; }

if test x"$lt_cv_prog_compiler_static_works" = xyes; then
:
else
  lt_prog_compiler_static=
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_c_o=no
  $RM -r confptest 2>/dev/null
  mkdir confptest
  cd confptest
  mkdir out
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext

  lt_compiler_flag="-o out/confptest2.$ac_objext"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.

```

```

# Note that $ac_compile itself does not contain backslashes and
begins
# with a dollar sign (not a hyphen), so the echo should work
correctly.
lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\}:&$lt_compiler_flag ;; t' \
-e 's: [^ ]*conftest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
(eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&5)
(eval "$lt_compile" 2>out/conftest.err)
ac_status=$?
cat out/conftest.err >&5
echo "$as_me:$LINENO: \$? = $ac_status" >&5
if (exit $ac_status) && test -s out/conftest2.$ac_objext
then
# The compiler can only warn and ignore the option if not
recognized
# So say no if there are warnings
$ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/conftest.exp
$SED '/^$/d; /^ *+/d' out/conftest.err >out/conftest.er2
if test ! -s out/conftest.er2 || diff out/conftest.exp
out/conftest.er2 >/dev/null; then
lt_cv_prog_compiler_c_o=yes
fi
fi
chmod u+w . 2>&5
$RM conftest*
# SGI C++ compiler will create directory out/ii_files/ for
# template instantiation
test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
$RM out/* && rmdir out
cd ..
$RM -r conftest
$RM conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o" >&5
$as_echo "$lt_cv_prog_compiler_c_o" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o+:} false; then :
$as_echo_n "(cached) " >&6

```

```

else
  lt_cv_prog_compiler_c_o=no
  $RM -r confptest 2>/dev/null
  mkdir confptest
  cd confptest
  mkdir out
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext

  lt_compiler_flag="-o out/confptest2.$ac_objext"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
  begins
  # with a dollar sign (not a hyphen), so the echo should work
  correctly.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\} :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "\"\`$as_me:$LINENO: $lt_compile\`" >&5)
  (eval "$lt_compile" 2>out/confptest.err)
  ac_status=$?
  cat out/confptest.err >&5
  echo "$as_me:$LINENO: ` $? = $ac_status`" >&5
  if (exit $ac_status) && test -s out/confptest2.$ac_objext
  then
    # The compiler can only warn and ignore the option if not
    recognized
    # So say no if there are warnings
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/confptest.exp
    $SED '/^$/d; /^ *+/d' out/confptest.err >out/confptest.er2
    if test ! -s out/confptest.er2 || diff out/confptest.exp
out/confptest.er2 >/dev/null; then
      lt_cv_prog_compiler_c_o=yes
    fi
  fi
  chmod u+w . 2>&5
  $RM confptest*
  # SGI C++ compiler will create directory out/ii_files/ for
  # template instantiation
  test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
  $RM out/* && rmdir out
  cd ..
  $RM -r confptest
  $RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o" >&5
$as_echo "$lt_cv_prog_compiler_c_o" >&6; }

```

```

hard_links="nottested"
if test "$lt_cv_prog_compiler_c_o" = no && test "$need_locks" != no;
then
  # do not overwrite the value of need_locks provided by the user
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking if we can lock
with hard links" >&5
$as_echo_n "checking if we can lock with hard links... " >&6; }
  hard_links=yes
  $RM conftest*
  ln conftest.a conftest.b 2>/dev/null && hard_links=no
  touch conftest.a
  ln conftest.a conftest.b 2>&5 || hard_links=no
  ln conftest.a conftest.b 2>/dev/null && hard_links=no
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $hard_links" >&5
$as_echo "$hard_links" >&6; }
  if test "$hard_links" = no; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: `gcc' does not
support `-c -o', so `make -j' may be unsafe" >&5
$as_echo "$as_me: WARNING: `gcc' does not support `-c -o', so `make
-j' may be unsafe" >&2;}
    need_locks=warn
  fi
else
  need_locks=no
fi

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the
$compiler linker ($LD) supports shared libraries" >&5
$as_echo_n "checking whether the $compiler linker ($LD) supports
shared libraries... " >&6; }

```

```

runpath_var=
allow_undefined_flag=
always_export_symbols=no
archive_cmds=
archive_expsym_cmds=
compiler_needs_object=no
enable_shared_with_static_runtimes=no
export_dynamic_flag_spec=
export_symbols_cmds='$NM $libobjs $convenience | $global_symbol_pipe
| $SED '\''s/.* //' | sort | uniq > $export_symbols'
hardcode_automatic=no
hardcode_direct=no
hardcode_direct_absolute=no

```

```

hardcode_libdir_flag_spec=
hardcode_libdir_separator=
hardcode_minus_L=no
hardcode_shlibpath_var=unsupported
inherit_rpath=no
link_all_deplibs=unknown
module_cmds=
module_expsym_cmds=
old_archive_from_new_cmds=
old_archive_from_expsyms_cmds=
thread_safe_flag_spec=
whole_archive_flag_spec=
# include_expsyms should be a list of space-separated symbols to be
*always*
# included in the symbol list
include_expsyms=
# exclude_expsyms can be an extended regexp of symbols to exclude
# it will be wrapped by ` (' and `)'$', so one must not match
beginning or
# end of line. Example: `a|bc|.*d.*' will exclude the symbols `a'
and `bc',
# as well as any symbol that contains `d'.
exclude_expsyms='_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*'
# Although _GLOBAL_OFFSET_TABLE_ is a valid symbol C name, most
a.out
# platforms (ab)use it in PIC code, but their linkers get confused
if
# the symbol is explicitly referenced. Since portable code cannot
# rely on this symbol name, it's probably fine to never include it
in
# preloaded symbol tables.
# Exclude shared library initialization/finalization symbols.
extract_expsyms_cmds=

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
# FIXME: the MSVC++ port hasn't been tested in a loooong time
# When not using gcc, we currently assume that we are using
# Microsoft Visual C++.
if test "$GCC" != yes; then
with_gnu_ld=no
fi
;;
interix*)
# we just hope/assume this is gcc and not c89 (= MSVC++)
with_gnu_ld=yes
;;
openbsd*)
with_gnu_ld=no
;;
esac

```

```

ld_shlibs=yes

# On some targets, GNU ld is compatible enough with the native
linker
# that we're better off using the native interface for both.
lt_use_gnu_ld_interface=no
if test "$with_gnu_ld" = yes; then
  case $host_os in
    aix*)
      # The AIX port of GNU ld has always aspired to compatibility
      # with the native linker. However, as the warning in the GNU ld
      # block says, versions before 2.19.5* couldn't really create
working
      # shared libraries, regardless of the interface used.
      case ` $LD -v 2>&1 ` in
        *\ (GNU\ Binutils)\ 2.19.5*) ;;
        *\ (GNU\ Binutils)\ 2.[2-9]*) ;;
        *\ (GNU\ Binutils)\ [3-9]*) ;;
        *)
          lt_use_gnu_ld_interface=yes
          ;;
      esac
      ;;
    *)
      lt_use_gnu_ld_interface=yes
      ;;
  esac
fi

if test "$lt_use_gnu_ld_interface" = yes; then
  # If archive_cmds runs LD, not CC, wlarc should be empty
  wlarc='${wl}'

  # Set some defaults for GNU ld with shared library support. These
  # are reset later if shared libraries are not supported. Putting
them
  # here allows them to be overridden if necessary.
  runpath_var=LD_RUN_PATH
  hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
  export_dynamic_flag_spec='${wl}--export-dynamic'
  # ancient GNU ld didn't support --whole-archive et. al.
  if $LD --help 2>&1 | $GREP 'no-whole-archive' > /dev/null; then
    whole_archive_flag_spec="$wlarc"--whole-archive$convenience
  "'$wlarc"--no-whole-archive'
  else
    whole_archive_flag_spec=
  fi
fi
supports_anon_versioning=no
case ` $LD -v 2>&1 ` in
  *GNU\ gold*) supports_anon_versioning=yes ;;
  *\ [01].* | *\ 2.[0-9].* | *\ 2.10.*) ;; # catch versions < 2.11
  *\ 2.11.93.0.2\ *) supports_anon_versioning=yes ;; # RH7.3 ...

```

```

    *\ 2.11.92.0.12\ *) supports_anon_versioning=yes ;; # Mandrake
8.2 ...
    *\ 2.11.*) ;; # other 2.11 versions
    *) supports_anon_versioning=yes ;;
esac

# See if GNU ld supports shared libraries.
case $host_os in
aix[3-9]*)
    # On AIX/PPC, the GNU linker is very broken
    if test "$host_cpu" != ia64; then
        ld_shlibs=no
        cat <<_LT_EOF 1>&2

*** Warning: the GNU linker, at least up to release 2.19, is reported
*** to be unable to reliably create shared libraries on AIX.
*** Therefore, libtool is disabling shared libraries support.  If you
*** really care for shared libraries, you may want to install binutils
*** 2.20 or above, or modify your PATH so that a non-GNU linker is
found.
*** You will then need to restart the configuration process.

_LT_EOF
    fi
    ;;

amigaos*)
    case $host_cpu in
powerpc)
        # see comment about AmigaOS4 .so support
        archive_cmds='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        archive_expsym_cmds=''
        ;;
m68k)
        archive_cmds='$RM $output_objdir/a2ixlibrary.data~$ECHO
"#define NAME $libname" > $output_objdir/a2ixlibrary.data~$ECHO
"#define LIBRARY_ID 1" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR
$AR_FLAGS $lib $libobjs~$RANLIB $lib~(cd $output_objdir && a2ixlibrary
-32)'
        hardcode_libdir_flag_spec='-L$libdir'
        hardcode_minus_L=yes
        ;;
    ;;
esac
    ;;

beos*)
    if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
        allow_undefined_flag=unsupported

```

```

# Joseph Beckenbach <jrb3@best.com> says some releases of gcc
# support --undefined. This deserves some investigation. FIXME
archive_cmds='$CC -nostart $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
else
ld_shlibs=no
fi
;;

cygwin* | mingw* | pw32* | cegcc*)
# _LT_TAGVAR(hardcode_libdir_flag_spec, ) is actually
meaningless,
# as there is no search path for DLLs.
hardcode_libdir_flag_spec='-L$libdir'
export_dynamic_flag_spec='${wl}--export-all-symbols'
allow_undefined_flag=unsupported
always_export_symbols=no
enable_shared_with_static_runtimes=yes
export_symbols_cmds='$NM $libobjs $convenience |
$global_symbol_pipe | $SED -e '\''/^([BCDGRS])[ ]/s/.*[ ]\([^\ ]*\)\//\1
DATA/;s/^\.*[ ]__nm_\([^\ ]*\)\[ ]\^[^\ ]*/\1 DATA/;^\^[AITW][
]/s/.* //'\' | sort | uniq > $export_symbols'

exclude_expsyms='[_]+GLOBAL_OFFSET_TABLE_|[_]+GLOBAL__[FID]_.*|[_]+hea
d_[A-Za-z0-9_]+_dll|[A-Za-z0-9_]+_dll_iname'

if $LD --help 2>&1 | $GREP 'auto-import' > /dev/null; then
archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags -o
$output_objdir/$soname ${wl}--enable-auto-image-base -Xlinker --out-
implib -Xlinker $lib'
# If the export-symbols file already is a .def file (1st line
# is EXPORTS), use it as is; otherwise, prepend...
archive_expsym_cmds='if test "x`$SED 1q $export_symbols`" =
xEXPORTS; then
cp $export_symbols $output_objdir/$soname.def;
else
echo EXPORTS > $output_objdir/$soname.def;
cat $export_symbols>> $output_objdir/$soname.def;
fi~
$CC -shared $output_objdir/$soname.def $libobjs $deplibs
$compiler_flags -o $output_objdir/$soname ${wl}--enable-auto-image-
base -Xlinker --out-implib -Xlinker $lib'
else
ld_shlibs=no
fi
;;

haiku*)
archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
link_all_deplibs=yes
;;

```



```

interix[3-9]*)
    hardcode_direct=no
    hardcode_shlibpath_var=no
    hardcode_libdir_flag_spec='${wl}-rpath,$libdir'
    export_dynamic_flag_spec='${wl}-E'
    # Hack: On Interix 3.x, we cannot compile PIC because of a
broken gcc.
    # Instead, shared libraries are loaded at an image base
(0x10000000 by
    # default) and relocated if they conflict, which is a slow very
memory
    # consuming and fragmenting process. To avoid this, we pick a
random,
    # 256 KiB-aligned image base between 0x50000000 and 0x6FFC0000
at link
    # time. Moving up from 0x10000000 also allows more sbrk(2)
space.
    archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--image-base,`expr ${RANDOM-$$} %
4096 / 2 \* 262144 + 1342177280` -o $lib'
    archive_expsym_cmds='sed "s,^,_" $export_symbols
>$output_objdir/$soname.expsym~$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--retain-symbols-
file,$output_objdir/$soname.expsym ${wl}--image-base,`expr ${RANDOM-
$$} % 4096 / 2 \* 262144 + 1342177280` -o $lib'
    ;;

gnu* | linux* | tpf* | k*bsd*-gnu | kopensolaris*-gnu)
    tmp_diet=no
    if test "$host_os" = linux-dietlibc; then
    case $cc_basename in
        diet\ *) tmp_diet=yes;; # linux-dietlibc with static linking
(!diet-dyn)
    esac
    fi
    if $LD --help 2>&1 | $EGREP ': supported targets:.* elf' >
/dev/null \
        && test "$tmp_diet" = no
    then
    tmp_addflag=' $pic_flag'
    tmp_sharedflag='-shared'
    case $cc_basename,$host_cpu in
        pgcc*) # Portland Group C compiler
            whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`; do test -n \"$conv\" &&
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` ${wl}--no-whole-archive'
            tmp_addflag=' $pic_flag'
            ;;
        pgf77* | pgf90* | pgf95* | pgfortran*)
            # Portland Group f77 and f90 compilers

```

```

        whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`; do test -n \"$conv\" &&
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` ${wl}--no-whole-archive'
        tmp_addflag=' $pic_flag -Mnomain' ;;
        ecc*,ia64* | icc*,ia64*) # Intel C compiler on ia64
        tmp_addflag=' -i_dynamic' ;;
        efc*,ia64* | ifort*,ia64*) # Intel Fortran compiler on ia64
        tmp_addflag=' -i_dynamic -nofor_main' ;;
        ifc* | ifort*) # Intel Fortran compiler
        tmp_addflag=' -nofor_main' ;;
        lf95*) # Lahey Fortran 8.1
        whole_archive_flag_spec=
        tmp_sharedflag='--shared' ;;
        xl[cC]* | bgxl[cC]* | mpixl[cC]*) # IBM XL C 8.0 on PPC (deal
with xlf below)
        tmp_sharedflag='-qmkshrojb'
        tmp_addflag= ;;
        nvcc*) # Cuda Compiler Driver 2.2
        whole_archive_flag_spec='${wl}--whole-archive`for conv in
$convenience\`; do test -n \"$conv\" &&
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` ${wl}--no-whole-archive'
        compiler_needs_object=yes
        ;;
    esac
    case ` $CC -V 2>&1 | sed 5q` in
    *Sun\ C*) # Sun C 5.9
        whole_archive_flag_spec='${wl}--whole-archive`new_convenience=;
for conv in $convenience\`; do test -z \"$conv\" ||
new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` ${wl}--no-whole-archive'
        compiler_needs_object=yes
        tmp_sharedflag='-G' ;;
    *Sun\ F*) # Sun Fortran 8.3
        tmp_sharedflag='-G' ;;
    esac
    archive_cmds='$CC "'$tmp_sharedflag"'$tmp_addflag"' $libobjs
$deplibs $compiler_flags ${wl}-soname $wl$soname -o $lib'

    if test "x$supports_anon_versioning" = xyes; then
        archive_expsym_cmds='echo "{ global:" >
$output_objdir/$libname.ver~
cat $export_symbols | sed -e "s/(.*)/\1;/ " >>
$output_objdir/$libname.ver~
echo "local: *; };" >> $output_objdir/$libname.ver~
$CC "'$tmp_sharedflag"'$tmp_addflag"' $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-version-script
${wl}$output_objdir/$libname.ver -o $lib'
    fi

    case $cc_basename in

```

```

    xlf* | bgf* | bgxlf* | mpixlf*)
    # IBM XL Fortran 10.1 on PPC cannot create shared libs itself
    whole_archive_flag_spec='--whole-archive$convenience --no-
whole-archive'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
    archive_cmds='$LD -shared $libobjs $deplibs $linker_flags -
soname $soname -o $lib'
    if test "x$supports_anon_versioning" = xyes; then
        archive_expsym_cmds='echo "{ global:" >
$output_objdir/$libname.ver~
        cat $export_symbols | sed -e "s/(.*)/\1;/ " >>
$output_objdir/$libname.ver~
        echo "local: *; };" >> $output_objdir/$libname.ver~
        $LD -shared $libobjs $deplibs $linker_flags -soname $soname
-version-script $output_objdir/$libname.ver -o $lib'
    fi
    ;;
esac
else
    ld_shlibs=no
fi
;;

netbsd*)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
        archive_cmds='$LD -Bshareable $libobjs $deplibs $linker_flags -o
$lib'
        wlarc=
    else
        archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
    fi
    ;;

solaris*)
    if $LD -v 2>&1 | $GREP 'BFD 2\.8' > /dev/null; then
        ld_shlibs=no
        cat <<_LT_EOF 1>&2

*** Warning: The releases 2.8.* of the GNU linker cannot reliably
*** create shared libraries on Solaris systems.  Therefore, libtool
*** is disabling shared libraries support.  We urge you to upgrade GNU
*** binutils to release 2.9.1 or newer.  Another option is to modify
*** your PATH or compiler configuration so that the native linker is
*** used, and then restart.

_LT_EOF
        elif $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then

```

```

        archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
        else
        ld_shlibs=no
        fi
        ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX*)
        case ` $LD -v 2>&1 ` in
                *\ [01].* | *\ 2.[0-9].* | *\ 2.1[0-5].*)
                ld_shlibs=no
                cat <<_LT_EOF 1>&2

*** Warning: Releases of the GNU linker prior to 2.16.91.0.3 can not
*** reliably create shared libraries on SCO systems.  Therefore,
libtool
*** is disabling shared libraries support.  We urge you to upgrade GNU
*** binutils to release 2.16.91.0.3 or newer.  Another option is to
modify
*** your PATH or compiler configuration so that the native linker is
*** used, and then restart.

_LT_EOF
        ;;
        *)
        # For security reasons, it is highly recommended that you
always
        # use absolute paths for naming shared libraries, and exclude
the
        # DT_RUNPATH tag from executables and libraries.  But doing so
        # requires that you compile everything twice, which is a pain.
        if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
                hardcoded_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
                archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags
${wl}-soname $wl$soname -o $lib'
                archive_expsym_cmds='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
        else
                ld_shlibs=no
        fi
        ;;
        esac
        ;;

sunos4*)
        archive_cmds='$LD -assert pure-text -Bshareable -o $lib $libobjs
$deplibs $linker_flags'

```

```

    wlarc=
    hardcode_direct=yes
    hardcode_shlibpath_var=no
    ;;

*)
    if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
        archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-file
$wl$export_symbols -o $lib'
        else
            ld_shlibs=no
        fi
    ;;
esac

if test "$ld_shlibs" = no; then
    runpath_var=
    hardcode_libdir_flag_spec=
    export_dynamic_flag_spec=
    whole_archive_flag_spec=
fi
else
    # PORTME fill in a description of your system's linker (not GNU
ld)
    case $host_os in
    aix3*)
        allow_undefined_flag=unsupported
        always_export_symbols=yes
        archive_expsym_cmds='$LD -o $output_objdir/$soname $libobjs
$deplibs $linker_flags -bE:$export_symbols -T512 -H512 -bM:SRE~$AR
$AR_FLAGS $lib $output_objdir/$soname'
        # Note: this linker hardcodes the directories in LIBPATH if
there
        # are no directories specified by -L.
        hardcode_minus_L=yes
        if test "$GCC" = yes && test -z "$lt_prog_compiler_static"; then
            # Neither direct hardcoding nor static linking is supported with
a
            # broken collect2.
            hardcode_direct=unsupported
        fi
    ;;
    aix[4-9]*)
        if test "$host_cpu" = ia64; then
            # On IA64, the linker does run time linking by default, so we
don't
            # have to do anything special.

```

```

aix_use_runtimelinking=no
exp_sym_flag='-Bexport'
no_entry_flag=""
else
# If we're using GNU nm, then we don't want the "-C" option.
# -C means demangle to AIX nm, but means don't demangle with GNU
nm
# Also, AIX nm treats weak defined symbols like other global
# defined symbols, whereas GNU nm marks them as "W".
if $NM -V 2>&1 | $GREP 'GNU' > /dev/null; then
    export_symbols_cmds='$NM -Bpg $libobjs $convenience | awk '\''{
if (((\ $ 2 == "T") || (\ $ 2 == "D") || (\ $ 2 == "B") || (\ $ 2 == "W"))
&& (substr(\ $ 3,1,1) != ".")) { print \ $ 3 } }'\'' | sort -u >
$export_symbols'
else
    export_symbols_cmds='$NM -BCpg $libobjs $convenience | awk
'\''{ if (((\ $ 2 == "T") || (\ $ 2 == "D") || (\ $ 2 == "B")) &&
(substr(\ $ 3,1,1) != ".")) { print \ $ 3 } }'\'' | sort -u >
$export_symbols'
fi
aix_use_runtimelinking=no

# Test if we are trying to use run time linking or normal
# AIX style linking. If -brtl is somewhere in LDFLAGS, we
# need to do runtime linking.
case $host_os in aix4.[23]|aix4.[23].*|aix[5-9]*)
    for ld_flag in $LDFLAGS; do
        if (test $ld_flag = "-brtl" || test $ld_flag = "-Wl,-brtl");
then
            aix_use_runtimelinking=yes
            break
        fi
    done
    ;;
esac

exp_sym_flag='-bexport'
no_entry_flag='-bnoentry'
fi

# When large executables or shared objects are built, AIX ld can
# have problems creating the table of contents. If linking a
library
# or program results in "error TOC overflow" add -mminimal-toc
to
# CXXFLAGS/CFLAGS for g++/gcc. In the cases where that is not
# enough to fix the problem, add -Wl,-bbigtoc to LDFLAGS.

archive_cmds=''
hardcode_direct=yes
hardcode_direct_absolute=yes
hardcode_libdir_separator=':'

```

```

link_all_deplibs=yes
file_list_spec='${wl}-f, '

if test "$GCC" = yes; then
case $host_os in aix4.[012]|aix4.[012].*)
# We only want to do this on AIX 4.2 and lower, the check
# below for broken collect2 doesn't work under 4.3+
collect2name=`${CC} -print-prog-name=collect2`
if test -f "$collect2name" &&
strings "$collect2name" | $GREP resolve_lib_name >/dev/null
then
# We have reworked collect2
:
else
# We have old collect2
hardcode_direct=unsupported
# It fails to find uninstalled libraries when the uninstalled
# path is not listed in the libpath. Setting hardcode_minus_L
# to unsupported forces relinking
hardcode_minus_L=yes
hardcode_libdir_flag_spec='-L$libdir'
hardcode_libdir_separator=
fi
;;
esac
shared_flag='-shared'
if test "$aix_use_runtimelinking" = yes; then
shared_flag="$shared_flag ''${wl}-G'
fi
else
# not using gcc
if test "$host_cpu" = ia64; then
# VisualAge C++, Version 5.5 for AIX 5L for IA-64, Beta 3 Release
# chokes on -Wl,-G. The following line is correct:
shared_flag='-G'
else
if test "$aix_use_runtimelinking" = yes; then
shared_flag='${wl}-G'
else
shared_flag='${wl}-bM:SRE'
fi
fi
fi

export_dynamic_flag_spec='${wl}-bexpall'
# It seems that -bexpall does not export symbols beginning with
# underscore (_), so it is better to generate a list of symbols
to export.
always_export_symbols=yes
if test "$aix_use_runtimelinking" = yes; then
# Warning - without using the other runtime loading flags (-
brtl),

```

```

        # -berok will link without error, but may produce a broken
library.
        allow_undefined_flag='-berok'
        # Determine the default libpath from the value encoded in an
        # empty executable.
        if test "${lt_cv_aix_libpath+set}" = set; then
aix_libpath=$lt_cv_aix_libpath
else
        if ${lt_cv_aix_libpath_+;} false; then :
        $as_echo_n "(cached) " >&6
else
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :

    lt_aix_libpath_sed='
        /Import File Strings/,/^$/ {
            /^0/ {
                s/^0 *\[^\]*\ *$/\1/
                p
            }
        }'
    lt_cv_aix_libpath_=`dump -H conftest$ac_exeext 2>/dev/null | $SED -n
-e "$lt_aix_libpath_sed"`
    # Check for a 64-bit object if we didn't find anything.
    if test -z "$lt_cv_aix_libpath_"; then
        lt_cv_aix_libpath_=`dump -HX64 conftest$ac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
    fi
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
if test -z "$lt_cv_aix_libpath_"; then
    lt_cv_aix_libpath_="/usr/lib:/lib"
fi

fi

aix_libpath=$lt_cv_aix_libpath_
fi

        hardcoded_libdir_flag_spec='${wl}-
bllibpath:$libdir:'"$aix_libpath"

```



```

        archive_expsym_cmds='$CC -o $output_objdir/$soname $libobjs
$deplibs '"'\${wl}$no_entry_flag"' $compiler_flags `if test
"x${allow_undefined_flag}" != "x"; then func_echo_all
"${wl}${allow_undefined_flag}"; else ;; fi`
'"'\${wl}$exp_sym_flag:\$export_symbols $shared_flag"
        else
        if test "$host_cpu" = ia64; then
            hardcode_libdir_flag_spec='${wl}-R $libdir:/usr/lib:/lib'
            allow_undefined_flag="-z nodefs"
            archive_expsym_cmds="\$CC $shared_flag" -o
$output_objdir/$soname $libobjs $deplibs '"'\${wl}$no_entry_flag"'
$compiler_flags ${wl}${allow_undefined_flag}
'"'\${wl}$exp_sym_flag:\$export_symbols"
        else
            # Determine the default libpath from the value encoded in an
            # empty executable.
            if test "${lt_cv_aix_libpath+set}" = set; then
                aix_libpath=$lt_cv_aix_libpath
            else
                if ${lt_cv_aix_libpath_+} false; then :
                    $as_echo_n "(cached) " >&6
                else
                    cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :

    lt_aix_libpath_sed='
        /Import File Strings/,/^$/ {
            /^0/ {
                s/^0 *\[^\]*\)* *$/\1/
                P
            }
        }'
    lt_cv_aix_libpath_=`dump -H conftest$ac_exeext 2>/dev/null | $SED -n
-e "$lt_aix_libpath_sed"`
    # Check for a 64-bit object if we didn't find anything.
    if test -z "$lt_cv_aix_libpath_"; then
        lt_cv_aix_libpath_=`dump -HX64 conftest$ac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
    fi
fi
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext

```

```

if test -z "$lt_cv_aix_libpath_"; then
  lt_cv_aix_libpath_="/usr/lib:/lib"
fi

fi

aix_libpath=$lt_cv_aix_libpath_
fi

  hardcode_libdir_flag_spec='${wl}-
bllibpath:$libdir:"$aix_libpath"
  # Warning - without using the other run time loading flags,
  # -berok will link without error, but may produce a broken
library.
  no_undefined_flag=' ${wl}-bernotok'
  allow_undefined_flag=' ${wl}-berok'
  if test "$with_gnu_ld" = yes; then
    # We only use this code for GNU lds that support --whole-
archive.
    whole_archive_flag_spec='${wl}--whole-archive$convenience
${wl}--no-whole-archive'
  else
    # Exported symbols can be pulled into shared objects from
archives
    whole_archive_flag_spec='$convenience'
  fi
  archive_cmds_need_lc=yes
  # This is similar to how AIX traditionally builds its shared
libraries.
  archive_expsym_cmds="\$CC $shared_flag" -o
$output_objdir/$soname $libobjs $deplibs ${wl}-bnoentry
$compiler_flags ${wl}-bE:$export_symbols${allow_undefined_flag}~$AR
$AR_FLAGS $output_objdir/$libname$release.a $output_objdir/$soname'
  fi
  fi
  ;;

amigaos*)
  case $host_cpu in
    powerpc)
      # see comment about AmigaOS4 .so support
      archive_cmds='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
      archive_expsym_cmds=''
    ;;
    m68k)
      archive_cmds='$RM $output_objdir/a2ixlibrary.data~$ECHO
#define NAME $libname" > $output_objdir/a2ixlibrary.data~$ECHO
#define LIBRARY_ID 1" >> $output_objdir/a2ixlibrary.data~$ECHO
#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR

```

```

$AR_FLAGS $lib $libobjs~$RANLIB $lib~(cd $output_objdir && a2ixlibrary
-32)'
        hardcode_libdir_flag_spec='-L$libdir'
        hardcode_minus_L=yes
        ;;
    esac
    ;;

bsdi[45]*)
    export_dynamic_flag_spec=-rdynamic
    ;;

cygwin* | mingw* | pw32* | cegcc*)
    # When not using gcc, we currently assume that we are using
    # Microsoft Visual C++.
    # hardcode_libdir_flag_spec is actually meaningless, as there is
    # no search path for DLLs.
    case $cc_basename in
    cl*)
        # Native MSVC
        hardcode_libdir_flag_spec=' '
        allow_undefined_flag=unsupported
        always_export_symbols=yes
        file_list_spec='@'
        # Tell ltmain to make .lib files, not .a files.
        libext=lib
        # Tell ltmain to make .dll files, not .so files.
        shrext_cmds=".dll"
        # FIXME: Setting linknames here is a bad hack.
        archive_cmds='$CC -o $output_objdir/$soname $libobjs
$compiler_flags $deplibs -Wl,-dll~linknames='
        archive_expsym_cmds='if test "x`$SED 1q $export_symbols`" =
xEXPORTS; then
            sed -n -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -EXPORT:\\\\\\\\\\\\\\\\1/' -
e '1\\\\\\\\!p' < $export_symbols > $output_objdir/$soname.exp;
        else
            sed -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -EXPORT:\\\\\\\\\\\\\\\\1/' <
$export_symbols > $output_objdir/$soname.exp;
        fi~
        $CC -o $tool_output_objdir$soname $libobjs $compiler_flags
$deplibs "@$tool_output_objdir$soname.exp" -Wl,-DLL,-
IMPLIB:"$tool_output_objdir$libname.dll.lib"~
        linknames='
    # The linker will not automatically build a static lib if we
    build a DLL.
    # _LT_TAGVAR(old_archive_from_new_cmds, )='true'
    enable_shared_with_static_runtimes=yes
    exclude_expsyms='_NULL_IMPORT_DESCRIPTOR|_IMPORT_DESCRIPTOR_.*'
    export_symbols_cmds='$NM $libobjs $convenience |
$global_symbol_pipe | $SED -e '\''/^([BCDGRS])[ ]/s/.*/[ ]\([^\
]*\)\/\1,DATA/'\'' | $SED -e '\''/^([AITW])[ ]/s/.*/[ ]//'\'' | sort |
uniq > $export_symbols'

```

```

# Don't use ranlib
old_postinstall_cmds='chmod 644 $oldlib'
postlink_cmds='lt_outputfile="@OUTPUT@"~
  lt_tool_outputfile="@TOOL_OUTPUT@"~
  case $lt_outputfile in
    *.exe|*.EXE) ;;
    *)
      lt_outputfile="$lt_outputfile.exe"
      lt_tool_outputfile="$lt_tool_outputfile.exe"
      ;;
  esac~
  if test "$MANIFEST_TOOL" != ":" && test -f
"$lt_outputfile.manifest"; then
    $MANIFEST_TOOL -manifest "$lt_tool_outputfile.manifest" -
outputresource:"$lt_tool_outputfile" || exit 1;
    $RM "$lt_outputfile.manifest";
  fi'
;;
*)
# Assume MSVC wrapper
hardcode_libdir_flag_spec=' '
allow_undefined_flag=unsupported
# Tell ltmain to make .lib files, not .a files.
libext=lib
# Tell ltmain to make .dll files, not .so files.
shrext_cmds=".dll"
# FIXME: Setting linknames here is a bad hack.
archive_cmds='$CC -o $lib $libobjs $compiler_flags `func_echo_all
"$deplibs" | $SED \'\'s/ -lc$//\'\'` -link -dll~linknames='
# The linker will automatically build a .lib file if we build a
DLL.
old_archive_from_new_cmds='true'
# FIXME: Should let the user specify the lib program.
old_archive_cmds='lib -OUT:$oldlib$oldobjs$old_deplibs'
enable_shared_with_static_runtimes=yes
;;
esac
;;

darwin* | rhapsody*)

archive_cmds_need_lc=no
hardcode_direct=no
hardcode_automatic=yes
hardcode_shlibpath_var=unsupported
if test "$lt_cv_ld_force_load" = "yes"; then
  whole_archive_flag_spec='`for conv in $convenience\``"; do test -
n \"$conv\" && new_convenience=\"$new_convenience ${wl}-
force_load,$conv\`; done; func_echo_all \"$new_convenience\``'
else

```

```

    whole_archive_flag_spec=''
fi
link_all_deplibs=yes
allow_undefined_flag="$_lt_dar_allow_undefined"
case $cc_basename in
    ifort*) _lt_dar_can_shared=yes ;;
    *) _lt_dar_can_shared=$GCC ;;
esac
if test "$_lt_dar_can_shared" = "yes"; then
    output_verbose_link_cmd=func_echo_all
    archive_cmds="\$CC -dynamiclib \$allow_undefined_flag -o \$lib
\$libobjs \$deplibs \$compiler_flags -install_name \$rpath/\$soname
\$verstring $_lt_dar_single_mod${_lt_dsymutil}"
    module_cmds="\$CC \$allow_undefined_flag -o \$lib -bundle
\$libobjs \$deplibs \$compiler_flags${_lt_dsymutil}"
    archive_expsym_cmds="sed 's,^,_, ' < \$export_symbols >
\$output_objdir/\${libname}-symbols.expsym~\$CC -dynamiclib
\$allow_undefined_flag -o \$lib \$libobjs \$deplibs \$compiler_flags -
install_name \$rpath/\$soname \$verstring
${_lt_dar_single_mod}${_lt_dar_export_syms}${_lt_dsymutil}"
    module_expsym_cmds="sed -e 's,^,_, ' < \$export_symbols >
\$output_objdir/\${libname}-symbols.expsym~\$CC \$allow_undefined_flag
-o \$lib -bundle \$libobjs \$deplibs
\$compiler_flags${_lt_dar_export_syms}${_lt_dsymutil}"

else
    ld_shlibs=no
fi

;;

dgux*)
    archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_libdir_flag_spec='-L$libdir'
    hardcode_shlibpath_var=no
;;

# FreeBSD 2.2.[012] allows us to include c++rt0.o to get C++
constructor
# support.  Future versions do this automatically, but an explicit
c++rt0.o
# does not break anything, and helps significantly (at the cost of
a little
# extra space).
freebsd2.2*)
    archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags /usr/lib/c++rt0.o'
    hardcode_libdir_flag_spec='-R$libdir'
    hardcode_direct=yes
    hardcode_shlibpath_var=no
;;

```

```

# Unfortunately, older versions of FreeBSD 2 do not have this
feature.
freebsd2.*)
    archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_direct=yes
    hardcode_minus_L=yes
    hardcode_shlibpath_var=no
    ;;

# FreeBSD 3 and greater uses gcc -shared to do shared libraries.
freebsd* | dragonfly*)
    archive_cmds='$CC -shared $pic_flag -o $lib $libobjs $deplibs
$compiler_flags'
    hardcode_libdir_flag_spec='-R$libdir'
    hardcode_direct=yes
    hardcode_shlibpath_var=no
    ;;

hpux9*)
    if test "$GCC" = yes; then
        archive_cmds='$RM $output_objdir/$soname~$CC -shared $pic_flag
${wl}+b ${wl}$install_libdir -o $output_objdir/$soname $libobjs
$deplibs $compiler_flags~test $output_objdir/$soname = $lib || mv
$output_objdir/$soname $lib'
    else
        archive_cmds='$RM $output_objdir/$soname~$LD -b +b
$install_libdir -o $output_objdir/$soname $libobjs $deplibs
$linker_flags~test $output_objdir/$soname = $lib || mv
$output_objdir/$soname $lib'
    fi
    hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'
    hardcode_libdir_separator=:
    hardcode_direct=yes

    # hardcode_minus_L: Not really in the search PATH,
    # but as the default location of the library.
    hardcode_minus_L=yes
    export_dynamic_flag_spec='${wl}-E'
    ;;

hpux10*)
    if test "$GCC" = yes && test "$with_gnu_ld" = no; then
        archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname ${wl}+b
${wl}$install_libdir -o $lib $libobjs $deplibs $compiler_flags'
    else
        archive_cmds='$LD -b +h $soname +b $install_libdir -o $lib
$libobjs $deplibs $linker_flags'
    fi
    if test "$with_gnu_ld" = no; then
        hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'

```

```

hardcode_libdir_separator=:
hardcode_direct=yes
hardcode_direct_absolute=yes
export_dynamic_flag_spec='${wl}-E'
# hardcode_minus_L: Not really in the search PATH,
# but as the default location of the library.
hardcode_minus_L=yes
fi
;;

hpux11*)
if test "$GCC" = yes && test "$with_gnu_ld" = no; then
case $host_cpu in
hppa*64*)
archive_cmds='$CC -shared ${wl}+h ${wl}$soname -o $lib $libobjs
$deplibs $compiler_flags'
;;
ia64*)
archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname
${wl}+nodefaulttrpath -o $lib $libobjs $deplibs $compiler_flags'
;;
*)
archive_cmds='$CC -shared $pic_flag ${wl}+h ${wl}$soname
${wl}+b ${wl}$install_libdir -o $lib $libobjs $deplibs
$compiler_flags'
;;
esac
else
case $host_cpu in
hppa*64*)
archive_cmds='$CC -b ${wl}+h ${wl}$soname -o $lib $libobjs
$deplibs $compiler_flags'
;;
ia64*)
archive_cmds='$CC -b ${wl}+h ${wl}$soname ${wl}+nodefaulttrpath
-o $lib $libobjs $deplibs $compiler_flags'
;;
*)
# Older versions of the 11.00 compiler do not understand -b yet
# (HP92453-01 A.11.01.20 doesn't, HP92453-01 B.11.X.35175-
35176.GP does)
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $CC
understands -b" >&5
$as_echo_n "checking if $CC understands -b... " >&6; }
if ${lt_cv_prog_compiler__b+:} false; then :
$as_echo_n "(cached) " >&6
else
lt_cv_prog_compiler__b=no
save_LDFLAGS="$LDFLAGS"
LDFLAGS="$LDFLAGS -b"
echo "$lt_simple_link_test_code" > conftest.$ac_ext

```

```

    if (eval $ac_link 2>confptest.err) && test -s confptest$ac_exeext;
then
    # The linker can only warn and ignore the option if not
recognized
    # So say no if there are warnings
    if test -s confptest.err; then
        # Append any errors to the config.log.
        cat confptest.err 1>&5
        $ECHO "$_lt_linker_boilerplate" | $SED '/^$/d' > confptest.exp
        $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2
        if diff confptest.exp confptest.er2 >/dev/null; then
            lt_cv_prog_compiler__b=yes
        fi
    else
        lt_cv_prog_compiler__b=yes
    fi
fi
$RM -r confptest*
LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler__b" >&5
$as_echo "$lt_cv_prog_compiler__b" >&6; }

if test x"$lt_cv_prog_compiler__b" = xyes; then
    archive_cmds='$CC -b ${wl}+h ${wl}$soname ${wl}+b
${wl}$install_libdir -o $lib $libobjs $deplibs $compiler_flags'
else
    archive_cmds='$LD -b +h $soname +b $install_libdir -o $lib
$libobjs $deplibs $linker_flags'
fi

    ;;
esac
fi
if test "$with_gnu_ld" = no; then
hardcode_libdir_flag_spec='${wl}+b ${wl}$libdir'
hardcode_libdir_separator=:

case $host_cpu in
hppa*64*|ia64*)
    hardcode_direct=no
    hardcode_shlibpath_var=no
    ;;
*)
    hardcode_direct=yes
    hardcode_direct_absolute=yes
    export_dynamic_flag_spec='${wl}-E'

    # hardcode_minus_L: Not really in the search PATH,
    # but as the default location of the library.

```



```

        hardcode_minus_L=yes
        ;;
    esac
    fi
    ;;

    irix5* | irix6* | nonstopux*)
        if test "$GCC" = yes; then
            archive_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname ${wl}$soname `test -n "$verstring" &&
func_echo_all "${wl}-set_version ${wl}$verstring"` ${wl}-
update_registry ${wl}${output_objdir}/so_locations -o $lib'
            # Try to use the -exported_symbol ld option, if it does not
            # work, assume that -exports_file does not work either and
            # implicitly export all symbols.
            # This should be the same for all languages, so no per-tag cache
            variable.
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the
$host_os linker accepts -exported_symbol" >&5
$as_echo_n "checking whether the $host_os linker accepts -
exported_symbol... " >&6; }
            if ${lt_cv_irix_exported_symbol+:} false; then :
                $as_echo_n "(cached) " >&6
            else
                save_LDFLAGS="$LDFLAGS"
                LDFLAGS="$LDFLAGS -shared ${wl}-exported_symbol ${wl}foo
${wl}-update_registry ${wl}/dev/null"
                cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */
int foo (void) { return 0; }
_ACEOF
            if ac_fn_c_try_link "$LINENO"; then :
                lt_cv_irix_exported_symbol=yes
            else
                lt_cv_irix_exported_symbol=no
            fi
            rm -f core conftest.err conftest.$ac_objext \
                conftest$ac_exeext conftest.$ac_ext
                LDFLAGS="$save_LDFLAGS"
            fi
            { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_irix_exported_symbol" >&5
$as_echo "$lt_cv_irix_exported_symbol" >&6; }
            if test "$lt_cv_irix_exported_symbol" = yes; then
                archive_expsym_cmds='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-soname ${wl}$soname `test -n "$verstring" &&
func_echo_all "${wl}-set_version ${wl}$verstring"` ${wl}-
update_registry ${wl}${output_objdir}/so_locations ${wl}-exports_file
${wl}$export_symbols -o $lib'
                fi
            else

```

```

        archive_cmds='$CC -shared $libobjs $deplibs $compiler_flags -
soname $soname `test -n "$verstring" && func_echo_all "-set_version
$verstring"` -update_registry ${output_objdir}/so_locations -o $lib'
        archive_expsym_cmds='$CC -shared $libobjs $deplibs
$compiler_flags -soname $soname `test -n "$verstring" && func_echo_all
"-set_version $verstring"` -update_registry
${output_objdir}/so_locations -exports_file $export_symbols -o $lib'
        fi
        archive_cmds_need_lc='no'
        hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
        hardcode_libdir_separator=:
        inherit_rpath=yes
        link_all_deplibs=yes
        ;;

netbsd*)
        if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
                archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags' # a.out
        else
                archive_cmds='$LD -shared -o $lib $libobjs $deplibs
$linker_flags' # ELF
        fi
        hardcode_libdir_flag_spec='-R$libdir'
        hardcode_direct=yes
        hardcode_shlibpath_var=no
        ;;

newsos6)
        archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
        hardcode_direct=yes
        hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
        hardcode_libdir_separator=:
        hardcode_shlibpath_var=no
        ;;

*nto* | *qnx*)
        ;;

openbsd*)
        if test -f /usr/libexec/ld.so; then
                hardcode_direct=yes
                hardcode_shlibpath_var=no
                hardcode_direct_absolute=yes
                if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
                        archive_cmds='$CC -shared $pic_flag -o $lib $libobjs $deplibs
$compiler_flags'
                        archive_expsym_cmds='$CC -shared $pic_flag -o $lib $libobjs
$deplibs $compiler_flags ${wl}-retain-symbols-file,$export_symbols'
                        hardcode_libdir_flag_spec='${wl}-rpath,$libdir'

```

```

        export_dynamic_flag_spec='${wl}-E'
    else
        case $host_os in
            openbsd[01].* | openbsd2.[0-7] | openbsd2.[0-7].*)
                archive_cmds='$LD -Bshareable -o $lib $libobjs $deplibs
$linker_flags'
                hardcode_libdir_flag_spec='-R$libdir'
                ;;
            *)
                archive_cmds='$CC -shared $pic_flag -o $lib $libobjs
$deplibs $compiler_flags'
                hardcode_libdir_flag_spec='${wl}-rpath,$libdir'
                ;;
        esac
    fi
else
    ld_shlibs=no
fi
;;

os2*)
    hardcode_libdir_flag_spec='-L$libdir'
    hardcode_minus_L=yes
    allow_undefined_flag=unsupported
    archive_cmds='$ECHO "LIBRARY $libname INITINSTANCE" >
$output_objdir/$libname.def~$ECHO "DESCRIPTION \"$libname\"" >>
$output_objdir/$libname.def~echo DATA >>
$output_objdir/$libname.def~echo " SINGLE NONSHARED" >>
$output_objdir/$libname.def~echo EXPORTS >>
$output_objdir/$libname.def~emxexp $libobjs >>
$output_objdir/$libname.def~$CC -Zdll -Zcrtdll -o $lib $libobjs
$deplibs $compiler_flags $output_objdir/$libname.def'
    old_archive_from_new_cmds='emximp -o $output_objdir/$libname.a
$output_objdir/$libname.def'
    ;;

osf3*)
    if test "$GCC" = yes; then
        allow_undefined_flag='${wl}-expect_unresolved ${wl}\*'
        archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags ${wl}-soname ${wl}$soname `test -n
"$sverstring" && func_echo_all "${wl}-set_version ${wl}$sverstring"`
${wl}-update_registry ${wl}${output_objdir}/so_locations -o $lib'
    else
        allow_undefined_flag=' -expect_unresolved \*'
        archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags -soname $soname `test -n "$sverstring" &&
func_echo_all "-set_version $sverstring"` -update_registry
${output_objdir}/so_locations -o $lib'
    fi
    archive_cmds_need_lc='no'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'

```

```

hardcode_libdir_separator=:
;;

osf4* | osf5*)    # as osf3* with the addition of -msym flag
  if test "$GCC" = yes; then
    allow_undefined_flag=' ${wl}-expect_unresolved ${wl}\*'
    archive_cmds='$CC -shared${allow_undefined_flag} $pic_flag
$libobjs $deplibs $compiler_flags ${wl}-msym ${wl}-soname ${wl}$soname
`test -n "$verstring" && func_echo_all "${wl}-set_version
${wl}$verstring"` ${wl}-update_registry
${wl}${output_objdir}/so_locations -o $lib'
    hardcode_libdir_flag_spec='${wl}-rpath ${wl}$libdir'
  else
    allow_undefined_flag=' -expect_unresolved \*'
    archive_cmds='$CC -shared${allow_undefined_flag} $libobjs
$deplibs $compiler_flags -msym -soname $soname `test -n "$verstring"
&& func_echo_all "-set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib'
    archive_expsym_cmds='for i in `cat $export_symbols`; do printf
"%s %s\n" -exported_symbol "\$i" >> $lib.exp; done; printf "%s\n" "-
hidden">> $lib.exp~
$CC -shared${allow_undefined_flag} ${wl}-input ${wl}$lib.exp
$compiler_flags $libobjs $deplibs -soname $soname `test -n
"$verstring" && $ECHO "-set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib~$RM $lib.exp'

# Both c and cxx compiler support -rpath directly
hardcode_libdir_flag_spec='-rpath $libdir'
fi
archive_cmds_need_lc='no'
hardcode_libdir_separator=:
;;

solaris*)
  no_undefined_flag=' -z defs'
  if test "$GCC" = yes; then
    wlarc='${wl}'
    archive_cmds='$CC -shared $pic_flag ${wl}-z ${wl}text ${wl}-h
${wl}$soname -o $lib $libobjs $deplibs $compiler_flags'
    archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)/\1;/\" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
$CC -shared $pic_flag ${wl}-z ${wl}text ${wl}-M ${wl}$lib.exp
${wl}-h ${wl}$soname -o $lib $libobjs $deplibs $compiler_flags~$RM
$lib.exp'
  else
    case ` $CC -V 2>&1` in
      *"Compilers 5.0"*)
        wlarc=''
        archive_cmds='$LD -G${allow_undefined_flag} -h $soname -o $lib
$libobjs $deplibs $linker_flags'

```

```

        archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)\/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
        $LD -G${allow_undefined_flag} -M $lib.exp -h $soname -o $lib
$libobjs $deplibs $linker_flags~$RM $lib.exp'
        ;;
*)
        wlarc='${wl}'
        archive_cmds='$CC -G${allow_undefined_flag} -h $soname -o $lib
$libobjs $deplibs $compiler_flags'
        archive_expsym_cmds='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)\/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
        $CC -G${allow_undefined_flag} -M $lib.exp -h $soname -o $lib
$libobjs $deplibs $compiler_flags~$RM $lib.exp'
        ;;
esac
fi
hardcode_libdir_flag_spec='-R$libdir'
hardcode_shlibpath_var=no
case $host_os in
solaris2.[0-5] | solaris2.[0-5].*) ;;
*)
# The compiler driver will combine and reorder linker options,
# but understands '-z linker_flag'. GCC discards it without
`$wl',
# but is careful enough not to reorder.
# Supported since Solaris 2.6 (maybe 2.5.1?)
if test "$GCC" = yes; then
    whole_archive_flag_spec='${wl}-z ${wl}allextract$convenience
${wl}-z ${wl}defaultextract'
else
    whole_archive_flag_spec='-z allextract$convenience -z
defaultextract'
fi
;;
esac
link_all_deplibs=yes
;;

sunos4*)
    if test "x$host_vendor" = xsequent; then
        # Use $CC to link under sequent, because it throws in some extra
.o
        # files that make .init and .fini sections work.
        archive_cmds='$CC -G ${wl}-h $soname -o $lib $libobjs $deplibs
$compiler_flags'
        else
            archive_cmds='$LD -assert pure-text -Bstatic -o $lib $libobjs
$deplibs $linker_flags'
        fi
        hardcode_libdir_flag_spec='-L$libdir'

```

```

hardcode_direct=yes
hardcode_minus_L=yes
hardcode_shlibpath_var=no
;;

sysv4)
  case $host_vendor in
    sni)
      archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
      hardcode_direct=yes # is this really true???
      ;;
    siemens)
      ## LD is ld it makes a PLAMLIB
      ## CC just makes a GrossModule.
      archive_cmds='$LD -G -o $lib $libobjs $deplibs $linker_flags'
      reload_cmds='$CC -r -o $output$reload_objs'
      hardcode_direct=no
      ;;
    motorola)
      archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
      hardcode_direct=no #Motorola manual says yes, but my tests say
they lie
      ;;
    esac
    runpath_var='LD_RUN_PATH'
    hardcode_shlibpath_var=no
    ;;

sysv4.3*)
  archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
  hardcode_shlibpath_var=no
  export_dynamic_flag_spec='-Bexport'
  ;;

sysv4*MP*)
  if test -d /usr/nec; then
    archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'
    hardcode_shlibpath_var=no
    runpath_var=LD_RUN_PATH
    hardcode_runpath_var=yes
    ld_shlibs=yes
  fi
  ;;

sysv4*uw2* | sysv5OpenUNIX* | sysv5UnixWare7.[01].[10]* |
unixware7* | sco3.2v5.0.[024]*)
  no_undefined_flag='${wl}-z,text'
  archive_cmds_need_lc=no

```

```

hardcode_shlibpath_var=no
runpath_var='LD_RUN_PATH'

if test "$GCC" = yes; then
  archive_cmds='$CC -shared ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
  archive_expsym_cmds='$CC -shared ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
else
  archive_cmds='$CC -G ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
  archive_expsym_cmds='$CC -G ${wl}-Bexport:$export_symbols ${wl}-
h,$soname -o $lib $libobjs $deplibs $compiler_flags'
fi
;;

sysv5* | sco3.2v5* | sco5v6*)
# Note: We can NOT use -z defs as we might desire, because we do
not
# link with -lc, and that would cause any symbols used from libc
to
# always be unresolved, which means just about no library would
# ever link correctly.  If we're not using GNU ld we use -z text
# though, which does catch some bad symbols but isn't as heavy-
handed
# as -z defs.
no_undefined_flag='${wl}-z,text'
allow_undefined_flag='${wl}-z,nodefs'
archive_cmds_need_lc=no
hardcode_shlibpath_var=no
hardcode_libdir_flag_spec='${wl}-R,$libdir'
hardcode_libdir_separator=':'
link_all_deplibs=yes
export_dynamic_flag_spec='${wl}-Bexport'
runpath_var='LD_RUN_PATH'

if test "$GCC" = yes; then
  archive_cmds='$CC -shared ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
  archive_expsym_cmds='$CC -shared ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
else
  archive_cmds='$CC -G ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
  archive_expsym_cmds='$CC -G ${wl}-Bexport:$export_symbols ${wl}-
h,$soname -o $lib $libobjs $deplibs $compiler_flags'
fi
;;

uts4*)
  archive_cmds='$LD -G -h $soname -o $lib $libobjs $deplibs
$linker_flags'

```

```

hardcode_libdir_flag_spec='-L$libdir'
hardcode_shlibpath_var=no
;;

*)
ld_shlibs=no
;;
esac

if test x$host_vendor = xsni; then
case $host in
sysv4 | sysv4.2uw2* | sysv4.3* | sysv5*)
export_dynamic_flag_spec='${wl}-Blargedynsym'
;;
esac
fi
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ld_shlibs" >&5
$as_echo "$ld_shlibs" >&6; }
test "$ld_shlibs" = no && can_build_shared=no

with_gnu_ld=$with_gnu_ld

```

```

#
# Do we need to explicitly link libc?
#
case "x$archive_cmds_need_lc" in
x|xyes)
# Assume -lc should be added
archive_cmds_need_lc=yes

if test "$enable_shared" = yes && test "$GCC" = yes; then
case $archive_cmds in
*~*)
# FIXME: we may have to deal with multi-command sequences.
;;
*$CC *)

```



```

        # Test whether the compiler implicitly links with -lc since on
some
        # systems, -lgcc has to come before -lc. If gcc already passes -
lc
        # to ld, don't add -lc before -lgcc.
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether -lc
should be explicitly linked in" >&5
$as_echo_n "checking whether -lc should be explicitly linked in... "
>&6; }
if ${lt_cv_archive_cmds_need_lc+:} false; then :
  $as_echo_n "(cached) " >&6
else
  $RM conftest*
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext

  if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; } 2>conftest.err; then
  soname=conftest
  lib=conftest
  libobjs=conftest.$ac_objext
  deplibs=
  wl=$lt_prog_compiler_wl
  pic_flag=$lt_prog_compiler_pic
  compiler_flags=-v
  linker_flags=-v
  verstring=
  output_objdir=.
  libname=conftest
  lt_save_allow_undefined_flag=$allow_undefined_flag
  allow_undefined_flag=
  if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$archive_cmds 2\>\&1 \ | $GREP \" -lc \" \>/dev/null 2\>\&1\""; } >&5
(eval $archive_cmds 2\>\&1 \ | $GREP \" -lc \" \>/dev/null 2\>\&1)
2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }
  then
    lt_cv_archive_cmds_need_lc=no
  else
    lt_cv_archive_cmds_need_lc=yes
  fi
  allow_undefined_flag=$lt_save_allow_undefined_flag
else
  cat conftest.err 1>&5
fi
$RM conftest*

```

```
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$lt_cv_archive_cmds_need_lc" >&5
$sas_echo "$lt_cv_archive_cmds_need_lc" >&6; }
    archive_cmds_need_lc=$lt_cv_archive_cmds_need_lc
    ;;
    esac
fi
;;
esac
```



```

    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking dynamic linker
characteristics" >&5
$sas_echo_n "checking dynamic linker characteristics... " >&6; }

if test "$GCC" = yes; then
  case $host_os in
    darwin*) lt_awk_arg="/^libraries:/,/LR/" ;;
    *) lt_awk_arg="/^libraries:/" ;;
  esac
  case $host_os in
    mingw* | cegcc*) lt_sed_strip_eq="s,=\([A-Za-z]:\) ,\1,g" ;;
    *) lt_sed_strip_eq="s,=/,/,g" ;;
  esac
  lt_search_path_spec=`$CC -print-search-dirs | awk $lt_awk_arg | $SED
-e "s/^libraries:/" -e $lt_sed_strip_eq`
  case $lt_search_path_spec in
    *\;*)
      # if the path contains ";" then we assume it to be the separator
      # otherwise default to the standard path separator (i.e. ":") - it
      is
      # assumed that no part of a normal pathname contains ";" but that
      should
      # okay in the real world where ";" in dirpaths is itself
      problematic.
      lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED 's;/;/
/g'`
      ;;
    *)
      lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED
"s/$PATH_SEPARATOR/ /g`
      ;;
  esac
  # Ok, now we have the path, separated by spaces, we can step through
  it
  # and add multilib dir if necessary.
  lt_tmp_lt_search_path_spec=
  lt_multi_os_dir=`$CC $CPPFLAGS $CFLAGS $LDFLAGS -print-multi-os-
directory 2>/dev/null`
  for lt_sys_path in $lt_search_path_spec; do
    if test -d "$lt_sys_path/$lt_multi_os_dir"; then
      lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec
$lt_sys_path/$lt_multi_os_dir"
    else
      test -d "$lt_sys_path" && \
        lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec
$lt_sys_path"
    fi
  done

```

```

        fi
    done
    lt_search_path_spec=`$ECHO "$lt_tmp_lt_search_path_spec" | awk '
BEGIN {RS=" "; FS="/|\n";} {
    lt_foo="";
    lt_count=0;
    for (lt_i = NF; lt_i > 0; lt_i--) {
        if ($lt_i != "" && $lt_i != ".") {
            if ($lt_i == "..") {
                lt_count++;
            } else {
                if (lt_count == 0) {
                    lt_foo="/" $lt_i lt_foo;
                } else {
                    lt_count--;
                }
            }
        }
    }
    if (lt_foo != "") { lt_freq[lt_foo]++; }
    if (lt_freq[lt_foo] == 1) { print lt_foo; }
}'`
    # AWK program above erroneously prepends '/' to C:/dos/paths
    # for these hosts.
    case $host_os in
        mingw* | cegcc*) lt_search_path_spec=`$ECHO "$lt_search_path_spec"
|\
        $SED 's,/\/([A-Za-z]:\),\1,g'` ;;
    esac
    sys_lib_search_path_spec=`$ECHO "$lt_search_path_spec" | $lt_NL2SP`
else
    sys_lib_search_path_spec="/lib /usr/lib /usr/local/lib"
fi
library_names_spec=
libname_spec='lib$name'
soname_spec=
shrext_cmds=".so"
postinstall_cmds=
postuninstall_cmds=
finish_cmds=
finish_eval=
shlibpath_var=
shlibpath_overrides_runpath=unknown
version_type=none
dynamic_linker="$host_os ld.so"
sys_lib_dlsearch_path_spec="/lib /usr/lib"
need_lib_prefix=unknown
hardcode_into_libs=no

# when you set need_version to no, make sure it does not cause -
set_version
# flags to be left without arguments

```

```

need_version=unknown

case $host_os in
aix3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname.a'
    shlibpath_var=LIBPATH

    # AIX 3 has no versioning support, so we append a major version to
the name.
    soname_spec='${libname}${release}${shared_ext}$major'
    ;;

aix[4-9]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    hardcode_into_libs=yes
    if test "$host_cpu" = ia64; then
        # AIX 5 supports IA64
        library_names_spec='${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext}$versuffix $libname${shared_ext}'
        shlibpath_var=LD_LIBRARY_PATH
    else
        # With GCC up to 2.95.x, collect2 would create an import file
# for dependence libraries. The import file would start with
# the line `#! .'. This would cause the generated library to
# depend on `.', always an invalid library. This was fixed in
# development snapshots of GCC prior to 3.0.
        case $host_os in
aix4 | aix4.[01] | aix4.[01].*)
            if { echo '#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__
>= 97)'
                echo ' yes '
                echo '#endif'; } | ${CC} -E - | $GREP yes > /dev/null; then
                :
            else
                can_build_shared=no
            fi
            ;;
*)
                ;;
esac
        # AIX (on Power*) has no versioning support, so currently we can
not hardcode correct
        # soname into executable. Probably we can add versioning support
to
        # collect2, so additional links can be useful in future.
        if test "$aix_use_runtimelinking" = yes; then
            # If using run time linking (on AIX 4.2 or later) use
lib<name>.so

```

```

    # instead of lib<name>.a to let people know that these are not
    # typical AIX shared libraries.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    else
    # We preserve .a as extension for shared libraries through
AIX4.2
    # and later when we are not doing run time linking.
    library_names_spec='${libname}${release}.a $libname.a'
    soname_spec='${libname}${release}${shared_ext}$major'
    fi
    shlibpath_var=LIBPATH
fi
;;

amigaos*)
case $host_cpu in
powerpc)
    # Since July 2007 AmigaOS4 officially supports .so libraries.
    # When compiling the executable, add -use-dynld -lsobjs: to the
compileline.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    ;;
m68k)
    library_names_spec='$libname.ixlibrary $libname.a'
    # Create ${libname}_ixlibrary.a entries in /sys/libs.
    finish_eval='for lib in `ls $libdir/*.ixlibrary 2>/dev/null`; do
libname=`func_echo_all "$lib" | $SED
\'\'s%^\./\([^/]*\)\.ixlibrary$%\1%\''\'`; test $RM
/sys/libs/${libname}_ixlibrary.a; $show "cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a"; cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a || exit 1; done'
    ;;
esac
;;

beos*)
    library_names_spec='${libname}${shared_ext}'
    dynamic_linker="$host_os ld.so"
    shlibpath_var=LIBRARY_PATH
    ;;

bsd[45]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH

```



```

    sys_lib_search_path_spec="/shlib /usr/lib /usr/X11/lib
/usr/contrib/lib /lib /usr/local/lib"
    sys_lib_dlsearch_path_spec="/shlib /usr/lib /usr/local/lib"
    # the default ld.so.conf also contains /usr/contrib/lib and
    # /usr/X11R6/lib (/usr/X11 is a link to /usr/X11R6), but let us
allow
    # libtool to hard-code these into programs
    ;;

cygwin* | mingw* | pw32* | cegcc*)
    version_type=windows
    shrext_cmds=".dll"
    need_version=no
    need_lib_prefix=no

    case $GCC,$cc_basename in
    yes,*)
        # gcc
        library_names_spec='$libname.dll.a'
        # DLL is installed to $(libdir)/../bin by postinstall_cmds
        postinstall_cmds='base_file=`basename \${file}`~
        dlpath=`$SHELL 2>&1 -c '\''. $dir/\'''\${base_file}'\''i; echo
\${dlname}'\''`~
        dldir=$destdir/`dirname \${dlpath}`~
        test -d \${dldir} || mkdir -p \${dldir}~
        $install_prog $dir/\${dlname} \${dldir}/\${dlname}~
        chmod a+x \${dldir}/\${dlname}~
        if test -n '\''\${striplib}'\'' && test -n '\''\${striplib}'\''; then
            eval '\''\${striplib} \${dldir}/\${dlname}'\'' || exit \${?};
        fi
        postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; echo
\${dlname}'\''`~
        dlpath=$dir/\${dldll}~
        $RM \${dlpath}'
        shlibpath_overrides_runpath=yes

    case $host_os in
    cygwin*)
        # Cygwin DLLs use 'cyg' prefix rather than 'lib'
        soname_spec=`echo \${libname} | sed -e 's/^lib/cyg/'``echo
\${release} | $SED -e 's/[.]\/-g'\${versuffix}\${shared_ext}'

        sys_lib_search_path_spec="$sys_lib_search_path_spec
/usr/lib/w32api"
        ;;
    mingw* | cegcc*)
        # MinGW DLLs use traditional 'lib' prefix
        soname_spec='\${libname}`echo \${release} | $SED -e 's/[.]\/-
/g'\${versuffix}\${shared_ext}'
        ;;
    pw32*)
        # pw32 DLLs use 'pw' prefix rather than 'lib'

```

```

        library_names_spec=`echo ${libname} | sed -e 's/^lib/pw/'`echo
${release} | $SED -e 's/[.]/-/g`${versuffix}${shared_ext}'
        ;;
    esac
    dynamic_linker='Win32 ld.exe'
    ;;

*,cl*)
    # Native MSVC
    libname_spec='$name'
    soname_spec='${libname}`echo ${release} | $SED -e 's/[.]/-/g`${versuffix}${shared_ext}'
library_names_spec='${libname}.dll.lib'

    case $build_os in
    mingw*)
        sys_lib_search_path_spec=
        lt_save_ifs=$IFS
        IFS=';'
        for lt_path in $LIB
        do
            IFS=$lt_save_ifs
            # Let DOS variable expansion print the short 8.3 style file
name.
            lt_path=`cd "$lt_path" 2>/dev/null && cmd //C "for %i in (".")
do @echo %~si"`
            sys_lib_search_path_spec="$sys_lib_search_path_spec $lt_path"
        done
        IFS=$lt_save_ifs
        # Convert to MSYS style.
        sys_lib_search_path_spec=`ECHO "$sys_lib_search_path_spec" |
sed -e 's|\\\\\\|/|g' -e 's| \\([a-zA-Z]\\):| /\\1|g' -e 's|^|/'`
        ;;
    cygwin*)
        # Convert to unix form, then to dos form, then back to unix form
# but this time dos style (no spaces!) so that the unix form
looks
# like /cygdrive/c/PROGRA~1:/cygdr...
        sys_lib_search_path_spec=`cygpath --path --unix "$LIB"`
        sys_lib_search_path_spec=`cygpath --path --dos
"$sys_lib_search_path_spec" 2>/dev/null`
        sys_lib_search_path_spec=`cygpath --path --unix
"$sys_lib_search_path_spec" | $SED -e "s/$PATH_SEPARATOR/ /g"`
        ;;
    *)
        sys_lib_search_path_spec="$LIB"
        if $ECHO "$sys_lib_search_path_spec" | $GREP '[c-zA-Z]:/'
>/dev/null; then
            # It is most probably a Windows format PATH.
            sys_lib_search_path_spec=`ECHO "$sys_lib_search_path_spec" |
$SED -e 's;/;/ /g'`
        else

```

```

        sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
$SED -e "s/$PATH_SEPARATOR/ /g"`
        fi
        # FIXME: find the short name or the path components, as spaces
are
        # common. (e.g. "Program Files" -> "PROGRA~1")
        ;;
    esac

    # DLL is installed to $(libdir)/../bin by postinstall_cmds
    postinstall_cmds='base_file=`basename \${file}`~
    dlpath=`$SHELL 2>&1 -c '\''. $dir/\'''\${base_file}'\''i; echo
\${dlname}'\''`~
    dldir=$destdir/`dirname \${dlpath}`~
    test -d \${dldir} || mkdir -p \${dldir}~
    $install_prog $dir/\${dlname} \${dldir}/\${dlname}'
    postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; echo
\${dlname}'\''`~
    dlpath=$dir/\${dldll}~
    $RM \${dlpath}'
    shlibpath_overrides_runpath=yes
    dynamic_linker='Win32 link.exe'
    ;;

*)
    # Assume MSVC wrapper
    library_names_spec='${libname}`echo ${release} | $SED -e 's/[.]'/-
/g`'\${versuffix}\${shared_ext} $libname.lib'
    dynamic_linker='Win32 ld.exe'
    ;;
esac
# FIXME: first we should search . and the directory the executable
is in
shlibpath_var=PATH
;;

darwin* | rhapsody*)
dynamic_linker="$host_os dyld"
version_type=darwin
need_lib_prefix=no
need_version=no
library_names_spec='${libname}${release}${major}${shared_ext}
\${libname}${shared_ext}'
soname_spec='${libname}${release}${major}${shared_ext}'
shlibpath_overrides_runpath=yes
shlibpath_var=DYLD_LIBRARY_PATH
shrext_cmds='`test .$module = .yes && echo .so || echo .dylib`'

sys_lib_search_path_spec="$sys_lib_search_path_spec /usr/local/lib"
sys_lib_dlsearch_path_spec='/usr/local/lib /lib /usr/lib'
;;

```

```

dgux*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname$shared_ext'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    ;;

freebsd* | dragonfly*)
    # DragonFly does not have a.out.  When/if they implement a new
    # versioning mechanism, adjust this.
    if test -x /usr/bin/objformat; then
        objformat=`/usr/bin/objformat`
    else
        case $host_os in
            freebsd[23].*) objformat=aout ;;
            *) objformat=elf ;;
        esac
    fi
    version_type=freebsd-$objformat
    case $version_type in
        freebsd-elf*)
            library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext} $libname${shared_ext}'
            need_version=no
            need_lib_prefix=no
            ;;
        freebsd-*)
            library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname${shared_ext}$versuffix'
            need_version=yes
            ;;
    esac
    shlibpath_var=LD_LIBRARY_PATH
    case $host_os in
        freebsd2.*)
            shlibpath_overrides_runpath=yes
            ;;
        freebsd3.[01]* | freebsdelf3.[01]*)
            shlibpath_overrides_runpath=yes
            hardcode_into_libs=yes
            ;;
        freebsd3.[2-9]* | freebsdelf3.[2-9]* | \
        freebsd4.[0-5] | freebsdelf4.[0-5] | freebsd4.1.1 | freebsdelf4.1.1)
            shlibpath_overrides_runpath=no
            hardcode_into_libs=yes
            ;;
        *) # from 4.6 on, and DragonFly
            shlibpath_overrides_runpath=yes

```

```

        hardcode_into_libs=yes
        ;;
    esac
    ;;

gnu*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;

haiku*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    dynamic_linker="$host_os runtime_loader"
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    sys_lib_dlsearch_path_spec='/boot/home/config/lib /boot/common/lib
/boot/system/lib'
    hardcode_into_libs=yes
    ;;

hpux9* | hpux10* | hpux11*)
    # Give a soname corresponding to the major version so that dld.sl
refuses to
    # link against other versions.
    version_type=sunos
    need_lib_prefix=no
    need_version=no
    case $host_cpu in
    ia64*)
        shrext_cmds='.so'
        hardcode_into_libs=yes
        dynamic_linker="$host_os dld.so"
        shlibpath_var=LD_LIBRARY_PATH
        shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
        library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
        soname_spec='${libname}${release}${shared_ext}$major'
        if test "X$HPUX_IA64_MODE" = X32; then

```

```

        sys_lib_search_path_spec="/usr/lib/hpux32 /usr/local/lib/hpux32
/usr/local/lib"
    else
        sys_lib_search_path_spec="/usr/lib/hpux64 /usr/local/lib/hpux64"
    fi
    sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
hppa*64*)
    shrext_cmds='.sl'
    hardcode_into_libs=yes
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=LD_LIBRARY_PATH # How should we handle SHLIB_PATH
    shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    sys_lib_search_path_spec="/usr/lib/pa20_64 /usr/ccs/lib/pa20_64"
    sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
*)
    shrext_cmds='.sl'
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=SHLIB_PATH
    shlibpath_overrides_runpath=no # +s is required to enable
SHLIB_PATH
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    ;;
esac
# HP-UX runs *really* slowly unless shared libraries are mode 555,
...
postinstall_cmds='chmod 555 $lib'
# or fails outright, so override atomically:
install_override_mode=555
;;

interix[3-9]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    dynamic_linker='Interix 3.x ld.so.1 (PE, like ELF)'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;

irix5* | irix6* | nonstopux*)

```

```

case $host_os in
  nonstopux*) version_type=nonstopux ;;
*)
  if test "$lt_cv_prog_gnu_ld" = yes; then
    version_type=linux # correct to gnu/linux during the next
big refactor
  else
    version_type=irix
  fi ;;
esac
need_lib_prefix=no
need_version=no
soname_spec='${libname}${release}${shared_ext}$major'
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext} $libname${shared_ext}'
case $host_os in
  irix5* | nonstopux*)
    libsuff= shlibsuff=
  ;;
*)
  case $LD in # libtool.m4 will add one of these switches to LD
*-32|*" -32 " | *-melf32bsmip|*" -melf32bsmip ")
    libsuff= shlibsuff= libmagic=32-bit;;
*-n32|*" -n32 " | *-melf32bmipn32|*" -melf32bmipn32 ")
    libsuff=32 shlibsuff=N32 libmagic=N32;;
*-64|*" -64 " | *-melf64bmip|*" -melf64bmip ")
    libsuff=64 shlibsuff=64 libmagic=64-bit;;
*) libsuff= shlibsuff= libmagic=never-match;;
esac
  ;;
esac
shlibpath_var=LD_LIBRARY${shlibsuff}_PATH
shlibpath_overrides_runpath=no
sys_lib_search_path_spec="/usr/lib${libsuff} /lib${libsuff}
/usr/local/lib${libsuff}"
sys_lib_dlsearch_path_spec="/usr/lib${libsuff} /lib${libsuff}"
hardcode_into_libs=yes
;;

# No shared lib support for Linux oldld, aout, or coff.
linux*oldld* | linux*aout* | linux*coff*)
  dynamic_linker=no
  ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu)
  version_type=linux # correct to gnu/linux during the next big
refactor
  need_lib_prefix=no
  need_version=no

```

```

library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
soname_spec='${libname}${release}${shared_ext}$major'
finish_cmds='PATH="$PATH:/sbin" ldconfig -n $libdir'
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=no

# Some binutils ld are patched to set DT_RUNPATH
if ${lt_cv_shlibpath_overrides_runpath+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_shlibpath_overrides_runpath=no
  save_LDFLAGS=$LDFLAGS
  save_libdir=$libdir
  eval "libdir=/foo; wl=\"\$lt_prog_compiler_wl\"; \
    LDFLAGS=\"\$LDFLAGS $hardcode_libdir_flag_spec\""
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  if ($OBJDUMP -p conftest$ac_exeext) 2>/dev/null | grep
"RUNPATH.*$libdir" >/dev/null; then :
    lt_cv_shlibpath_overrides_runpath=yes
  fi
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LDFLAGS=$save_LDFLAGS
libdir=$save_libdir

fi

shlibpath_overrides_runpath=$lt_cv_shlibpath_overrides_runpath

# This implies no fast_install, which is unacceptable.
# Some rework will be needed to allow for fast_install
# before this can be enabled.
hardcode_into_libs=yes

# Append ld.so.conf contents to the search path
if test -f /etc/ld.so.conf; then
  lt_ld_extra=`awk '/^include / { system(sprintf("cd /etc; cat %s
2>/dev/null", \$2)); skip = 1; } { if (!skip) print \$0; skip = 0; }'`

```



```

< /etc/ld.so.conf | $SED -e 's/#.*//;^[ ]*hwcap[ ]/d;s/[:, ]/
/g;s/=[^=]*$/;/s/=[^= ]* / /g;s/"//g;/^$/d' | tr '\n' ' '
    sys_lib_dlsearch_path_spec="/lib /usr/lib $lt_ld_extra"
fi

# We used to test for /lib/ld.so.1 and disable shared libraries on
# powerpc, because MkLinux only supported shared libraries with the
# GNU dynamic linker. Since this was broken with cross compilers,
# most powerpc-linux boxes support dynamic linking these days and
# people can always --disable-shared, the test was removed, and we
# assume the GNU/Linux dynamic linker is in use.
dynamic_linker='GNU/Linux ld.so'
;;

netbsd*)
    version_type=sunos
    need_lib_prefix=no
    need_version=no
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
        library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
        finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
        dynamic_linker='NetBSD (a.out) ld.so'
    else
        library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
        soname_spec='${libname}${release}${shared_ext}$major'
        dynamic_linker='NetBSD ld.elf_so'
    fi
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    ;;

newsos6)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    ;;

*nto* | *qnx*)
    version_type=qnx
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no

```

```

hardcode_into_libs=yes
dynamic_linker='ldqnx.so'
;;

openbsd*)
version_type=sunos
sys_lib_dlsearch_path_spec="/usr/lib"
need_lib_prefix=no
# Some older versions of OpenBSD (3.3 at least) *do* need versioned
libs.
case $host_os in
  openbsd3.3 | openbsd3.3.*) need_version=yes ;;
  *) need_version=no ;;
esac
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
shlibpath_var=LD_LIBRARY_PATH
if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
  case $host_os in
    openbsd2.[89] | openbsd2.[89].*)
      shlibpath_overrides_runpath=no
      ;;
    *)
      shlibpath_overrides_runpath=yes
      ;;
  esac
else
  shlibpath_overrides_runpath=yes
fi
;;

os2*)
libname_spec='$name'
shrext_cmds=".dll"
need_lib_prefix=no
library_names_spec='$libname${shared_ext} $libname.a'
dynamic_linker='OS/2 ld.exe'
shlibpath_var=LIBPATH
;;

osf3* | osf4* | osf5*)
version_type=osf
need_lib_prefix=no
need_version=no
soname_spec='${libname}${release}${shared_ext}$major'
library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
shlibpath_var=LD_LIBRARY_PATH
sys_lib_search_path_spec="/usr/shlib /usr/ccs/lib /usr/lib/cmplrs/cc
/usr/lib /usr/local/lib /var/shlib"

```

```

    sys_lib_dlsearch_path_spec="$sys_lib_search_path_spec"
    ;;

rdos*)
    dynamic_linker=no
    ;;

solaris*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    # ldd complains unless libraries are executable
    postinstall_cmds='chmod +x $lib'
    ;;

sunos4*)
    version_type=sunos
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/usr/etc" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    if test "$with_gnu_ld" = yes; then
        need_lib_prefix=no
    fi
    need_version=yes
    ;;

sysv4 | sysv4.3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    case $host_vendor in
        sni)
            shlibpath_overrides_runpath=no
            need_lib_prefix=no
            runpath_var=LD_RUN_PATH
            ;;
        siemens)
            need_lib_prefix=no
            ;;
        motorola)

```

```

        need_lib_prefix=no
        need_version=no
        shlibpath_overrides_runpath=no
        sys_lib_search_path_spec='/lib /usr/lib /usr/ccs/lib'
        ;;
    esac
    ;;

sysv4*MP*)
    if test -d /usr/nec ;then
        version_type=linux # correct to gnu/linux during the next big
refactor
        library_names_spec='${libname}${shared_ext}.${versuffix}
${libname}${shared_ext}.${major} ${libname}${shared_ext}'
        soname_spec='${libname}${shared_ext}.${major}'
        shlibpath_var=LD_LIBRARY_PATH
    fi
    ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
    version_type=freebsd-elf
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}${versuffix}
${libname}${release}${shared_ext} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}${major}'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    if test "$with_gnu_ld" = yes; then
        sys_lib_search_path_spec='/usr/local/lib /usr/gnu/lib /usr/ccs/lib
/usr/lib /lib'
    else
        sys_lib_search_path_spec='/usr/ccs/lib /usr/lib'
        case $host_os in
            sco3.2v5*)
                sys_lib_search_path_spec="$sys_lib_search_path_spec /lib"
                ;;
        esac
    fi
    sys_lib_dlsearch_path_spec='/usr/lib'
    ;;

tpf*)
    # TPF is a cross-target only. Preferred cross-host = GNU/Linux.
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}${versuffix}
${libname}${release}${shared_ext}.${major} ${libname}${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH

```

```

shlibpath_overrides_runpath=no
hardcode_into_libs=yes
;;

uts4*)
  version_type=linux # correct to gnu/linux during the next big
  refactor
  library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
  soname_spec='${libname}${release}${shared_ext}$major'
  shlibpath_var=LD_LIBRARY_PATH
  ;;

*)
  dynamic_linker=no
  ;;
esac
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dynamic_linker" >&5
$as_echo "$dynamic_linker" >&6; }
test "$dynamic_linker" = no && can_build_shared=no

variables_saved_for_relink="PATH $shlibpath_var $runpath_var"
if test "$GCC" = yes; then
  variables_saved_for_relink="$variables_saved_for_relink
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH"
fi

if test "${lt_cv_sys_lib_search_path_spec+set}" = set; then
  sys_lib_search_path_spec="$lt_cv_sys_lib_search_path_spec"
fi
if test "${lt_cv_sys_lib_dlsearch_path_spec+set}" = set; then
  sys_lib_dlsearch_path_spec="$lt_cv_sys_lib_dlsearch_path_spec"
fi

```



```

    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking how to hardcode
library paths into programs" >&5
$sas_echo_n "checking how to hardcode library paths into programs... "
>&6; }
hardcode_action=
if test -n "$hardcode_libdir_flag_spec" ||
    test -n "$runpath_var" ||
    test "X$hardcode_automatic" = "Xyes" ; then

    # We can hardcode non-existent directories.
    if test "$hardcode_direct" != no &&
        # If the only mechanism to avoid hardcoding is shlibpath_var, we
        # have to relink, otherwise we might link with an installed
library
        # when we should be linking with a yet-to-be-installed one
        ## test "$_LT_TAGVAR(hardcode_shlibpath_var, )" != no &&
        test "$hardcode_minus_L" != no; then
        # Linking always hardcodes the temporary library directory.
        hardcode_action=relink
    else
        # We can link without hardcoding, and we can hardcode nonexisting
dirs.
        hardcode_action=immediate
    fi
else
    # We cannot hardcode anything, or else we can only hardcode existing
# directories.
    hardcode_action=unsupported
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $hardcode_action" >&5
$sas_echo "$hardcode_action" >&6; }

```

```

if test "$hardcode_action" = relink ||
  test "$inherit_rpath" = yes; then
  # Fast installation is not supported
  enable_fast_install=no
elif test "$shlibpath_overrides_runpath" = yes ||
  test "$enable_shared" = no; then
  # Fast installation is not necessary
  enable_fast_install=needless
fi

```

```

if test "x$enable_dlopen" != xyes; then
  enable_dlopen=unknown
  enable_dlopen_self=unknown
  enable_dlopen_self_static=unknown
else
  lt_cv_dlopen=no
  lt_cv_dlopen_libs=

  case $host_os in
    beos*)
      lt_cv_dlopen="load_add_on"
      lt_cv_dlopen_libs=
      lt_cv_dlopen_self=yes
      ;;

    mingw* | pw32* | cegcc*)
      lt_cv_dlopen="LoadLibrary"
      lt_cv_dlopen_libs=
      ;;

    cygwin*)
      lt_cv_dlopen="dlopen"
      lt_cv_dlopen_libs=
      ;;

    darwin*)
      # if libdl is installed we need to link against it
      { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -ldl" >&5
$as_echo_n "checking for dlopen in -ldl... " >&6; }
if ${ac_cv_lib_dl_dlopen+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-ldl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext

```



```

/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int
main ()
{
return dlopen ();
    ;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_dl_dlopen=yes
else
    ac_cv_lib_dl_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_lib_dl_dlopen"
>&5
$as_echo "$ac_cv_lib_dl_dlopen" >&6; }
if test "x$ac_cv_lib_dl_dlopen" = xyes; then :
    lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-ldl"
else

    lt_cv_dlopen="dyld"
    lt_cv_dlopen_libs=
    lt_cv_dlopen_self=yes

fi

;;

*)
    ac_fn_c_check_func "$LINENO" "shl_load" "ac_cv_func_shl_load"
if test "x$ac_cv_func_shl_load" = xyes; then :
    lt_cv_dlopen="shl_load"
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for shl_load in -
ldld" >&5
$as_echo_n "checking for shl_load in -ldld... " >&6; }
if ${ac_cv_lib_dld_shl_load+:} false; then :
    $as_echo_n "(cached) " >&6
else

```

```

    ac_check_lib_save_LIBS=$LIBS
LIBS="-ldld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char shl_load ();
int
main ()
{
return shl_load ();
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_dld_shl_load=yes
else
    ac_cv_lib_dld_shl_load=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dld_shl_load" >&5
$as_echo "$ac_cv_lib_dld_shl_load" >&6; }
if test "x$ac_cv_lib_dld_shl_load" = xyes; then :
    lt_cv_dlopen="shl_load" lt_cv_dlopen_libs="-ldld"
else
    ac_fn_c_check_func "$LINENO" "dlopen" "ac_cv_func_dlopen"
if test "x$ac_cv_func_dlopen" = xyes; then :
    lt_cv_dlopen="dlopen"
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -
ldl" >&5
$as_echo_n "checking for dlopen in -ldl... " >&6; }
if ${ac_cv_lib_dl_dlopen+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-ldl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC

```

```

    builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int
main ()
{
return dlopen ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_dl_dlopen=yes
else
ac_cv_lib_dl_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_lib_dl_dlopen"
>&5
$as_echo "$ac_cv_lib_dl_dlopen" >&6; }
if test "x$ac_cv_lib_dl_dlopen" = xyes; then :
lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-ldl"
else
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for dlopen in -
lsvld" >&5
$as_echo_n "checking for dlopen in -lsvld... " >&6; }
if ${ac_cv_lib_svld_dlopen+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-lsvld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dlopen ();
int
main ()
{
return dlopen ();
;
return 0;

```

```

}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_svld_dlopen=yes
else
  ac_cv_lib_svld_dlopen=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_svld_dlopen" >&5
$as_echo "$ac_cv_lib_svld_dlopen" >&6; }
if test "x$ac_cv_lib_svld_dlopen" = xyes; then :
  lt_cv_dlopen="dlopen" lt_cv_dlopen_libs="-lsvld"
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for dld_link in -
ldld" >&5
$as_echo_n "checking for dld_link in -ldld... " >&6; }
if ${ac_cv_lib_dld_dld_link+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-ldld $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char dld_link ();
int
main ()
{
return dld_link ();
  ;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_dld_dld_link=yes
else
  ac_cv_lib_dld_dld_link=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dld_dld_link" >&5
$as_echo "$ac_cv_lib_dld_dld_link" >&6; }
if test "x$ac_cv_lib_dld_dld_link" = xyes; then :
  lt_cv_dlopen="dld_link" lt_cv_dlopen_libs="-ldld"
fi

fi

fi

fi

fi

fi

;;
esac

if test "x$lt_cv_dlopen" != xno; then
  enable_dlopen=yes
else
  enable_dlopen=no
fi

case $lt_cv_dlopen in
dlopen)
  save_CPPFLAGS="$CPPFLAGS"
  test "x$ac_cv_header_dlfcn_h" = xyes && CPPFLAGS="$CPPFLAGS -
DHAVE_DLFCN_H"

  save_LDFLAGS="$LDFLAGS"
  wl=$lt_prog_compiler_wl eval LDFLAGS="\`$LDFLAGS
$export_dynamic_flag_spec`"

  save_LIBS="$LIBS"
  LIBS="$lt_cv_dlopen_libs $LIBS"

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether a
program can dlopen itself" >&5
$as_echo_n "checking whether a program can dlopen itself... " >&6; }
if ${lt_cv_dlopen_self+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :
    lt_cv_dlopen_self=cross

```

```

else
    lt_dlunknown=0; lt_dlno_uscore=1; lt_dlneed_uscore=2
    lt_status=$lt_dlunknown
    cat > conftest.$ac_ext <<_LT_EOF
#line $LINENO "configure"
#include "confdefs.h"

#if HAVE_DLFCN_H
#include <dlfcn.h>
#endif

#include <stdio.h>

#ifdef RTLD_GLOBAL
# define LT_DLGLOBAL      RTLD_GLOBAL
#else
# ifdef DL_GLOBAL
#   define LT_DLGLOBAL    DL_GLOBAL
# else
#   define LT_DLGLOBAL    0
# endif
#endif

/* We may have to define LT_DLLAZY_OR_NOW in the command line if we
   find out it does not work in some platform. */
#ifndef LT_DLLAZY_OR_NOW
# ifdef RTLD_LAZY
#   define LT_DLLAZY_OR_NOW      RTLD_LAZY
# else
#   ifdef DL_LAZY
#     define LT_DLLAZY_OR_NOW    DL_LAZY
#   else
#     ifdef RTLD_NOW
#       define LT_DLLAZY_OR_NOW  RTLD_NOW
#     else
#       ifdef DL_NOW
#         define LT_DLLAZY_OR_NOW    DL_NOW
#       else
#         define LT_DLLAZY_OR_NOW    0
#       endif
#     endif
#   endif
# endif
#endif

/* When -fvisibility=hidden is used, assume the code has been annotated
   correspondingly for the symbols needed. */
#if defined(__GNUC__) && (((__GNUC__ == 3) && (__GNUC_MINOR__ >= 3))
|| (__GNUC__ > 3))
int fnord () __attribute__((visibility("default")));
#endif

```

```

int fnord () { return 42; }
int main ()
{
  void *self = dlopen (0, LT_DLGLOBAL|LT_DLLAZY_OR_NOW);
  int status = $lt_dlunknown;

  if (self)
    {
      if (dlsym (self,"fnord"))      status = $lt_dlno_uscore;
      else
        {
          if (dlsym( self,"_fnord")) status = $lt_dlneed_uscore;
          else puts (dlerror ());
        }
      /* dlclose (self); */
    }
  else
    puts (dlerror ());

  return status;
}
_LT_EOF
if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}: \"$ac_link\""; }
>&5
(eval $ac_link) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; } && test -s conftest${ac_exeext} 2>/dev/null;
then
  (./conftest; exit; ) >&5 2>/dev/null
  lt_status=$?
  case x$lt_status in
    x$lt_dlno_uscore) lt_cv_dlopen_self=yes ;;
    x$lt_dlneed_uscore) lt_cv_dlopen_self=yes ;;
    x$lt_dlunknown|x*) lt_cv_dlopen_self=no ;;
  esac
  else :
    # compilation failed
    lt_cv_dlopen_self=no
  fi
fi
rm -fr conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_dlopen_self"
>&5
$as_echo "$lt_cv_dlopen_self" >&6; }

  if test "x$lt_cv_dlopen_self" = xyes; then
    wl=$lt_prog_compiler_wl eval LDFLAGS="\`$LDFLAGS
$lt_prog_compiler_static\`"

```

```

        { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether a
statically linked program can dlopen itself" >&5
$as_echo_n "checking whether a statically linked program can dlopen
itself... " >&6; }
if ${lt_cv_dlopen_self_static+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :
    lt_cv_dlopen_self_static=cross
  else
    lt_dlunknown=0; lt_dlno_uscore=1; lt_dlneed_uscore=2
    lt_status=$lt_dlunknown
    cat > conftest.$ac_ext <<_LT_EOF
#line $LINENO "configure"
#include "confdefs.h"

#if HAVE_DLFCN_H
#include <dlfcn.h>
#endif

#include <stdio.h>

#ifdef RTLD_GLOBAL
#  define LT_DLGLOBAL      RTLD_GLOBAL
#else
#  ifdef DL_GLOBAL
#    define LT_DLGLOBAL      DL_GLOBAL
#  else
#    define LT_DLGLOBAL      0
#  endif
#endif

/* We may have to define LT_DLLAZY_OR_NOW in the command line if we
   find out it does not work in some platform. */
#ifndef LT_DLLAZY_OR_NOW
#  ifdef RTLD_LAZY
#    define LT_DLLAZY_OR_NOW      RTLD_LAZY
#  else
#    ifdef DL_LAZY
#      define LT_DLLAZY_OR_NOW      DL_LAZY
#    else
#      ifdef RTLD_NOW
#        define LT_DLLAZY_OR_NOW RTLD_NOW
#      else
#        ifdef DL_NOW
#          define LT_DLLAZY_OR_NOW      DL_NOW
#        else
#          define LT_DLLAZY_OR_NOW      0
#        endif
#      endif
#    endif
#  endif
#endif
#endif

```



```

#endif

/* When -fvisibility=hidden is used, assume the code has been annotated
   correspondingly for the symbols needed.  */
#if defined(__GNUC__) && (((__GNUC__ == 3) && (__GNUC_MINOR__ >= 3))
|| (__GNUC__ > 3))
int fnord () __attribute__((visibility("default")));
#endif

int fnord () { return 42; }
int main ()
{
    void *self = dlopen (0, LT_DLGLOBAL|LT_DLLAZY_OR_NOW);
    int status = $lt_dlunknown;

    if (self)
    {
        if (dlsym (self,"fnord")          status = $lt_dlno_uscore;
        else
        {
            if (dlsym( self,"_fnord")    status = $lt_dlneed_uscore;
            else puts (dlerror ());
        }
        /* dlclose (self); */
    }
    else
        puts (dlerror ());

    return status;
}
_LT_EOF
if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}: \"$ac_link\""; }
>&5
(eval $ac_link) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \$? = $ac_status" >&5
test $ac_status = 0; } && test -s conftest${ac_exeext} 2>/dev/null;
then
    (./conftest; exit; ) >&5 2>/dev/null
    lt_status=$?
    case x$lt_status in
        x$lt_dlno_uscore) lt_cv_dlopen_self_static=yes ;;
        x$lt_dlneed_uscore) lt_cv_dlopen_self_static=yes ;;
        x$lt_dlunknown|x*) lt_cv_dlopen_self_static=no ;;
    esac
else :
    # compilation failed
    lt_cv_dlopen_self_static=no
fi
fi
rm -fr conftest*

```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_dlopen_self_static" >&5
$as_echo "$lt_cv_dlopen_self_static" >&6; }
  fi

  CPPFLAGS="$save_CPPFLAGS"
  LDFLAGS="$save_LDFLAGS"
  LIBS="$save_LIBS"
  ;;
esac

case $lt_cv_dlopen_self in
yes|no) enable_dlopen_self=$lt_cv_dlopen_self ;;
*) enable_dlopen_self=unknown ;;
esac

case $lt_cv_dlopen_self_static in
yes|no) enable_dlopen_self_static=$lt_cv_dlopen_self_static ;;
*) enable_dlopen_self_static=unknown ;;
esac
fi

striplib=
old_striplib=
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether stripping
libraries is possible" >&5
$as_echo_n "checking whether stripping libraries is possible... " >&6;
}
if test -n "$STRIP" && $STRIP -V 2>&1 | $GREP "GNU strip" >/dev/null;
then
  test -z "$old_striplib" && old_striplib="$STRIP --strip-debug"
  test -z "$striplib" && striplib="$STRIP --strip-unneeded"
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }

```

```

else
# FIXME - insert some real tests, host_os isn't really good enough
case $host_os in
darwin*)
    if test -n "$STRIP" ; then
        striplib="$STRIP -x"
        old_striplib="$STRIP -S"
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    fi
    ;;
*)
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    ;;
esac
fi

# Report which library types will actually be built
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if libtool
supports shared libraries" >&5
$as_echo_n "checking if libtool supports shared libraries... " >&6; }
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $can_build_shared"
>&5
$as_echo "$can_build_shared" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
shared libraries" >&5
$as_echo_n "checking whether to build shared libraries... " >&6; }
test "$can_build_shared" = "no" && enable_shared=no

# On AIX, shared libraries and static libraries use the same
namespace, and
# are all built from PIC.
case $host_os in
aix3*)
    test "$enable_shared" = yes && enable_static=no
    if test -n "$RANLIB"; then

```

```

        archive_cmds="$archive_cmds~\${RANLIB} \${lib}"
        postinstall_cmds='${RANLIB} \${lib}'
    fi
    ;;

aix[4-9]*)
    if test "$host_cpu" != ia64 && test "$aix_use_runtimelinking" = no
; then
        test "$enable_shared" = yes && enable_static=no
        fi
        ;;
    esac
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_shared" >&5
$as_echo "$enable_shared" >&6; }

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
static libraries" >&5
$as_echo_n "checking whether to build static libraries... " >&6; }
    # Make sure either enable_shared or enable_static is yes.
    test "$enable_shared" = yes || enable_static=yes
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_static" >&5
$as_echo "$enable_static" >&6; }

fi
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

CC="$lt_save_CC"

    if test -n "$CXX" && ( test "X$CXX" != "Xno" &&
( (test "X$CXX" = "Xg++" && `g++ -v >/dev/null 2>&1` ) ||
(test "X$CXX" != "Xg++")) ) ; then
        ac_ext=cpp
        ac_cpp='$CXXCPP $CPPFLAGS'
        ac_compile='$CXX -c $CXXFLAGS $CPPFLAGS conftest.$ac_ext >&5'
        ac_link='$CXX -o conftest$ac_exeext $CXXFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
        ac_compiler_gnu=$ac_cv_cxx_compiler_gnu
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking how to run the C++
preprocessor" >&5
$as_echo_n "checking how to run the C++ preprocessor... " >&6; }
        if test -z "$CXXCPP"; then
            if ${ac_cv_prog_CXXCPP+:} false; then :
                $as_echo_n "(cached) " >&6
            else

```

```

        # Double quotes because CXXCPP needs to be expanded
        for CXXCPP in "$CXX -E" "/lib/cpp"
        do
            ac_preproc_ok=false
        for ac_cxx_preproc_warn_flag in ' yes'
        do
            # Use a header file that comes with gcc, so configuring glibc
            # with a fresh cross-compiler works.
            # Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
            # <limits.h> exists even on freestanding compilers.
            # On the NeXT, cc -E runs the code through the compiler's parser,
            # not just through cpp. "Syntax error" is here to catch this case.
            cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#ifdef __STDC__
#include <limits.h>
#else
#include <assert.h>
#endif

                Syntax error

            _ACEOF
        if ac_fn_cxx_try_cpp "$LINENO"; then :

        else
            # Broken: fails on valid input.
            continue
        fi
        rm -f conftest.err conftest.i conftest.$ac_ext

            # OK, works on sane cases. Now check whether nonexistent headers
            # can be detected and how.
            cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <ac_nonexistent.h>
            _ACEOF
        if ac_fn_cxx_try_cpp "$LINENO"; then :
            # Broken: success on invalid input.
            continue
        else
            # Passes both tests.
            ac_preproc_ok=:
            break
        fi
        rm -f conftest.err conftest.i conftest.$ac_ext

        done
        # Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
        rm -f conftest.i conftest.err conftest.$ac_ext
        if $ac_preproc_ok; then :
            break
        fi

```

```

done
ac_cv_prog_CXXCPP=$CXXCPP

fi
CXXCPP=$ac_cv_prog_CXXCPP
else
ac_cv_prog_CXXCPP=$CXXCPP
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $CXXCPP" >&5
$as_echo "$CXXCPP" >&6; }
ac_preproc_ok=false
for ac_cxx_preproc_warn_flag in '' yes
do
# Use a header file that comes with gcc, so configuring glibc
# with a fresh cross-compiler works.
# Prefer <limits.h> to <assert.h> if __STDC__ is defined, since
# <limits.h> exists even on freestanding compilers.
# On the NeXT, cc -E runs the code through the compiler's parser,
# not just through cpp. "Syntax error" is here to catch this case.
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#ifdef __STDC__
# include <limits.h>
#else
# include <assert.h>
#endif

                Syntax error

_ACEOF
if ac_fn_cxx_try_cpp "$LINENO"; then :

else
# Broken: fails on valid input.
continue
fi
rm -f conftest.err conftest.i conftest.$ac_ext

# OK, works on sane cases. Now check whether nonexistent headers
# can be detected and how.
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <ac_nonexistent.h>
_ACEOF
if ac_fn_cxx_try_cpp "$LINENO"; then :
# Broken: success on invalid input.
continue
else
# Passes both tests.
ac_preproc_ok=:
break
fi
rm -f conftest.err conftest.i conftest.$ac_ext

```

```

done
# Because of `break', _AC_PREPROC_IFELSE's cleaning code was skipped.
rm -f confptest.i confptest.err confptest.$ac_ext
if $ac_preproc_ok; then :

else
  { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `'$ac_pwd':"
>&5
$as_echo "$as_me: error: in `'$ac_pwd':" >&2;}
as_fn_error $? "C++ preprocessor `"$CXXCPP`" fails sanity check
See `config.log' for more details" "$LINENO" 5; }
fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS confptest.$ac_ext >&5'
ac_link='$CC -o confptest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
confptest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

else
  _lt_caught_CXX_error=yes
fi

ac_ext=cpp
ac_cpp='$CXXCPP $CPPFLAGS'
ac_compile='$CXX -c $CXXFLAGS $CPPFLAGS confptest.$ac_ext >&5'
ac_link='$CXX -o confptest$ac_exeext $CXXFLAGS $CPPFLAGS $LDFLAGS
confptest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_cxx_compiler_gnu

archive_cmds_need_lc_CXX=no
allow_undefined_flag_CXX=
always_export_symbols_CXX=no
archive_expsym_cmds_CXX=
compiler_needs_object_CXX=no
export_dynamic_flag_spec_CXX=
hardcode_direct_CXX=no
hardcode_direct_absolute_CXX=no
hardcode_libdir_flag_spec_CXX=
hardcode_libdir_separator_CXX=
hardcode_minus_L_CXX=no
hardcode_shlibpath_var_CXX=unsupported
hardcode_automatic_CXX=no
inherit_rpath_CXX=no
module_cmds_CXX=
module_expsym_cmds_CXX=
link_all_deplibs_CXX=unknown
old_archive_cmds_CXX=$old_archive_cmds
reload_flag_CXX=$reload_flag
reload_cmds_CXX=$reload_cmds
no_undefined_flag_CXX=

```

```

whole_archive_flag_spec_CXX=
enable_shared_with_static_runtimes_CXX=no

# Source file extension for C++ test sources.
ac_ext=cpp

# Object file extension for compiled C++ test sources.
objext=o
objext_CXX=$objext

# No sense in running all these tests if we already determined that
# the CXX compiler isn't working. Some variables (like enable_shared)
# are currently assumed to apply to all compilers on this platform,
# and will be corrupted by setting them based on a non-working
# compiler.
if test "$_lt_caught_CXX_error" != yes; then
  # Code to be used in simple compile tests
  lt_simple_compile_test_code="int some_variable = 0;"

  # Code to be used in simple link tests
  lt_simple_link_test_code='int main(int, char *[]) { return(0); }'

  # ltmain only uses $CC for tagged configurations so make sure $CC is
  # set.

# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC

  # save warnings/boilerplate of simple test code
  ac_outfile=conftest.$ac_objext
  echo "$lt_simple_compile_test_code" >conftest.$ac_ext
  eval "$ac_compile" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d'
  >conftest.err
  _lt_compiler_boilerplate=`cat conftest.err`
  $RM conftest*

  ac_outfile=conftest.$ac_objext
  echo "$lt_simple_link_test_code" >conftest.$ac_ext
  eval "$ac_link" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d' >conftest.err
  _lt_linker_boilerplate=`cat conftest.err`

```



```
$RM -r conftest*
```

```
# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_CFLAGS=$CFLAGS
lt_save_LD=$LD
lt_save_GCC=$GCC
GCC=$GXX
lt_save_with_gnu_ld=$with_gnu_ld
lt_save_path_LD=$lt_cv_path_LD
if test -n "${lt_cv_prog_gnu_ldcxx+set}"; then
  lt_cv_prog_gnu_ld=$lt_cv_prog_gnu_ldcxx
else
  $as_unset lt_cv_prog_gnu_ld
fi
if test -n "${lt_cv_path_LDCXX+set}"; then
  lt_cv_path_LD=$lt_cv_path_LDCXX
else
  $as_unset lt_cv_path_LD
fi
test -z "${LDCXX+set}" || LD=$LDCXX
CC=${CXX-"c++"}
CFLAGS=$CXXFLAGS
compiler=$CC
compiler_CXX=$CC
for cc_temp in $compiler""; do
case $cc_temp in
  compile | *[\//]compile | ccache | *[\//]ccache ) ;;
  distcc | *[\//]distcc | purify | *[\//]purify ) ;;
  \-*) ;;
  *) break;;
esac
done
cc_basename=`$ECHO "$cc_temp" | $SED "s%.*/%%; s%^\$host_alias-%%"`

if test -n "$compiler"; then
  # We don't want -fno-exception when compiling C++ code, so set the
  # no_builtin_flag separately
  if test "$GXX" = yes; then
    lt_prog_compiler_no_builtin_flag_CXX=' -fno-builtin'
  else
    lt_prog_compiler_no_builtin_flag_CXX=
  fi

  if test "$GXX" = yes; then
    # Set up default GNU C++ configuration

# Check whether --with-gnu-ld was given.
```

```

if test "${with_gnu_ld+set}" = set; then :
  withval=$with_gnu_ld; test "$withval" = no || with_gnu_ld=yes
else
  with_gnu_ld=no
fi

ac_prog=ld
if test "$GCC" = yes; then
  # Check if gcc -print-prog-name=ld gives a path.
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for ld used by
$CC" >&5
$as_echo_n "checking for ld used by $CC... " >&6; }
  case $host in
  *-*-mingw*)
    # gcc leaves a trailing carriage return which upsets mingw
    ac_prog=`($CC -print-prog-name=ld) 2>&5 | tr -d '\015'` ;;
  *)
    ac_prog=`($CC -print-prog-name=ld) 2>&5` ;;
  esac
  case $ac_prog in
  # Accept absolute paths.
  [\\/] * | ?:[\\/] *)
    re_direlt='/[^\/] [^\/]*/\.\./'
    # Canonicalize the pathname of ld
    ac_prog=`$ECHO "$ac_prog"| $SED 's%\\%/%g'`
    while $ECHO "$ac_prog" | $GREP "$re_direlt" > /dev/null 2>&1; do
      ac_prog=`$ECHO $ac_prog| $SED "s%$re_direlt%/"`
    done
    test -z "$LD" && LD="$ac_prog"
    ;;
  "")
    # If it fails, then pretend we aren't using GCC.
    ac_prog=ld
    ;;
  *)
    # If it is relative, then search for the first ld in PATH.
    with_gnu_ld=unknown
    ;;
  esac
elif test "$with_gnu_ld" = yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for GNU ld" >&5
$as_echo_n "checking for GNU ld... " >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for non-GNU ld"
>&5
$as_echo_n "checking for non-GNU ld... " >&6; }
fi
if ${lt_cv_path_LD+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test -z "$LD"; then
    lt_save_ifs="$IFS"; IFS=$PATH_SEPARATOR

```

```

for ac_dir in $PATH; do
  IFS="$lt_save_ifs"
  test -z "$ac_dir" && ac_dir=.
  if test -f "$ac_dir/$ac_prog" || test -f
"$ac_dir/$ac_prog$ac_exeext"; then
    lt_cv_path_LD="$ac_dir/$ac_prog"
    # Check to see if the program is GNU ld.  I'd rather use --
version,
    # but apparently some variants of GNU ld only accept -v.
    # Break only if it was the GNU/non-GNU ld that we prefer.
    case `"$lt_cv_path_LD" -v 2>&1 </dev/null` in
      *GNU* | *'with BFD'*)
        test "$with_gnu_ld" != no && break
        ;;
      *)
        test "$with_gnu_ld" != yes && break
        ;;
    esac
  fi
done
IFS="$lt_save_ifs"
else
  lt_cv_path_LD="$LD" # Let the user override the test with a path.
fi
fi

LD="$lt_cv_path_LD"
if test -n "$LD"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LD" >&5
$as_echo "$LD" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi
test -z "$LD" && as_fn_error $? "no acceptable ld found in \$PATH"
"$LINENO" 5
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if the linker ($LD)
is GNU ld" >&5
$as_echo_n "checking if the linker ($LD) is GNU ld... " >&6; }
if ${lt_cv_prog_gnu_ld+:} false; then :
  $as_echo_n "(cached) " >&6
else
  # I'd rather use --version here, but apparently some GNU lds only
accept -v.
  case `"$LD" -v 2>&1 </dev/null` in
    *GNU* | *'with BFD'*)
      lt_cv_prog_gnu_ld=yes
      ;;
    *)
      lt_cv_prog_gnu_ld=no
      ;;
  esac

```

```

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $lt_cv_prog_gnu_ld"
>&5
$as_echo "$lt_cv_prog_gnu_ld" >&6; }
with_gnu_ld=$lt_cv_prog_gnu_ld

# Check if GNU C++ uses GNU ld as the underlying linker, since
the
# archiving commands below assume that GNU ld is being used.
if test "$with_gnu_ld" = yes; then
    archive_cmds_CXX='$CC $pic_flag -shared -nostdlib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname $wl$soname -o $lib'
    archive_expsym_cmds_CXX='$CC $pic_flag -shared -nostdlib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname $wl$soname ${wl}-retain-symbols-file $wl$export_symbols -
o $lib'

    hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
    export_dynamic_flag_spec_CXX='${wl}--export-dynamic'

# If archive_cmds runs LD, not CC, wlarc should be empty
# XXX I think wlarc can be eliminated in ltcf-cxx, but I need
to
#     investigate it a little bit more. (MM)
wlarc='${wl}'

# ancient GNU ld didn't support --whole-archive et. al.
if eval "`$CC -print-prog-name=ld` --help 2>&1" |
$GREP 'no-whole-archive' > /dev/null; then
    whole_archive_flag_spec_CXX="$wlarc"--whole-
archive$convenience "'$wlarc"--no-whole-archive'
    else
        whole_archive_flag_spec_CXX=
    fi
else
    with_gnu_ld=no
    wlarc=

# A generic and very simple default shared library creation
# command for GNU C++ for the case where it uses the native
# linker, instead of GNU ld.  If possible, this setting should
# overridden to take advantage of the native linker features
on
# the platform it is being used on.

```

```

        archive_cmds_CXX='$CC -shared -nostdlib $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags -o $lib'
    fi

    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used when
    # linking a shared library.
    output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext
2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'

else
    GXX=no
    with_gnu_ld=no
    wlarc=
fi

# PORTME: fill in a description of your system's C++ link
characteristics
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the
$compiler linker ($LD) supports shared libraries" >&5
$as_echo_n "checking whether the $compiler linker ($LD) supports
shared libraries... " >&6; }
ld_shlibs_CXX=yes
case $host_os in
    aix3*)
        # FIXME: insert proper C++ library support
        ld_shlibs_CXX=no
        ;;
    aix[4-9]*)
        if test "$host_cpu" = ia64; then
            # On IA64, the linker does run time linking by default, so
we don't
            # have to do anything special.
            aix_use_runtimelinking=no
            exp_sym_flag='-Bexport'
            no_entry_flag=""
        else
            aix_use_runtimelinking=no

            # Test if we are trying to use run time linking or normal
            # AIX style linking. If -brtl is somewhere in LDFLAGS, we
            # need to do runtime linking.
            case $host_os in aix4.[23]|aix4.[23].*|aix[5-9]*)
                for ld_flag in $LDFLAGS; do
                    case $ld_flag in
                        *-brtl*)
                            aix_use_runtimelinking=yes
                            break
                            ;;
                    esac
                done
                ;;
            ;;
        esac
done
;;

```

```

    esac

    exp_sym_flag='-bexport'
    no_entry_flag='-bnoentry'
fi

# When large executables or shared objects are built, AIX ld
can
# have problems creating the table of contents.  If linking a
library
# or program results in "error TOC overflow" add -mminimal-toc
to
# CXXFLAGS/CFLAGS for g++/gcc.  In the cases where that is not
# enough to fix the problem, add -Wl,-bbigtoc to LDFLAGS.

archive_cmds_CXX=''
hardcode_direct_CXX=yes
hardcode_direct_absolute_CXX=yes
hardcode_libdir_separator_CXX=':'
link_all_deplibs_CXX=yes
file_list_spec_CXX='${wl}-f,'

if test "$GXX" = yes; then
    case $host_os in aix4.[012]|aix4.[012].*)
        # We only want to do this on AIX 4.2 and lower, the check
        # below for broken collect2 doesn't work under 4.3+
        collect2name=`${CC} -print-prog-name=collect2`
        if test -f "$collect2name" &&
            strings "$collect2name" | $GREP resolve_lib_name >/dev/null
        then
            # We have reworked collect2
            :
        else
            # We have old collect2
            hardcode_direct_CXX=unsupported
            # It fails to find uninstalled libraries when the uninstalled
            # path is not listed in the libpath.  Setting
hardcode_minus_L
            # to unsupported forces relinking
            hardcode_minus_L_CXX=yes
            hardcode_libdir_flag_spec_CXX='-L$libdir'
            hardcode_libdir_separator_CXX=
        fi
    esac
    shared_flag='-shared'
    if test "$aix_use_runtimelinking" = yes; then
        shared_flag="$shared_flag "'${wl}-G'
    fi
else
    # not using gcc
    if test "$host_cpu" = ia64; then

```

```

        # VisualAge C++, Version 5.5 for AIX 5L for IA-64, Beta 3
Release
        # chokes on -Wl,-G. The following line is correct:
        shared_flag='-G'
        else
        if test "$aix_use_runtimelinking" = yes; then
            shared_flag='${wl}-G'
        else
            shared_flag='${wl}-bM:SRE'
        fi
        fi
        fi

        export_dynamic_flag_spec_CXX='${wl}-bexpall'
        # It seems that -bexpall does not export symbols beginning
with
        # underscore (_), so it is better to generate a list of
symbols to
        # export.
        always_export_symbols_CXX=yes
        if test "$aix_use_runtimelinking" = yes; then
            # Warning - without using the other runtime loading flags (-
brtl),
            # -berok will link without error, but may produce a broken
library.
            allow_undefined_flag_CXX='-berok'
            # Determine the default libpath from the value encoded in an
empty
            # executable.
            if test "${lt_cv_aix_libpath+set}" = set; then
                aix_libpath=$lt_cv_aix_libpath
            else
                if ${lt_cv_aix_libpath_CXX+:} false; then :
                    $as_echo_n "(cached) " >&6
                else
                    cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_cxx_try_link "$LINENO"; then :

    lt_aix_libpath_sed='
        /Import File Strings/,/^$/ {
            /^0/ {
                s/^0 *\[^\]*\ *$/\1/

```

```

        }
    }'
    lt_cv_aix_libpath_CXX=`dump -H conftest$sac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
    # Check for a 64-bit object if we didn't find anything.
    if test -z "$lt_cv_aix_libpath_CXX"; then
        lt_cv_aix_libpath_CXX=`dump -HX64 conftest$sac_exeext 2>/dev/null
| $SED -n -e "$lt_aix_libpath_sed"`
    fi
fi
rm -f core conftest.err conftest.$sac_objext \
conftest$sac_exeext conftest.$sac_ext
if test -z "$lt_cv_aix_libpath_CXX"; then
    lt_cv_aix_libpath_CXX="/usr/lib:/lib"
fi

fi

aix_libpath=$lt_cv_aix_libpath_CXX
fi

        hardcode_libdir_flag_spec_CXX='${wl}-
bllibpath:$libdir:'"$aix_libpath"

        archive_expsym_cmds_CXX='$CC -o $output_objdir/$soname
$libobjs $deplibs '"'\${wl}$no_entry_flag"' $compiler_flags `if test
"x${allow_undefined_flag}" != "x"; then func_echo_all
"${wl}${allow_undefined_flag}"; else ;; fi`
'"'\${wl}$exp_sym_flag:\$export_symbols $shared_flag"
    else
        if test "$host_cpu" = ia64; then
            hardcode_libdir_flag_spec_CXX='${wl}-R $libdir:/usr/lib:/lib'
            allow_undefined_flag_CXX="-z nodefs"
            archive_expsym_cmds_CXX="\$CC $shared_flag" -o
$output_objdir/$soname $libobjs $deplibs '"'\${wl}$no_entry_flag"'
$compiler_flags ${wl}${allow_undefined_flag}
'"'\${wl}$exp_sym_flag:\$export_symbols"
        else
            # Determine the default libpath from the value encoded in an
            # empty executable.
            if test "${lt_cv_aix_libpath+set}" = set; then
                aix_libpath=$lt_cv_aix_libpath
            else
                if ${lt_cv_aix_libpath_CXX+:} false; then :
                    $as_echo_n "(cached) " >&6
                else
                    cat confdefs.h - <<_ACEOF >>conftest.$sac_ext
/* end confdefs.h.  */

int
main ()

```



```

{
    ;
    return 0;
}
_ACEOF
if ac_fn_cxx_try_link "$LINENO"; then :

    lt_aix_libpath_sed='
        /Import File Strings/,/^$/ {
            /^0/ {
                s/^0 *\[^\ ]*\) *$/\1/
                p
            }
        }'

    lt_cv_aix_libpath_CXX=`dump -H conftest$sac_exeext 2>/dev/null |
$SED -n -e "$lt_aix_libpath_sed"`
    # Check for a 64-bit object if we didn't find anything.
    if test -z "$lt_cv_aix_libpath_CXX"; then
        lt_cv_aix_libpath_CXX=`dump -HX64 conftest$sac_exeext 2>/dev/null
| $SED -n -e "$lt_aix_libpath_sed"`
    fi
fi
rm -f core conftest.err conftest.$sac_objext \
    conftest$sac_exeext conftest.$sac_ext
if test -z "$lt_cv_aix_libpath_CXX"; then
    lt_cv_aix_libpath_CXX="/usr/lib:/lib"
fi

fi

aix_libpath=$lt_cv_aix_libpath_CXX
fi

    hardcode_libdir_flag_spec_CXX='${wl}-
bllibpath:$libdir:'"$aix_libpath"
    # Warning - without using the other run time loading flags,
    # -berok will link without error, but may produce a broken
library.
    no_undefined_flag_CXX=' ${wl}-bernotok'
    allow_undefined_flag_CXX=' ${wl}-berok'
    if test "$with_gnu_ld" = yes; then
        # We only use this code for GNU lds that support --whole-
archive.
        whole_archive_flag_spec_CXX='${wl}--whole-
archive$convenience ${wl}--no-whole-archive'
    else
        # Exported symbols can be pulled into shared objects from
archives
        whole_archive_flag_spec_CXX='$convenience'
    fi
    archive_cmds_need_lc_CXX=yes

```

```

        # This is similar to how AIX traditionally builds its shared
        # libraries.
        archive_expsym_cmds_CXX="\$CC \$shared_flag" -o
$output_objdir/$soname $libobjs $deplibs ${wl}-bnoentry
$compiler_flags ${wl}-bE:$export_symbols${allow_undefined_flag}~$AR
$AR_FLAGS $output_objdir/$libname$release.a $output_objdir/$soname'
        fi
        fi
        ;;

beos*)
    if $LD --help 2>&1 | $GREP ': supported targets:.* elf' >
/dev/null; then
        allow_undefined_flag_CXX=unsupported
        # Joseph Beckenbach <jrb3@best.com> says some releases of gcc
        # support --undefined. This deserves some investigation.
    FIXME
        archive_cmds_CXX='$CC -nostart $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
        else
            ld_shlibs_CXX=no
        fi
        ;;

chorus*)
    case $cc_basename in
        *)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        esac
        ;;

cygwin* | mingw* | pw32* | cegcc*)
    case $GXX,$cc_basename in
        ,cl* | no,cl*)
            # Native MSVC
            # hardcode_libdir_flag_spec is actually meaningless, as there
is
            # no search path for DLLs.
            hardcode_libdir_flag_spec_CXX=' '
            allow_undefined_flag_CXX=unsupported
            always_export_symbols_CXX=yes
            file_list_spec_CXX='@'
            # Tell ltmain to make .lib files, not .a files.
            libext=lib
            # Tell ltmain to make .dll files, not .so files.
            shrext_cmds=".dll"
            # FIXME: Setting linknames here is a bad hack.
            archive_cmds_CXX='$CC -o $output_objdir/$soname $libobjs
$compiler_flags $deplibs -Wl,-dll~linknames='

```

```

        archive_expsym_cmds_CXX='if test "x`$SED 1q $export_symbols`" =
xEXPORTS; then
            $SED -n -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -
EXPORT:\\\\\\\\\\\\\\\\1/' -e '1\\\\\\\\!p' < $export_symbols >
$output_objdir/$soname.exp;
        else
            $SED -e 's/\\\\\\\\\\\\\\\\(.*\\\\\\\\\\\\\\\\)/-link\\\\\\\\ -EXPORT:\\\\\\\\\\\\\\\\1/' <
$export_symbols > $output_objdir/$soname.exp;
        fi~
        $CC -o $tool_output_objdir$soname $libobjs $compiler_flags
$deplibs "@$tool_output_objdir$soname.exp" -Wl,-DLL,-
IMPLIB:"$tool_output_objdir$libname.dll.lib"~
        linknames='
        # The linker will not automatically build a static lib if we
build a DLL.
        # _LT_TAGVAR(old_archive_from_new_cmds, CXX)='true'
enable_shared_with_static_runtimes_CXX=yes
        # Don't use ranlib
old_postinstall_cmds_CXX='chmod 644 $oldlib'
postlink_cmds_CXX='lt_outputfile="@OUTPUT@"~
        lt_tool_outputfile="@TOOL_OUTPUT@"~
        case $lt_outputfile in
        *.exe|*.EXE) ;;
        *)
            lt_outputfile="$lt_outputfile.exe"
            lt_tool_outputfile="$lt_tool_outputfile.exe"
            ;;
        esac~
        func_to_tool_file "$lt_outputfile"~
        if test "$MANIFEST_TOOL" != ":" && test -f
"$lt_outputfile.manifest"; then
            $MANIFEST_TOOL -manifest "$lt_tool_outputfile.manifest" -
outputresource:"$lt_tool_outputfile" || exit 1;
            $RM "$lt_outputfile.manifest";
        fi'
        ;;
*)
        # g++
        # _LT_TAGVAR(hardcode_libdir_flag_spec, CXX) is actually
meaningless,
        # as there is no search path for DLLs.
hardcode_libdir_flag_spec_CXX='-L$libdir'
export_dynamic_flag_spec_CXX='${wl}--export-all-symbols'
allow_undefined_flag_CXX=unsupported
always_export_symbols_CXX=no
enable_shared_with_static_runtimes_CXX=yes

        if $LD --help 2>&1 | $GREP 'auto-import' > /dev/null; then
            archive_cmds_CXX='$CC -shared -nostdlib $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags -o
$output_objdir/$soname ${wl}--enable-auto-image-base -Xlinker --out-
implib -Xlinker $lib'

```

```

# If the export-symbols file already is a .def file (1st line
# is EXPORTS), use it as is; otherwise, prepend...
archive_expsym_cmds_CXX='if test "x`$SED 1q $export_symbols`"
= xEXPORTS; then
    cp $export_symbols $output_objdir/$soname.def;
else
    echo EXPORTS > $output_objdir/$soname.def;
    cat $export_symbols >> $output_objdir/$soname.def;
fi~
$CC -shared -nostdlib $output_objdir/$soname.def
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags -o
$output_objdir/$soname ${wl}--enable-auto-image-base -Xlinker --out-
implib -Xlinker $lib'
else
    ld_shlibs_CXX=no
fi
;;
esac
;;
darwin* | rhapsody*)

archive_cmds_need_lc_CXX=no
hardcode_direct_CXX=no
hardcode_automatic_CXX=yes
hardcode_shlibpath_var_CXX=unsupported
if test "$lt_cv_ld_force_load" = "yes"; then
    whole_archive_flag_spec_CXX='`for conv in $convenience\`; do
test -n "`$conv`" && new_convenience="$new_convenience ${wl}-
force_load,$conv\`; done; func_echo_all "`$new_convenience`''
else
    whole_archive_flag_spec_CXX=''
fi
link_all_deplibs_CXX=yes
allow_undefined_flag_CXX="$lt_dar_allow_undefined"
case $cc_basename in
    ifort*) _lt_dar_can_shared=yes ;;
    *) _lt_dar_can_shared=$GCC ;;
esac
if test "$lt_dar_can_shared" = "yes"; then
    output_verbose_link_cmd=func_echo_all
    archive_cmds_CXX="\$CC -dynamiclib \$allow_undefined_flag -o \$lib
\$libobjs \$deplibs \$compiler_flags -install_name \$rpath/\$soname
\$verstring \$lt_dar_single_mod${_lt_dsymutil}"
    module_cmds_CXX="\$CC \$allow_undefined_flag -o \$lib -bundle
\$libobjs \$deplibs \$compiler_flags${_lt_dsymutil}"
    archive_expsym_cmds_CXX="sed 's,^,_, ' < \$export_symbols >
\$output_objdir/\${libname}-symbols.expsym~\$CC -dynamiclib
\$allow_undefined_flag -o \$lib \$libobjs \$deplibs \$compiler_flags -
install_name \$rpath/\$soname \$verstring
\${_lt_dar_single_mod}\${_lt_dar_export_syms}\${_lt_dsymutil}"

```

```

    module_expsym_cmds_CXX="sed -e 's,^,_,,' < \${export_symbols} >
\${output_objdir}/\${libname}-symbols.expsym~\${CC} \${allow_undefined_flag}
-o \${lib} -bundle \${libobjs} \${deplibs}
\${compiler_flags}\${_lt_dar_export_syms}\${_lt_dsymutil}"
    if test "\${lt_cv_apple_cc_single_mod}" != "yes"; then
        archive_cmds_CXX="\${CC} -r -keep_private_externs -nostdlib -o
\${lib}-master.o \${libobjs}~\${CC} -dynamiclib \${allow_undefined_flag} -o
\${lib} \${lib}-master.o \${deplibs} \${compiler_flags} -install_name
\${rpath}/\${soname} \${verstring}\${_lt_dsymutil}"
        archive_expsym_cmds_CXX="sed 's,^,_,,' < \${export_symbols} >
\${output_objdir}/\${libname}-symbols.expsym~\${CC} -r -
keep_private_externs -nostdlib -o \${lib}-master.o \${libobjs}~\${CC} -
dynamiclib \${allow_undefined_flag} -o \${lib} \${lib}-master.o \${deplibs}
\${compiler_flags} -install_name \${rpath}/\${soname}
\${verstring}\${_lt_dar_export_syms}\${_lt_dsymutil}"
    fi

else
ld_shlibs_CXX=no
fi

;;

dgux*)
    case \${cc_basename} in
        ec++*)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        ghcx*)
            # Green Hills C++ Compiler
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        *)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
    esac
;;

freebsd2.*)
    # C++ shared libraries reported to be fairly broken before
# switch to ELF
    ld_shlibs_CXX=no
    ;;

freebsd-elf*)
    archive_cmds_need_lc_CXX=no
    ;;

freebsd* | dragonfly*)

```

```

# FreeBSD 3 and later use GNU C++ and GNU ld with standard ELF
# conventions
ld_shlibs_CXX=yes
;;

gnu*)
;;

haiku*)
archive_cmds_CXX='$CC -shared $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
link_all_deplibs_CXX=yes
;;

hpux9*)
hardcode_libdir_flag_spec_CXX='${wl}+b ${wl}$libdir'
hardcode_libdir_separator_CXX=:
export_dynamic_flag_spec_CXX='${wl}-E'
hardcode_direct_CXX=yes
hardcode_minus_L_CXX=yes # Not in the search PATH,
                          # but as the default
                          # location of the library.

case $cc_basename in
CC*)
# FIXME: insert proper C++ library support
ld_shlibs_CXX=no
;;
aCC*)
archive_cmds_CXX='$RM $output_objdir/$soname~$CC -b
${wl}+b ${wl}$install_libdir -o $output_objdir/$soname $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags~test
$output_objdir/$soname = $lib || mv $output_objdir/$soname $lib'
# Commands to make compiler produce verbose output that
lists
# what "hidden" libraries, object files and flags are used
when
# linking a shared library.
#
# There doesn't appear to be a way to prevent this
compiler from
# explicitly linking system object files so we need to
strip them
# from the output so that they don't get included in the
library
# dependencies.
output_verbose_link_cmd='templist=`($CC -b $CFLAGS -v
conftest.$objext 2>&1) | $EGREP "\-L" `; list=""; for z in $templist;
do case $z in conftest.$objext) list="$list $z";; *.$objext);; *)
list="$list $z";; esac; done; func_echo_all "$list"
;;
*)

```

```

        if test "$GXX" = yes; then
            archive_cmds_CXX='$RM $output_objdir/$soname~$CC -shared
-nostdlib $pic_flag ${wl}+b ${wl}$install_libdir -o
$output_objdir/$soname $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags~test $output_objdir/$soname = $lib ||
mv $output_objdir/$soname $lib'
        else
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
        fi
    ;;
esac
;;

hpux10*|hpux11*)
    if test $with_gnu_ld = no; then
        hardcode_libdir_flag_spec_CXX='${wl}+b ${wl}$libdir'
        hardcode_libdir_separator_CXX=:

        case $host_cpu in
            hppa*64*|ia64*)
                ;;
            *)
                export_dynamic_flag_spec_CXX='${wl}-E'
                ;;
        esac
    fi
    case $host_cpu in
        hppa*64*|ia64*)
            hardcode_direct_CXX=no
            hardcode_shlibpath_var_CXX=no
            ;;
        *)
            hardcode_direct_CXX=yes
            hardcode_direct_absolute_CXX=yes
            hardcode_minus_L_CXX=yes # Not in the search PATH,
                                     # but as the default
                                     # location of the library.
            ;;
    esac

    case $cc_basename in
        CC*)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        aCC*)
            case $host_cpu in
                hppa*64*)
                    archive_cmds_CXX='$CC -b ${wl}+h ${wl}$soname -o $lib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
                    ;;
            esac
        ;;
    esac

```

```

        ia64*)
            archive_cmds_CXX='$CC -b ${wl}+h ${wl}$soname
${wl}+nodefaulttrpath -o $lib $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags'
            ;;
        *)
            archive_cmds_CXX='$CC -b ${wl}+h ${wl}$soname ${wl}+b
${wl}$install_libdir -o $lib $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags'
            ;;
    esac
    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used
when
    # linking a shared library.
    #
    # There doesn't appear to be a way to prevent this compiler
from
    # explicitly linking system object files so we need to strip
them
    # from the output so that they don't get included in the
library
    # dependencies.
    output_verbose_link_cmd='templist=`($CC -b $CFLAGS -v
conftest.$objext 2>&1) | $GREP "\-L" `; list=""; for z in $templist; do
case $z in conftest.$objext) list="$list $z";; *.$objext);; *)
list="$list $z";;esac; done; func_echo_all "$list"'
    ;;
    *)
    if test "$GXX" = yes; then
        if test $with_gnu_ld = no; then
            case $host_cpu in
                hppa*64*)
                    archive_cmds_CXX='$CC -shared -nostdlib -fPIC ${wl}+h
${wl}$soname -o $lib $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags'
                    ;;
                ia64*)
                    archive_cmds_CXX='$CC -shared -nostdlib $pic_flag
${wl}+h ${wl}$soname ${wl}+nodefaulttrpath -o $lib $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags'
                    ;;
                *)
                    archive_cmds_CXX='$CC -shared -nostdlib $pic_flag
${wl}+h ${wl}$soname ${wl}+b ${wl}$install_libdir -o $lib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
                    ;;
            esac
        fi
    else
        # FIXME: insert proper C++ library support
        ld_shlibs_CXX=no
    fi

```



```

        fi
        ;;
    esac
    ;;

    interix[3-9]*)
hardcode_direct_CXX=no
hardcode_shlibpath_var_CXX=no
hardcode_libdir_flag_spec_CXX='${wl}-rpath,$libdir'
export_dynamic_flag_spec_CXX='${wl}-E'
# Hack: On Interix 3.x, we cannot compile PIC because of a broken
gcc.
# Instead, shared libraries are loaded at an image base
(0x10000000 by
# default) and relocated if they conflict, which is a slow very
memory
# consuming and fragmenting process. To avoid this, we pick a
random,
# 256 KiB-aligned image base between 0x50000000 and 0x6FFC0000 at
link
# time. Moving up from 0x10000000 also allows more sbrk(2)
space.
archive_cmds_CXX='$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--image-base,`expr ${RANDOM-$$} %
4096 / 2 \* 262144 + 1342177280` -o $lib'
archive_expsym_cmds_CXX='sed "s,^,_" $export_symbols
>$output_objdir/$soname.expsym~$CC -shared $pic_flag $libobjs $deplibs
$compiler_flags ${wl}-h,$soname ${wl}--retain-symbols-
file,$output_objdir/$soname.expsym ${wl}--image-base,`expr ${RANDOM-
$$} % 4096 / 2 \* 262144 + 1342177280` -o $lib'
    ;;
    irix5* | irix6*)
        case $cc_basename in
            CC*)
                # SGI C++
                archive_cmds_CXX='$CC -shared -all -multigot $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags -soname $soname
`test -n "$verstring" && func_echo_all "--set_version $verstring"` -
update_registry ${output_objdir}/so_locations -o $lib'

                # Archives containing C++ object files must be created using
                # "CC -ar", where "CC" is the IRIX C++ compiler. This is
                # necessary to make sure instantiated templates are included
                # in the archive.
                old_archive_cmds_CXX='$CC -ar -WR,-u -o $oldlib $oldobjs'
            ;;
            *)
                if test "$GXX" = yes; then
                    if test "$with_gnu_ld" = no; then
                        archive_cmds_CXX='$CC -shared $pic_flag -nostdlib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname ${wl}$soname `test -n "$verstring" && func_echo_all

```

```

"${wl}-set_version ${wl}$verstring"` ${wl}-update_registry
${wl}${output_objdir}/so_locations -o $lib'
    else
        archive_cmds_CXX='$CC -shared $pic_flag -nostdlib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname ${wl}$soname `test -n "$verstring" && func_echo_all
"${wl}-set_version ${wl}$verstring"` -o $lib'
        fi
    fi
    link_all_deplibs_CXX=yes
;;
esac
hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
hardcode_libdir_separator_CXX=:
inherit_rpath_CXX=yes
;;

linux* | k*bsd*-gnu | kopensolaris*-gnu)
case $cc_basename in
    KCC*)
        # Kuck and Associates, Inc. (KAI) C++ Compiler

        # KCC will only create a shared library if the output file
        # ends with ".so" (or ".sl" for HP-UX), so rename the library
        # to its proper name (with version) after linking.
        archive_cmds_CXX='tempext=`echo $shared_ext | $SED -e
\'\'s/\([^\()0-9A-Za-z{}]\)/\\\\\\\\1/g\'\'`; templib=`echo $lib | $SED -e
"s/\${tempext}\..*/.so/"`; $CC $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags --soname $soname -o \${templib}; mv
\${templib} $lib'
        archive_expsym_cmds_CXX='tempext=`echo $shared_ext | $SED -e
\'\'s/\([^\()0-9A-Za-z{}]\)/\\\\\\\\1/g\'\'`; templib=`echo $lib | $SED -e
"s/\${tempext}\..*/.so/"`; $CC $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags --soname $soname -o \${templib} ${wl}-
retain-symbols-file,$export_symbols; mv \${templib} $lib'
        # Commands to make compiler produce verbose output that lists
        # what "hidden" libraries, object files and flags are used
when
        # linking a shared library.
        #
        # There doesn't appear to be a way to prevent this compiler
from
        # explicitly linking system object files so we need to strip
them
        # from the output so that they don't get included in the
library
        # dependencies.
        output_verbose_link_cmd='templist=`$CC $CFLAGS -v
conftest.$objext -o libconftest$shared_ext 2>&1 | $GREP "ld"`; rm -f
libconftest$shared_ext; list=""; for z in $templist; do case $z in
conftest.$objext) list="$list $z";; *.objext);; *) list="$list
$z";;esac; done; func_echo_all "$list"'

```

```

hardcode_libdir_flag_spec_CXX='${wl}-rpath,$libdir'
export_dynamic_flag_spec_CXX='${wl}--export-dynamic'

# Archives containing C++ object files must be created using
# "CC -Bstatic", where "CC" is the KAI C++ compiler.
old_archive_cmds_CXX='$CC -Bstatic -o $oldlib $oldobjs'
;;
icpc* | ecpc* )
# Intel C++
with_gnu_ld=yes
# version 8.0 and above of icpc choke on multiply defined
symbols
# if we add $predep_objects and $postdep_objects, however 7.1
and
# earlier do not add the objects themselves.
case ` $CC -V 2>&1 ` in
*"Version 7."*)
    archive_cmds_CXX='$CC -shared $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags ${wl}-soname $wl$soname -o
$lib'
    archive_expsym_cmds_CXX='$CC -shared $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags ${wl}-soname
$wl$soname ${wl}-retain-symbols-file $wl$export_symbols -o $lib'
    ;;
*) # Version 8.0 or newer
    tmp_idyn=
    case $host_cpu in
    ia64*) tmp_idyn=' -i_dynamic';;
    esac
    archive_cmds_CXX='$CC -shared"$tmp_idyn"' $libobjs
$deplibs $compiler_flags ${wl}-soname $wl$soname -o $lib'
    archive_expsym_cmds_CXX='$CC -shared"$tmp_idyn"' $libobjs
$deplibs $compiler_flags ${wl}-soname $wl$soname ${wl}-retain-symbols-
file $wl$export_symbols -o $lib'
    ;;
esac
archive_cmds_need_lc_CXX=no
hardcode_libdir_flag_spec_CXX='${wl}-rpath,$libdir'
export_dynamic_flag_spec_CXX='${wl}--export-dynamic'
whole_archive_flag_spec_CXX='${wl}--whole-archive$convenience
${wl}--no-whole-archive'
;;
pgCC* | pgcpp*)
# Portland Group C++ compiler
case ` $CC -V ` in
*pgCC\ [1-5].* | *pgcpp\ [1-5].*)
    prelink_cmds_CXX='tpldir=Template.dir~
rm -rf $tpldir~
$CC --prelink_objects --instantiation_dir $tpldir $objs
$libobjs $compile_deplibs~

```

```

        compile_command="$compile_command `find $tpldir -name \*.o
| sort | $NL2SP`"'
        old_archive_cmds_CXX='tpldir=Template.dir~
rm -rf $tpldir~
$CC --prelink_objects --instantiation_dir $tpldir
Soldobjs$old_deplibs~
$AR $AR_FLAGS $oldlib$oldobjs$old_deplibs `find $tpldir -
name \*.o | sort | $NL2SP`~
$RANLIB $oldlib'
        archive_cmds_CXX='tpldir=Template.dir~
rm -rf $tpldir~
$CC --prelink_objects --instantiation_dir $tpldir
$predep_objects $libobjs $deplibs $convenience $postdep_objects~
$CC -shared $pic_flag $predep_objects $libobjs $deplibs
`find $tpldir -name \*.o | sort | $NL2SP` $postdep_objects
$compiler_flags ${wl}-soname ${wl}$soname -o $lib'
        archive_expsym_cmds_CXX='tpldir=Template.dir~
rm -rf $tpldir~
$CC --prelink_objects --instantiation_dir $tpldir
$predep_objects $libobjs $deplibs $convenience $postdep_objects~
$CC -shared $pic_flag $predep_objects $libobjs $deplibs
`find $tpldir -name \*.o | sort | $NL2SP` $postdep_objects
$compiler_flags ${wl}-soname ${wl}$soname ${wl}-retain-symbols-file
${wl}$export_symbols -o $lib'
        ;;
        *) # Version 6 and above use weak symbols
        archive_cmds_CXX='$CC -shared $pic_flag $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags ${wl}-soname
${wl}$soname -o $lib'
        archive_expsym_cmds_CXX='$CC -shared $pic_flag
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname ${wl}$soname ${wl}-retain-symbols-file
${wl}$export_symbols -o $lib'
        ;;
    esac

    hardcode_libdir_flag_spec_CXX='${wl}--rpath ${wl}$libdir'
    export_dynamic_flag_spec_CXX='${wl}--export-dynamic'
    whole_archive_flag_spec_CXX='${wl}--whole-archive`for conv in
$convenience\`\`; do test -n \`${conv}\` &&
new_convenience="\$new_convenience,$conv\`; done; func_echo_all
\`${new_convenience}\` ` ${wl}--no-whole-archive'
    ;;
    cxx*)
        # Compaq C++
        archive_cmds_CXX='$CC -shared $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags ${wl}-soname $wl$soname -o
$lib'
        archive_expsym_cmds_CXX='$CC -shared $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags ${wl}-soname $wl$soname -o
$lib ${wl}-retain-symbols-file $wl$export_symbols'

```

```

runpath_var=LD_RUN_PATH
hardcode_libdir_flag_spec_CXX='-rpath $libdir'
hardcode_libdir_separator_CXX=:

# Commands to make compiler produce verbose output that lists
# what "hidden" libraries, object files and flags are used
when
# linking a shared library.
#
# There doesn't appear to be a way to prevent this compiler
from
# explicitly linking system object files so we need to strip
them
# from the output so that they don't get included in the
library
# dependencies.
output_verbose_link_cmd='templist=`$CC -shared $CFLAGS -v
conftest.$objext 2>&1 | $GREP "ld"; templist=`func_echo_all
"$templist" | $SED "s/\(^.*ld.*\)\( .*ld .*$\)/\1/"`; list=""; for z
in $templist; do case $z in conftest.$objext) list="$list $z";;
*.$objext);; *) list="$list $z";; esac; done; func_echo_all "X$list" |
$Xsed'

;;
xl* | mpixl* | bgxl*)
# IBM XL 8.0 on PPC, with GNU ld
hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
export_dynamic_flag_spec_CXX='${wl}--export-dynamic'
archive_cmds_CXX='$CC -qmkshrobj $libobjs $deplibs
$compiler_flags ${wl}-soname $wl$soname -o $lib'
if test "x$supports_anon_versioning" = xyes; then
archive_expsym_cmds_CXX='echo "{ global:" >
$output_objdir/$libname.ver~
cat $export_symbols | sed -e "s/\(.*\)/\1;/" >>
$output_objdir/$libname.ver~
echo "local: *; };" >> $output_objdir/$libname.ver~
$CC -qmkshrobj $libobjs $deplibs $compiler_flags ${wl}-
soname $wl$soname ${wl}-version-script
${wl}$output_objdir/$libname.ver -o $lib'
fi
;;
*)
case ` $CC -V 2>&1 | sed 5q` in
*Sun\ C*)
# Sun C++ 5.9
no_undefined_flag_CXX=' -zdefs'
archive_cmds_CXX='$CC -G${allow_undefined_flag} -h$soname -
o $lib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags'
archive_expsym_cmds_CXX='$CC -G${allow_undefined_flag} -
h$soname -o $lib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags ${wl}-retain-symbols-file ${wl}$export_symbols'
hardcode_libdir_flag_spec_CXX='-R$libdir'

```

```

        whole_archive_flag_spec_CXX='${wl}--whole-
archive`new_convenience=; for conv in $convenience\`; do test -z
\`${conv}\` || new_convenience=\`${new_convenience},${conv}\`; done;
func_echo_all \`${new_convenience}\` \`${wl}--no-whole-archive'
        compiler_needs_object_CXX=yes

        # Not sure whether something based on
        # $CC $CFLAGS -v conftest.$objext -o libconfctest$shared_ext
2>&1
        # would be better.
        output_verbose_link_cmd='func_echo_all'

        # Archives containing C++ object files must be created
using
        # "CC -xar", where "CC" is the Sun C++ compiler. This is
        # necessary to make sure instantiated templates are
included
        # in the archive.
        old_archive_cmds_CXX='$CC -xar -o $oldlib $oldobjs'
        ;;
    esac
    ;;
esac
;;

lynxos*)
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
    ;;

m88k*)
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
    ;;

mvs*)
    case $cc_basename in
        cxx*)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        *)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
    esac
    ;;

netbsd*)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
        archive_cmds_CXX='$LD -Bshareable -o $lib $predep_objects
$libobjs $deplibs $postdep_objects $linker_flags'

```

```

    wlarc=
    hardcode_libdir_flag_spec_CXX='-R$libdir'
    hardcode_direct_CXX=yes
    hardcode_shlibpath_var_CXX=no
fi
# Workaround some broken pre-1.5 toolchains
output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext
2>&1 | $GREP conftest.$objext | $SED -e "s:-lgcc -lc -lgcc::"'
;;

*nto* | *qnx*)
    ld_shlibs_CXX=yes
;;

openbsd2*)
    # C++ shared libraries are fairly broken
    ld_shlibs_CXX=no
;;

openbsd*)
if test -f /usr/libexec/ld.so; then
    hardcode_direct_CXX=yes
    hardcode_shlibpath_var_CXX=no
    hardcode_direct_absolute_CXX=yes
    archive_cmds_CXX='$CC -shared $pic_flag $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags -o $lib'
    hardcode_libdir_flag_spec_CXX='${wl}-rpath,$libdir'
    if test -z "`echo __ELF__ | $CC -E - | grep __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
        archive_expsym_cmds_CXX='$CC -shared $pic_flag
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-retain-symbols-file,$export_symbols -o $lib'
        export_dynamic_flag_spec_CXX='${wl}-E'
        whole_archive_flag_spec_CXX="$wlarc"'--whole-
archive$convenience "'$wlarc"'--no-whole-archive'
    fi
    output_verbose_link_cmd=func_echo_all
else
    ld_shlibs_CXX=no
fi
;;

osf3* | osf4* | osf5*)
    case $cc_basename in
        KCC*)
            # Kuck and Associates, Inc. (KAI) C++ Compiler

            # KCC will only create a shared library if the output file
            # ends with ".so" (or ".sl" for HP-UX), so rename the library
            # to its proper name (with version) after linking.
            archive_cmds_CXX='tempext=`echo $shared_ext | $SED -e
'\''s/\([^\()0-9A-Za-z{\}\]\)/\\\\\\\\1/g'\''`; templib=`echo "$lib" | $SED

```

```

-e "s/\${tempext}\..*/.so/"`; $CC $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags --soname $soname -o \${templib}; mv
\${templib} $lib'

hardcode_libdir_flag_spec_CXX='${wl}-rpath,$libdir'
hardcode_libdir_separator_CXX=:

# Archives containing C++ object files must be created using
# the KAI C++ compiler.
case $host in
  osf3*) old_archive_cmds_CXX='$CC -Bstatic -o $oldlib
$oldobjs' ;;
  *) old_archive_cmds_CXX='$CC -o $oldlib $oldobjs' ;;
esac
;;
RCC*)
# Rational C++ 2.4.1
# FIXME: insert proper C++ library support
ld_shlibs_CXX=no
;;
cxx*)
case $host in
  osf3*)
    allow_undefined_flag_CXX=' ${wl}-expect_unresolved
${wl}\*'
    archive_cmds_CXX='$CC -shared${allow_undefined_flag}
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-soname $soname `test -n "$verstring" && func_echo_all "${wl}-
set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib'
    hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
    ;;
  *)
    allow_undefined_flag_CXX=' -expect_unresolved \*'
    archive_cmds_CXX='$CC -shared${allow_undefined_flag}
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags -
msym -soname $soname `test -n "$verstring" && func_echo_all "-
set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib'
    archive_expsym_cmds_CXX='for i in `cat $export_symbols`;
do printf "%s %s\n" -exported_symbol "\$i" >> $lib.exp; done~
echo "-hidden">> $lib.exp~
$CC -shared$allow_undefined_flag $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags -msym -soname
$soname ${wl}-input ${wl}$lib.exp `test -n "$verstring" && $ECHO "-
set_version $verstring"` -update_registry
${output_objdir}/so_locations -o $lib~
$RM $lib.exp'
    hardcode_libdir_flag_spec_CXX=' -rpath $libdir'
    ;;
;;
esac

```



```

hardcode_libdir_separator_CXX=:

# Commands to make compiler produce verbose output that lists
# what "hidden" libraries, object files and flags are used
when
# linking a shared library.
#
# There doesn't appear to be a way to prevent this compiler
from
# explicitly linking system object files so we need to strip
them
# from the output so that they don't get included in the
library
# dependencies.
output_verbose_link_cmd='templist=`$CC -shared $CFLAGS -v
confstest.$objext 2>&1 | $GREP "ld" | $GREP -v "ld:"`;
templist=`func_echo_all "$templist" | $SED "s/\(^.*ld.*\)\(
.*ld.*$\)/\1/"`; list=""; for z in $templist; do case $z in
confstest.$objext) list="$list $z";; *.objext);; *) list="$list
$z";;esac; done; func_echo_all "$list"
;;
*)
if test "$GXX" = yes && test "$with_gnu_ld" = no; then
allow_undefined_flag_CXX=' ${wl}-expect_unresolved ${wl}\*'
case $host in
osf3*)
archive_cmds_CXX='$CC -shared -nostdlib
${allow_undefined_flag} $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags ${wl}-soname ${wl}$soname `test -n
"$verstring" && func_echo_all "${wl}-set_version ${wl}$verstring"`
${wl}-update_registry ${wl}${output_objdir}/so_locations -o $lib'
;;
*)
archive_cmds_CXX='$CC -shared $pic_flag -nostdlib
${allow_undefined_flag} $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags ${wl}-msym ${wl}-soname ${wl}$soname
`test -n "$verstring" && func_echo_all "${wl}-set_version
${wl}$verstring"` ${wl}-update_registry
${wl}${output_objdir}/so_locations -o $lib'
;;
esac

hardcode_libdir_flag_spec_CXX='${wl}-rpath ${wl}$libdir'
hardcode_libdir_separator_CXX=:

# Commands to make compiler produce verbose output that
lists
# what "hidden" libraries, object files and flags are used
when
# linking a shared library.
output_verbose_link_cmd='$CC -shared $CFLAGS -v
confstest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'

```

```

else
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
fi
;;
esac
;;

psos*)
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
    ;;

sunos4*)
    case $cc_basename in
        CC*)
            # Sun C++ 4.x
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        lcc*)
            # Lucid
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
        *)
            # FIXME: insert proper C++ library support
            ld_shlibs_CXX=no
            ;;
    esac
    ;;

solaris*)
    case $cc_basename in
        CC* | sunCC*)
            # Sun C++ 4.2, 5.x and Centerline C++
            archive_cmds_need_lc_CXX=yes
            no_undefined_flag_CXX='-zdefs'
            archive_cmds_CXX='$CC -G${allow_undefined_flag} -h$soname -o
$lib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags'
            archive_expsym_cmds_CXX='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)\/\1;/\" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
$CC -G${allow_undefined_flag} ${wl}-M ${wl}$lib.exp -
h$soname -o $lib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags~$RM $lib.exp'

            hardcode_libdir_flag_spec_CXX='-R$libdir'
            hardcode_shlibpath_var_CXX=no
        case $host_os in

```

```

solaris2.[0-5] | solaris2.[0-5].*) ;;
*)
# The compiler driver will combine and reorder linker
options,
# but understands `-z linker_flag'.
# Supported since Solaris 2.6 (maybe 2.5.1?)
whole_archive_flag_spec_CXX='-z allextract$convenience -z
defaultextract'
;;
esac
link_all_deplibs_CXX=yes

output_verbose_link_cmd='func_echo_all'

# Archives containing C++ object files must be created using
# "CC -xar", where "CC" is the Sun C++ compiler. This is
# necessary to make sure instantiated templates are included
# in the archive.
old_archive_cmds_CXX='$CC -xar -o $oldlib $oldobjs'
;;
gcx*)
# Green Hills C++ Compiler
archive_cmds_CXX='$CC -shared $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags ${wl}-h $wl$soname -o $lib'

# The C++ compiler must be used to create the archive.
old_archive_cmds_CXX='$CC $LDFLAGS -archive -o $oldlib
$oldobjs'
;;
*)
# GNU C++ compiler with Solaris linker
if test "$GXX" = yes && test "$with_gnu_ld" = no; then
no_undefined_flag_CXX=' ${wl}-z ${wl}defs'
if $CC --version | $GREP -v '^2\.7' > /dev/null; then
archive_cmds_CXX='$CC -shared $pic_flag -nostdlib
$LDFLAGS $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags ${wl}-h $wl$soname -o $lib'
archive_expsym_cmds_CXX='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)\/\1;/" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
$CC -shared $pic_flag -nostdlib ${wl}-M $wl$lib.exp -o
$lib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags~$RM $lib.exp'

# Commands to make compiler produce verbose output that
lists
# what "hidden" libraries, object files and flags are
used when
# linking a shared library.
output_verbose_link_cmd='$CC -shared $CFLAGS -v
conftest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'
else

```

```

        # g++ 2.7 appears to require '-G' NOT '-shared' on this
        # platform.
        archive_cmds_CXX='$CC -G -nostdlib $LDFLAGS
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
${wl}-h $wl$soname -o $lib'
        archive_expsym_cmds_CXX='echo "{ global:" > $lib.exp~cat
$export_symbols | $SED -e "s/\(.*\)/\1;/>" >> $lib.exp~echo "local: *;
};" >> $lib.exp~
        $CC -G -nostdlib ${wl}-M $wl$lib.exp -o $lib
$predep_objects $libobjs $deplibs $postdep_objects $compiler_flags~$RM
$lib.exp'

        # Commands to make compiler produce verbose output that
lists
        # what "hidden" libraries, object files and flags are
used when
        # linking a shared library.
        output_verbose_link_cmd='$CC -G $CFLAGS -v
conftest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'
fi

        hardcode_libdir_flag_spec_CXX='${wl}-R $wl$libdir'
        case $host_os in
solaris2.[0-5] | solaris2.[0-5].*) ;;
*)
        whole_archive_flag_spec_CXX='${wl}-z
${wl}allextract$convenience ${wl}-z ${wl}defaultextract'
        ;;
esac
fi
;;
esac
;;

sysv4*uw2* | sysv5OpenUNIX* | sysv5UnixWare7.[01].[10]* |
unixware7* | sco3.2v5.0.[024]*)
no_undefined_flag_CXX='${wl}-z,text'
archive_cmds_need_lc_CXX=no
hardcode_shlibpath_var_CXX=no
runpath_var='LD_RUN_PATH'

case $cc_basename in
CC*)
        archive_cmds_CXX='$CC -G ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
        archive_expsym_cmds_CXX='$CC -G ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
        ;;
*)
        archive_cmds_CXX='$CC -shared ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'

```

```

        archive_expsym_cmds_CXX='$CC -shared ${wl}-
Bexport:$export_symbols ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
        ;;
    esac
    ;;

    sysv5* | sco3.2v5* | sco5v6*)
# Note: We can NOT use -z defs as we might desire, because we do
not
# link with -lc, and that would cause any symbols used from libc
to
# always be unresolved, which means just about no library would
# ever link correctly.  If we're not using GNU ld we use -z text
# though, which does catch some bad symbols but isn't as heavy-
handed
# as -z defs.
no_undefined_flag_CXX='${wl}-z,text'
allow_undefined_flag_CXX='${wl}-z,nodefs'
archive_cmds_need_lc_CXX=no
hardcode_shlibpath_var_CXX=no
hardcode_libdir_flag_spec_CXX='${wl}-R,$libdir'
hardcode_libdir_separator_CXX=':'
link_all_deplibs_CXX=yes
export_dynamic_flag_spec_CXX='${wl}-Bexport'
runpath_var='LD_RUN_PATH'

    case $cc_basename in
        CC*)
            archive_cmds_CXX='$CC -G ${wl}-h,$soname -o $lib $libobjs
$deplibs $compiler_flags'
            archive_expsym_cmds_CXX='$CC -G ${wl}-Bexport:$export_symbols
${wl}-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
            old_archive_cmds_CXX='$CC -Tprelink_objects $oldobjs~
            "$old_archive_cmds_CXX"
            reload_cmds_CXX='$CC -Tprelink_objects $reload_objs~
            "$reload_cmds_CXX"
            ;;
        *)
            archive_cmds_CXX='$CC -shared ${wl}-h,$soname -o $lib
$libobjs $deplibs $compiler_flags'
            archive_expsym_cmds_CXX='$CC -shared ${wl}-
Bexport:$export_symbols ${wl}-h,$soname -o $lib $libobjs $deplibs
$compiler_flags'
            ;;
    esac
    ;;

    tandem*)
        case $cc_basename in
            NCC*)
                # NonStop-UX NCC 3.20

```

```

        # FIXME: insert proper C++ library support
        ld_shlibs_CXX=no
        ;;
        *)
        # FIXME: insert proper C++ library support
        ld_shlibs_CXX=no
        ;;
    esac
    ;;

vxworks*)
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
    ;;

*)
    # FIXME: insert proper C++ library support
    ld_shlibs_CXX=no
    ;;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ld_shlibs_CXX"
>&5
$as_echo "$ld_shlibs_CXX" >&6; }
test "$ld_shlibs_CXX" = no && can_build_shared=no

GCC_CXX="$GXX"
LD_CXX="$LD"

## CAVEAT EMPTOR:
## There is no encapsulation within the following macros, do not
change
## the running order or otherwise move them around unless you know
exactly
## what you are doing...
# Dependencies to place before and after the object being linked:
predep_objects_CXX=
postdep_objects_CXX=
predeps_CXX=
postdeps_CXX=
compiler_lib_search_path_CXX=

cat > conftest.$ac_ext <<_LT_EOF
class Foo
{
public:
    Foo (void) { a = 0; }
private:
    int a;
};
_LT_EOF

```

```

_lt_libdeps_save_CFLAGS=$CFLAGS
case "$CC $CFLAGS" in #(
*\ -flto*\ *) CFLAGS="$CFLAGS -fno-lto" ;;
*\ -fwhopr*\ *) CFLAGS="$CFLAGS -fno-whopr" ;;
*\ -fuse-linker-plugin*\ *) CFLAGS="$CFLAGS -fno-use-linker-plugin" ;;
esac

if { { eval echo "\$as_me\":${as_lineno-$LINENO}: \"$ac_compile\"";
} >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }; then
# Parse the compiler output and extract the necessary
# objects, libraries and library flags.

# Sentinel used to keep track of whether or not we are before
# the conftest object file.
pre_test_object_deps_done=no

for p in `eval "$output_verbose_link_cmd"`; do
  case ${prev}${p} in

    -L* | -R* | -l*)
      # Some compilers place space between "-{L,R}" and the path.
      # Remove the space.
      if test $p = "-L" ||
         test $p = "-R"; then
        prev=$p
        continue
      fi

      # Expand the sysroot to ease extracting the directories later.
      if test -z "$prev"; then
        case $p in
          -L*) func_stripname_cnf '-L' '' "$p"; prev=-L;
p=$func_stripname_result ;;
          -R*) func_stripname_cnf '-R' '' "$p"; prev=-R;
p=$func_stripname_result ;;
          -l*) func_stripname_cnf '-l' '' "$p"; prev=-l;
p=$func_stripname_result ;;
        esac
      fi
      case $p in
        =*) func_stripname_cnf '=' '' "$p";
p=$lt_sysroot$func_stripname_result ;;
      esac
      if test "$pre_test_object_deps_done" = no; then
        case ${prev}${p} in
          -L | -R)
            # Internal compiler library paths should come after those

```

```

        # provided the user.  The postdeps already come after the
        # user supplied libs so there is no need to process them.
        if test -z "$compiler_lib_search_path_CXX"; then
            compiler_lib_search_path_CXX="${prev}${p}"
        else

compiler_lib_search_path_CXX="${compiler_lib_search_path_CXX}
${prev}${p}"
        fi
        ;;
        # The "-l" case would never come before the object being
        # linked, so don't bother handling this case.
    esac
    else
        if test -z "$postdeps_CXX"; then
            postdeps_CXX="${prev}${p}"
        else
            postdeps_CXX="${postdeps_CXX} ${prev}${p}"
        fi
        fi
        prev=
        ;;

*.lto.$objext) ;; # Ignore GCC LTO objects
*.$objext)
    # This assumes that the test object file only shows up
    # once in the compiler output.
    if test "$p" = "conftest.$objext"; then
        pre_test_object_deps_done=yes
        continue
    fi

    if test "$pre_test_object_deps_done" = no; then
        if test -z "$predep_objects_CXX"; then
            predep_objects_CXX="$p"
        else
            predep_objects_CXX="$predep_objects_CXX $p"
        fi
    else
        if test -z "$postdep_objects_CXX"; then
            postdep_objects_CXX="$p"
        else
            postdep_objects_CXX="$postdep_objects_CXX $p"
        fi
        fi
        ;;

*) ;; # Ignore the rest.

    esac
done

```



```

    # Clean up.
    rm -f a.out a.exe
else
    echo "libtool.m4: error: problem compiling CXX test program"
fi

$RM -f confest.$objext
CFLAGS=$_lt_libdeps_save_CFLAGS

# PORTME: override above test on systems where it is broken
case $host_os in
interix[3-9]*)
    # Interix 3.5 installs completely hosed .la files for C++, so rather
    than
    # hack all around it, let's just trust "g++" to DTRT.
    predep_objects_CXX=
    postdep_objects_CXX=
    postdeps_CXX=
    ;;

linux*)
    case ` $CC -V 2>&1 | sed 5q ` in
    *Sun\ C*)
        # Sun C++ 5.9

        # The more standards-conforming stlport4 library is
        # incompatible with the Cstd library. Avoid specifying
        # it if it's in CXXFLAGS. Ignore libCrun as
        # -library=stlport4 depends on it.
        case " $CXX $CXXFLAGS " in
        *" -library=stlport4 ")
            solaris_use_stlport4=yes
            ;;
        esac

        if test "$solaris_use_stlport4" != yes; then
            postdeps_CXX='-library=Cstd -library=Crun'
        fi
        ;;
    esac
    ;;

solaris*)
    case $cc_basename in
    CC* | sunCC*)
        # The more standards-conforming stlport4 library is
        # incompatible with the Cstd library. Avoid specifying
        # it if it's in CXXFLAGS. Ignore libCrun as
        # -library=stlport4 depends on it.
        case " $CXX $CXXFLAGS " in
        *" -library=stlport4 ")
            solaris_use_stlport4=yes

```

```

    ;;
esac

# Adding this requires a known-good setup of shared libraries for
# Sun compiler versions before 5.6, else PIC objects from an old
# archive will be linked into the output, leading to subtle bugs.
if test "$solaris_use_stlport4" != yes; then
    postdeps_CXX='-library=Cstd -library=Crun'
fi
;;
esac
;;
esac

case " $postdeps_CXX " in
*" -lc "*) archive_cmds_need_lc_CXX=no ;;
esac
    compiler_lib_search_dirs_CXX=
if test -n "${compiler_lib_search_path_CXX}"; then
    compiler_lib_search_dirs_CXX=`echo " ${compiler_lib_search_path_CXX}"
| ${SED} -e 's! -L! !g' -e 's!^ !!'\`
fi

```

```

    lt_prog_compiler_wl_CXX=
lt_prog_compiler_pic_CXX=
lt_prog_compiler_static_CXX=

# C++ specific cases for pic, static, wl, etc.
if test "$GXX" = yes; then
    lt_prog_compiler_wl_CXX='-Wl,'
    lt_prog_compiler_static_CXX='-static'

case $host_os in
aix*)
    # All AIX code is PIC.
    if test "$host_cpu" = ia64; then
    # AIX 5 now supports IA64 processor
    lt_prog_compiler_static_CXX='-Bstatic'
    fi
    ;;

amigaos*)
    case $host_cpu in
    powerpc)
        # see comment about AmigaOS4 .so support
        lt_prog_compiler_pic_CXX='-fPIC'
        ;;
    m68k)
        # FIXME: we need at least 68020 code to build shared
libraries, but
        # adding the '-m68020' flag to GCC prevents building
anything better,
        # like '-m68040'.
        lt_prog_compiler_pic_CXX='-m68020 -resident32 -malways-
restore-a4'
        ;;
    esac
    ;;

beos* | irix5* | irix6* | nonstopux* | osf3* | osf4* | osf5*)
    # PIC is the default for these OSes.
    ;;

mingw* | cygwin* | os2* | pw32* | cegcc*)
    # This hack is so that the source file can tell whether it is
being
    # built for inclusion in a dll (and should export symbols for
example).
    # Although the cygwin gcc ignores -fPIC, still need this for
old-style
    # (--disable-auto-import) libraries
    lt_prog_compiler_pic_CXX='-DDLL_EXPORT'
    ;;

```

```

darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    lt_prog_compiler_pic_CXX='-fno-common'
    ;;
*djgpp*)
    # DJGPP does not support shared libraries at all
    lt_prog_compiler_pic_CXX=
    ;;
haiku*)
    # PIC is the default for Haiku.
    # The "-static" flag exists, but is broken.
    lt_prog_compiler_static_CXX=
    ;;
interix[3-9]*)
    # Interix 3.x gcc -fpic/-fPIC options generate broken code.
    # Instead, we relocate shared libraries at runtime.
    ;;
sysv4*MP*)
    if test -d /usr/nec; then
    lt_prog_compiler_pic_CXX=-Kconform_pic
    fi
    ;;
hpux*)
    # PIC is the default for 64-bit PA HP-UX, but not for 32-bit
    # PA HP-UX. On IA64 HP-UX, PIC is the default but the pic flag
    # sets the default TLS model and affects inlining.
    case $host_cpu in
    hppa*64*)
    ;;
    *)
    lt_prog_compiler_pic_CXX='-fPIC'
    ;;
    esac
    ;;
*qnx* | *nto*)
    # QNX uses GNU C++, but need to define -shared option too,
otherwise
    # it will coredump.
    lt_prog_compiler_pic_CXX='-fPIC -shared'
    ;;
*)
    lt_prog_compiler_pic_CXX='-fPIC'
    ;;
esac
else
case $host_os in
aix[4-9]*)
    # All AIX code is PIC.
    if test "$host_cpu" = ia64; then
        # AIX 5 now supports IA64 processor
        lt_prog_compiler_static_CXX='-Bstatic'

```

```

else
  lt_prog_compiler_static_CXX='-bnso -bI:/lib/syscalls.exp'
fi
;;
chorus*)
case $cc_basename in
cxch68*)
  # Green Hills C++ Compiler
  # _LT_TAGVAR(lt_prog_compiler_static, CXX)="--
no_auto_instantiation -u __main -u __premain -u _abort -r
$COOL_DIR/lib/libOrb.a $MVME_DIR/lib/CC/libC.a
$MVME_DIR/lib/classix/libcx.s.a"
  ;;
esac
;;
mingw* | cygwin* | os2* | pw32* | cegcc*)
# This hack is so that the source file can tell whether it is
being
# built for inclusion in a dll (and should export symbols for
example).
lt_prog_compiler_pic_CXX='-DDLL_EXPORT'
;;
dgux*)
case $cc_basename in
ec++*)
  lt_prog_compiler_pic_CXX='-KPIC'
  ;;
ghcx*)
  # Green Hills C++ Compiler
  lt_prog_compiler_pic_CXX='-pic'
  ;;
*)
  ;;
esac
;;
freebsd* | dragonfly*)
# FreeBSD uses GNU C++
;;
hpux9* | hpux10* | hpux11*)
case $cc_basename in
CC*)
  lt_prog_compiler_wl_CXX='-Wl,'
  lt_prog_compiler_static_CXX='${wl}-a ${wl}archive'
  if test "$host_cpu" != ia64; then
    lt_prog_compiler_pic_CXX='+Z'
  fi
  ;;
aCC*)
  lt_prog_compiler_wl_CXX='-Wl,'
  lt_prog_compiler_static_CXX='${wl}-a ${wl}archive'
  case $host_cpu in
  hppa*64*|ia64*)

```

```

        # +Z the default
        ;;
    *)
        lt_prog_compiler_pic_CXX='+Z'
        ;;
    esac
    ;;
    *)
        ;;
esac
;;
interix*)
# This is c89, which is MS Visual C++ (no shared libs)
# Anyone wants to do a port?
;;
irix5* | irix6* | nonstopux*)
case $cc_basename in
    CC*)
        lt_prog_compiler_wl_CXX='-Wl,'
        lt_prog_compiler_static_CXX='-non_shared'
        # CC pic flag -KPIC is the default.
        ;;
    *)
        ;;
esac
;;
linux* | k*bsd*-gnu | kopensolaris*-gnu)
case $cc_basename in
    KCC*)
        # KAI C++ Compiler
        lt_prog_compiler_wl_CXX='--backend -Wl,'
        lt_prog_compiler_pic_CXX='-fPIC'
        ;;
    ecpc* )
        # old Intel C++ for x86_64 which still supported -KPIC.
        lt_prog_compiler_wl_CXX='-Wl,'
        lt_prog_compiler_pic_CXX='-KPIC'
        lt_prog_compiler_static_CXX='-static'
        ;;
    icpc* )
        # Intel C++, used to be incompatible with GCC.
        # ICC 10 doesn't accept -KPIC any more.
        lt_prog_compiler_wl_CXX='-Wl,'
        lt_prog_compiler_pic_CXX='-fPIC'
        lt_prog_compiler_static_CXX='-static'
        ;;
    pgCC* | pgcpp*)
        # Portland Group C++ compiler
        lt_prog_compiler_wl_CXX='-Wl,'
        lt_prog_compiler_pic_CXX='-fpic'
        lt_prog_compiler_static_CXX='-Bstatic'
        ;;

```

```

cxx*)
  # Compaq C++
  # Make sure the PIC flag is empty.  It appears that all Alpha
  # Linux and Compaq Tru64 Unix objects are PIC.
  lt_prog_compiler_pic_CXX=
  lt_prog_compiler_static_CXX='-non_shared'
  ;;
xlc* | xlc* | bgxl[cC]* | mpixl[cC]*)
  # IBM XL 8.0, 9.0 on PPC and BlueGene
  lt_prog_compiler_wl_CXX='-Wl,'
  lt_prog_compiler_pic_CXX='-qpik'
  lt_prog_compiler_static_CXX='-qstaticlink'
  ;;
*)
  case ` $CC -V 2>&1 | sed 5q ` in
  *Sun\ C*)
    # Sun C++ 5.9
    lt_prog_compiler_pic_CXX='-KPIC'
    lt_prog_compiler_static_CXX='-Bstatic'
    lt_prog_compiler_wl_CXX='-Qoption ld '
    ;;
  esac
  ;;
esac
;;
lynxos*)
;;
m88k*)
;;
mvs*)
case $cc_basename in
  cxx*)
    lt_prog_compiler_pic_CXX='-W c,exportall'
    ;;
  *)
    ;;
esac
;;
netbsd*)
;;
*qnx* | *nto*)
  # QNX uses GNU C++, but need to define -shared option too,
otherwise
  # it will coredump.
  lt_prog_compiler_pic_CXX='-fPIC -shared'
  ;;
osf3* | osf4* | osf5*)
case $cc_basename in
  KCC*)
    lt_prog_compiler_wl_CXX='--backend -Wl,'
    ;;
  RCC*)

```

```

    # Rational C++ 2.4.1
    lt_prog_compiler_pic_CXX='-pic'
    ;;
cxx*)
    # Digital/Compaq C++
    lt_prog_compiler_wl_CXX='-Wl,'
    # Make sure the PIC flag is empty.  It appears that all Alpha
    # Linux and Compaq Tru64 Unix objects are PIC.
    lt_prog_compiler_pic_CXX=
    lt_prog_compiler_static_CXX='-non_shared'
    ;;
*)
    ;;
esac
;;
psos*)
;;
solaris*)
case $cc_basename in
  CC* | sunCC*)
    # Sun C++ 4.2, 5.x and Centerline C++
    lt_prog_compiler_pic_CXX='-KPIC'
    lt_prog_compiler_static_CXX='-Bstatic'
    lt_prog_compiler_wl_CXX='-Qoption ld '
    ;;
  gcx*)
    # Green Hills C++ Compiler
    lt_prog_compiler_pic_CXX='-PIC'
    ;;
  *)
    ;;
esac
;;
sunos4*)
case $cc_basename in
  CC*)
    # Sun C++ 4.x
    lt_prog_compiler_pic_CXX='-pic'
    lt_prog_compiler_static_CXX='-Bstatic'
    ;;
  lcc*)
    # Lucid
    lt_prog_compiler_pic_CXX='-pic'
    ;;
  *)
    ;;
esac
;;
sysv5* | unixware* | sco3.2v5* | sco5v6* | OpenUNIX*)
case $cc_basename in
  CC*)
    lt_prog_compiler_wl_CXX='-Wl,'

```



```

        lt_prog_compiler_pic_CXX='-KPIC'
        lt_prog_compiler_static_CXX='-Bstatic'
        ;;
    esac
    ;;
    tandem*)
    case $cc_basename in
        NCC*)
            # NonStop-UX NCC 3.20
            lt_prog_compiler_pic_CXX='-KPIC'
            ;;
        *)
            ;;
    esac
    ;;
    vxworks*)
    ;;
    *)
        lt_prog_compiler_can_build_shared_CXX=no
        ;;
    esac
fi

case $host_os in
    # For platforms which do not support PIC, -DPIC is meaningless:
    *djgpp*)
        lt_prog_compiler_pic_CXX=
        ;;
    *)
        lt_prog_compiler_pic_CXX="$lt_prog_compiler_pic_CXX -DPIC"
        ;;
    esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $compiler option
to produce PIC" >&5
$as_echo_n "checking for $compiler option to produce PIC... " >&6; }
if ${lt_cv_prog_compiler_pic_CXX+:} false; then :
    $as_echo_n "(cached) " >&6
else
    lt_cv_prog_compiler_pic_CXX=$lt_prog_compiler_pic_CXX
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic_CXX" >&5
$as_echo "$lt_cv_prog_compiler_pic_CXX" >&6; }
lt_prog_compiler_pic_CXX=$lt_cv_prog_compiler_pic_CXX

#
# Check to make sure the PIC flag actually works.
#
if test -n "$lt_prog_compiler_pic_CXX"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler PIC
flag $lt_prog_compiler_pic_CXX works" >&5

```

```

$as_echo_n "checking if $compiler PIC flag $lt_prog_compiler_pic_CXX
works... " >&6; }
if ${lt_cv_prog_compiler_pic_works_CXX+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_pic_works_CXX=no
  ac_outfile=confptest.$ac_objext
  echo "$lt_simple_compile_test_code" > confptest.$ac_ext
  lt_compiler_flag="$lt_prog_compiler_pic_CXX -DPIC"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "confptest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
  begins
  # with a dollar sign (not a hyphen), so the echo should work
  correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag :; t' \
-e 's: [^ ]*confptest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&5)
  (eval "$lt_compile" 2>confptest.err)
  ac_status=$?
  cat confptest.err >&5
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  if (exit $ac_status) && test -s "$ac_outfile"; then
    # The compiler can only warn and ignore the option if not
    recognized
    # So say no if there are warnings other than the usual output.
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >confptest.exp
    $SED '/^$/d; /^ *+/d' confptest.err >confptest.er2
    if test ! -s confptest.er2 || diff confptest.exp confptest.er2
>/dev/null; then
      lt_cv_prog_compiler_pic_works_CXX=yes
    fi
  fi
  $RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_pic_works_CXX" >&5
$as_echo "$lt_cv_prog_compiler_pic_works_CXX" >&6; }

if test x"$lt_cv_prog_compiler_pic_works_CXX" = xyes; then
  case $lt_prog_compiler_pic_CXX in
    "" | " ") ;;
    *) lt_prog_compiler_pic_CXX="$lt_prog_compiler_pic_CXX" ;;
  esac
else
  lt_prog_compiler_pic_CXX=
  lt_prog_compiler_can_build_shared_CXX=no
fi

```

```
fi
```

```
#
# Check to make sure the static flag actually works.
#
wl=$lt_prog_compiler_wl_CXX eval
lt_tmp_static_flag="\$lt_prog_compiler_static_CXX\"
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler static
flag $lt_tmp_static_flag works" >&5
$as_echo_n "checking if $compiler static flag $lt_tmp_static_flag
works... " >&6; }
if ${lt_cv_prog_compiler_static_works_CXX+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_static_works_CXX=no
  save_LDFLAGS="$LDFLAGS"
  LDFLAGS="$LDFLAGS $lt_tmp_static_flag"
  echo "$lt_simple_link_test_code" > conftest.$ac_ext
  if (eval $ac_link 2>conftest.err) && test -s conftest.$ac_exeext;
then
  # The linker can only warn and ignore the option if not
  recognized
  # So say no if there are warnings
  if test -s conftest.err; then
    # Append any errors to the config.log.
    cat conftest.err 1>&5
    $ECHO "$lt_linker_boilerplate" | $SED '/^$/d' > conftest.exp
    $SED '/^$/d; /^ *+/d' conftest.err >conftest.er2
    if diff conftest.exp conftest.er2 >/dev/null; then
      lt_cv_prog_compiler_static_works_CXX=yes
    fi
  else
    lt_cv_prog_compiler_static_works_CXX=yes
  fi
fi
$RM -r conftest*
LDFLAGS="$save_LDFLAGS"

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_static_works_CXX" >&5
$as_echo "$lt_cv_prog_compiler_static_works_CXX" >&6; }

if test x"$lt_cv_prog_compiler_static_works_CXX" = xyes; then
:
else
  lt_prog_compiler_static_CXX=
```

fi

```
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o_CXX+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_c_o_CXX=no
  $RM -r conftest 2>/dev/null
  mkdir conftest
  cd conftest
  mkdir out
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext

  lt_compiler_flag="-o out/conftest2.$ac_objext"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "conftest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\ :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*conftest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&5)
  (eval "$lt_compile" 2>out/conftest.err)
  ac_status=$?
  cat out/conftest.err >&5
  echo "$as_me:$LINENO: \$? = $ac_status" >&5
  if (exit $ac_status) && test -s out/conftest2.$ac_objext
  then
    # The compiler can only warn and ignore the option if not
recognized
    # So say no if there are warnings
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/conftest.exp
    $SED '/^$/d; /^ *+/d' out/conftest.err >out/conftest.er2
    if test ! -s out/conftest.er2 || diff out/conftest.exp
out/conftest.er2 >/dev/null; then
      lt_cv_prog_compiler_c_o_CXX=yes
    fi
  fi
fi
  chmod u+w . 2>&5
  $RM conftest*
  # SGI C++ compiler will create directory out/ii_files/ for
  # template instantiation
```

```

test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
$RM out/* && rmdir out
cd ..
$RM -r conftest
$RM conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o_CXX" >&5
$as_echo "$lt_cv_prog_compiler_c_o_CXX" >&6; }

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if $compiler
supports -c -o file.$ac_objext" >&5
$as_echo_n "checking if $compiler supports -c -o file.$ac_objext... "
>&6; }
if ${lt_cv_prog_compiler_c_o_CXX+:} false; then :
  $as_echo_n "(cached) " >&6
else
  lt_cv_prog_compiler_c_o_CXX=no
  $RM -r conftest 2>/dev/null
  mkdir conftest
  cd conftest
  mkdir out
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext

  lt_compiler_flag="-o out/conftest2.$ac_objext"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "conftest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and
begins
  # with a dollar sign (not a hyphen), so the echo should work
correctly.
  lt_compile=`echo "$ac_compile" | $SED \
-e 's:.*FLAGS}\{0,1}\} :&$lt_compiler_flag ;; t' \
-e 's: [^ ]*conftest\.: $lt_compiler_flag&; t' \
-e 's:$: $lt_compiler_flag:'`
  (eval echo "`echo \"$as_me:$LINENO: $lt_compile\"`" >&5)
  (eval "$lt_compile" 2>out/conftest.err)
  ac_status=$?
  cat out/conftest.err >&5
  echo "$as_me:$LINENO: `echo $ac_status`" >&5
  if (exit $ac_status) && test -s out/conftest2.$ac_objext
  then
    # The compiler can only warn and ignore the option if not
recognized
    # So say no if there are warnings
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >
out/conftest.exp
    $SED '/^$/d; /^ *+/d' out/conftest.err >out/conftest.er2

```

```

    if test ! -s out/confptest.er2 || diff out/confptest.exp
out/confptest.er2 >/dev/null; then
        lt_cv_prog_compiler_c_o_CXX=yes
    fi
    fi
    chmod u+w . 2>&5
    $RM confptest*
    # SGI C++ compiler will create directory out/ii_files/ for
    # template instantiation
    test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
    $RM out/* && rmdir out
    cd ..
    $RM -r confptest
    $RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_prog_compiler_c_o_CXX" >&5
$as_echo "$lt_cv_prog_compiler_c_o_CXX" >&6; }

hard_links="nottested"
if test "$lt_cv_prog_compiler_c_o_CXX" = no && test "$need_locks" !=
no; then
    # do not overwrite the value of need_locks provided by the user
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking if we can lock
with hard links" >&5
$as_echo_n "checking if we can lock with hard links... " >&6; }
    hard_links=yes
    $RM confptest*
    ln confptest.a confptest.b 2>/dev/null && hard_links=no
    touch confptest.a
    ln confptest.a confptest.b 2>&5 || hard_links=no
    ln confptest.a confptest.b 2>/dev/null && hard_links=no
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $hard_links" >&5
$as_echo "$hard_links" >&6; }
    if test "$hard_links" = no; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: \`$CC' does not
support \`-c -o', so \`make -j' may be unsafe" >&5
$as_echo "$as_me: WARNING: \`$CC' does not support \`-c -o', so \`make
-j' may be unsafe" >&2;}
        need_locks=warn
    fi
else
    need_locks=no
fi

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether the
$compiler linker ($LD) supports shared libraries" >&5
$as_echo_n "checking whether the $compiler linker ($LD) supports
shared libraries... " >&6; }

    export_symbols_cmds_CXX='$NM $libobjs $convenience |
$global_symbol_pipe | $SED '\''s/.*/\|'\'' | sort | uniq >
$export_symbols'
    exclude_expsyms_CXX='_GLOBAL_OFFSET_TABLE_|_GLOBAL__[FID]_.*'
    case $host_os in
aix[4-9]*)
    # If we're using GNU nm, then we don't want the "-C" option.
    # -C means demangle to AIX nm, but means don't demangle with GNU
nm
    # Also, AIX nm treats weak defined symbols like other global
defined
    # symbols, whereas GNU nm marks them as "W".
    if $NM -V 2>&1 | $GREP 'GNU' > /dev/null; then
        export_symbols_cmds_CXX='$NM -Bpg $libobjs $convenience | awk
'\''{ if (((\ $ 2 == "T") || (\ $ 2 == "D") || (\ $ 2 == "B") || (\ $ 2 ==
"W")) && (substr(\ $ 3,1,1) != ".")) { print \ $ 3 } }'\'' | sort -u >
$export_symbols'
    else
        export_symbols_cmds_CXX='$NM -BCpg $libobjs $convenience | awk
'\''{ if (((\ $ 2 == "T") || (\ $ 2 == "D") || (\ $ 2 == "B")) &&
(substr(\ $ 3,1,1) != ".")) { print \ $ 3 } }'\'' | sort -u >
$export_symbols'
    fi
    ;;
pw32*)
    export_symbols_cmds_CXX="$ltdll_cmds"
    ;;
cygwin* | mingw* | cegcc*)
    case $cc_basename in
cl*)

exclude_expsyms_CXX='_NULL_IMPORT_DESCRIPTOR|_IMPORT_DESCRIPTOR_.*'
        ;;
*)
        export_symbols_cmds_CXX='$NM $libobjs $convenience |
$global_symbol_pipe | $SED -e '\''/^([BCDGRS])[ ]/s/.*/[ ]\([^\ ]*\)/\1
DATA;/s/^\.[ ]__nm__\([^\ ]*\)/[ ]^\ ]*/\1 DATA;/^I[ ]/d;/^[AITW][
]/s/.*/\|'\'' | sort | uniq > $export_symbols'

exclude_expsyms_CXX='[_]+GLOBAL_OFFSET_TABLE_|[_]+GLOBAL__[FID]_.*|[_]
+head_[A-Za-z0-9_]+_dll|[A-Za-z0-9_]+_dll_iname'
        ;;
    esac
    ;;
*)

```

```

    export_symbols_cmds_CXX='$NM $libobjs $convenience |
$global_symbol_pipe | $SED '\''s/.* //'\' | sort | uniq >
$export_symbols'
    ;;
esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ld_shlibs_CXX" >&5
$as_echo "$ld_shlibs_CXX" >&6; }
test "$ld_shlibs_CXX" = no && can_build_shared=no

with_gnu_ld_CXX=$with_gnu_ld

#
# Do we need to explicitly link libc?
#
case "x$archive_cmds_need_lc_CXX" in
x|yes)
    # Assume -lc should be added
    archive_cmds_need_lc_CXX=yes

    if test "$enable_shared" = yes && test "$GCC" = yes; then
        case $archive_cmds_CXX in
        *'~'* )
            # FIXME: we may have to deal with multi-command sequences.
            ;;
        '$CC '* )
            # Test whether the compiler implicitly links with -lc since on
some
            # systems, -lgcc has to come before -lc. If gcc already passes -
lc
            # to ld, don't add -lc before -lgcc.
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether -lc
should be explicitly linked in" >&5
$as_echo_n "checking whether -lc should be explicitly linked in... "
>&6; }
if ${lt_cv_archive_cmds_need_lc_CXX+:} false; then :
    $as_echo_n "(cached) " >&6
else
    $RM conftest*
    echo "$lt_simple_compile_test_code" > conftest.$ac_ext

    if { { eval echo "\"\$as_me\":${as_lineno-$LINENO}:
\"$ac_compile\""; } >&5
(eval $ac_compile) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; } 2>conftest.err; then

```



```

soname=confptest
lib=confptest
libobjs=confptest.$ac_objext
deplibs=
wl=$lt_prog_compiler_wl_CXX
pic_flag=$lt_prog_compiler_pic_CXX
compiler_flags=-v
linker_flags=-v
verstring=
output_objdir=.
libname=confptest
lt_save_allow_undefined_flag=$allow_undefined_flag_CXX
allow_undefined_flag_CXX=
if { { eval echo "\"\$as_me\"":${as_lineno-$LINENO}:
\"$archive_cmds_CXX 2\>\&1 \|| $GREP \" -lc \" \>/dev/null 2\>\&1\""; }
>&5
(eval $archive_cmds_CXX 2\>\&1 \|| $GREP \" -lc \" \>/dev/null
2\>\&1) 2>&5
ac_status=$?
$as_echo "$as_me:${as_lineno-$LINENO}: \ $? = $ac_status" >&5
test $ac_status = 0; }
then
    lt_cv_archive_cmds_need_lc_CXX=no
else
    lt_cv_archive_cmds_need_lc_CXX=yes
fi
allow_undefined_flag_CXX=$lt_save_allow_undefined_flag
else
    cat confptest.err 1>&5
fi
$RM confptest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$lt_cv_archive_cmds_need_lc_CXX" >&5
$as_echo "$lt_cv_archive_cmds_need_lc_CXX" >&6; }
    archive_cmds_need_lc_CXX=$lt_cv_archive_cmds_need_lc_CXX
    ;;
esac
fi
;;
esac

```



```

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking dynamic linker
characteristics" >&5
$as_echo_n "checking dynamic linker characteristics... " >&6; }

library_names_spec=
libname_spec='lib$name'
soname_spec=
shrext_cmds=".so"
postinstall_cmds=
postuninstall_cmds=
finish_cmds=
finish_eval=
shlibpath_var=
shlibpath_overrides_runpath=unknown
version_type=none
dynamic_linker="$host_os ld.so"
sys_lib_dlsearch_path_spec="/lib /usr/lib"
need_lib_prefix=unknown
hardcode_into_libs=no

# when you set need_version to no, make sure it does not cause -
set_version
# flags to be left without arguments
need_version=unknown

case $host_os in
aix3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname.a'
    shlibpath_var=LIBPATH

    # AIX 3 has no versioning support, so we append a major version to
the name.
    soname_spec='${libname}${release}${shared_ext}$major'
    ;;

aix[4-9]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    hardcode_into_libs=yes
    if test "$host_cpu" = ia64; then
        # AIX 5 supports IA64
        library_names_spec='${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext}$versuffix $libname${shared_ext}'
        shlibpath_var=LD_LIBRARY_PATH
    else
        # With GCC up to 2.95.x, collect2 would create an import file
# for dependence libraries.  The import file would start with

```

```

# the line `#! .' . This would cause the generated library to
# depend on `.', always an invalid library. This was fixed in
# development snapshots of GCC prior to 3.0.
case $host_os in
  aix4 | aix4.[01] | aix4.[01].*)
    if { echo '#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__
>= 97)';
      echo ' yes '
      echo '#endif'; } | ${CC} -E - | $GREP yes > /dev/null; then
      :
    else
      can_build_shared=no
    fi
  ;;
esac
# AIX (on Power*) has no versioning support, so currently we can
not hardcode correct
# soname into executable. Probably we can add versioning support
to
# collect2, so additional links can be useful in future.
if test "$aix_use_runtimelinking" = yes; then
  # If using run time linking (on AIX 4.2 or later) use
lib<name>.so
  # instead of lib<name>.a to let people know that these are not
  # typical AIX shared libraries.
  library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
else
  # We preserve .a as extension for shared libraries through
AIX4.2
  # and later when we are not doing run time linking.
  library_names_spec='${libname}${release}.a $libname.a'
  soname_spec='${libname}${release}${shared_ext}$major'
fi
shlibpath_var=LIBPATH
fi
;;

amigaos*)
case $host_cpu in
  powerpc)
    # Since July 2007 AmigaOS4 officially supports .so libraries.
    # When compiling the executable, add -use-dynld -Lsobjs: to the
    compileline.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    ;;
  m68k)
    library_names_spec='$libname.ixlibrary $libname.a'
    # Create ${libname}_ixlibrary.a entries in /sys/libs.
    finish_eval='for lib in `ls $libdir/*.ixlibrary 2>/dev/null`; do
libname=`func_echo_all "$lib" | $SED

```

```

'\''s%^.*\/\([^\/]*\)\.ixlibrary$%\1%\''`; test $RM
/sys/libs/${libname}_ixlibrary.a; $show "cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a"; cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a || exit 1; done'
;;
esac
;;

beos*)
    library_names_spec='${libname}${shared_ext}'
    dynamic_linker="$host_os ld.so"
    shlibpath_var=LIBRARY_PATH
;;

bsdi[45]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    sys_lib_search_path_spec="/shlib /usr/lib /usr/X11/lib
/usr/contrib/lib /lib /usr/local/lib"
    sys_lib_dlsearch_path_spec="/shlib /usr/lib /usr/local/lib"
    # the default ld.so.conf also contains /usr/contrib/lib and
    # /usr/X11R6/lib (/usr/X11 is a link to /usr/X11R6), but let us
allow
    # libtool to hard-code these into programs
;;

cygwin* | mingw* | pw32* | cegcc*)
    version_type=windows
    shrext_cmds=".dll"
    need_version=no
    need_lib_prefix=no

    case $GCC,$cc_basename in
    yes,*)
        # gcc
        library_names_spec='$libname.dll.a'
        # DLL is installed to $(libdir)/../bin by postinstall_cmds
        postinstall_cmds='base_file=`basename \${file}`~
        dlpath=`$SHELL 2>&1 -c \'\'. $dir/\''\${base_file}\'\'i; echo
\${dlname}\'\'`~
        dldir=$destdir/`dirname \${dlpath}`~
        test -d \${dldir} || mkdir -p \${dldir}~
        $install_prog $dir/\${dlname} \${dldir}/\${dlname}~
        chmod a+x \${dldir}/\${dlname}~
        if test -n \'\'$striplib\'\' && test -n \'\'$striplib\'\' ; then
            eval \'\'$striplib \${dldir}/\${dlname}\'\' || exit \${?};

```

```

    fi'
    postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\`'. $file; echo
\${dldllname}`'\`~
    dlpath=$dir/\${dldll}~
    $RM \${dlpath}'
    shlibpath_overrides_runpath=yes

case $host_os in
cygwin*)
    # Cygwin DLLs use 'cyg' prefix rather than 'lib'
    soname_spec=`echo ${libname} | sed -e 's/^lib/cyg/'``echo
${release} | $SED -e 's/[.]/-/g'`${versuffix}${shared_ext}'

    ;;
mingw* | cegcc*)
    # MinGW DLLs use traditional 'lib' prefix
    soname_spec='${libname}`echo ${release} | $SED -e 's/[.]/-/
/g'`${versuffix}${shared_ext}'

    ;;
pw32*)
    # pw32 DLLs use 'pw' prefix rather than 'lib'
    library_names_spec=`echo ${libname} | sed -e 's/^lib/pw/'``echo
${release} | $SED -e 's/[.]/-/g'`${versuffix}${shared_ext}'

    ;;
esac
dynamic_linker='Win32 ld.exe'

;;

*,cl*)
    # Native MSVC
    libname_spec='$name'
    soname_spec='${libname}`echo ${release} | $SED -e 's/[.]/-/
/g'`${versuffix}${shared_ext}'
    library_names_spec='${libname}.dll.lib'

case $build_os in
mingw*)
    sys_lib_search_path_spec=
    lt_save_ifs=$IFS
    IFS=';'
    for lt_path in $LIB
    do
        IFS=$lt_save_ifs
        # Let DOS variable expansion print the short 8.3 style file
name.
        lt_path=`cd "$lt_path" 2>/dev/null && cmd //C "for %i in (".")
do @echo %~si"`
        sys_lib_search_path_spec="$sys_lib_search_path_spec $lt_path"
    done
    IFS=$lt_save_ifs
    # Convert to MSYS style.

```

```

        sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
sed -e 's|\\|/|g' -e 's| \|[a-zA-Z]|:|/|g' -e 's|^|'|`
        ;;
    cygwin*)
        # Convert to unix form, then to dos form, then back to unix form
        # but this time dos style (no spaces!) so that the unix form
looks
        # like /cygdrive/c/PROGRA~1:/cygdr...
        sys_lib_search_path_spec=`cygpath --path --unix "$LIB"`
        sys_lib_search_path_spec=`cygpath --path --dos
"$sys_lib_search_path_spec" 2>/dev/null`
        sys_lib_search_path_spec=`cygpath --path --unix
"$sys_lib_search_path_spec" | $SED -e "s/$PATH_SEPARATOR/ /g"`
        ;;
    *)
        sys_lib_search_path_spec="$LIB"
        if $ECHO "$sys_lib_search_path_spec" | $GREP '[c-zC-Z]:/'
>/dev/null; then
            # It is most probably a Windows format PATH.
            sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
$SED -e 's/;/ /g'`
        else
            sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" |
$SED -e "s/$PATH_SEPARATOR/ /g"`
        fi
        # FIXME: find the short name or the path components, as spaces
are
        # common. (e.g. "Program Files" -> "PROGRA~1")
        ;;
    esac

    # DLL is installed to $(libdir)/../bin by postinstall_cmds
    postinstall_cmds='base_file=`basename \${file}`~
dldir=`$SHELL 2>&1 -c '\''. $dir/\'''\${base_file}'\'''\`i; echo
\${dldir}'\'''\`~
dldir=$destdir/`dirname \${dldir}`~
test -d \${dldir} || mkdir -p \${dldir}~
$install_prog $dir/\${dldir} \${dldir}/\${dldir}'
    postuninstall_cmds='dldir=`$SHELL 2>&1 -c '\''. $file; echo
\${dldir}'\'''\`~
dldir=$dir/\${dldir}~
$RM \${dldir}'
    shlibpath_overrides_runpath=yes
    dynamic_linker='Win32 link.exe'
    ;;

    *)
        # Assume MSVC wrapper
        library_names_spec='${libname}`echo ${release} | $SED -e 's/[.]/-/g'`${versuffix}${shared_ext} $libname.lib'
        dynamic_linker='Win32 ld.exe'
    ;;

```

```

    esac
    # FIXME: first we should search . and the directory the executable
    is in
    shlibpath_var=PATH
    ;;

darwin* | rhapsody*)
    dynamic_linker="$host_os dyld"
    version_type=darwin
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${major}$shared_ext
${libname}$shared_ext'
    soname_spec='${libname}${release}${major}$shared_ext'
    shlibpath_overrides_runpath=yes
    shlibpath_var=DYLD_LIBRARY_PATH
    shrext_cmds=`test ".$module" = .yes && echo .so || echo .dylib`

    sys_lib_dlsearch_path_spec='/usr/local/lib /lib /usr/lib'
    ;;

dgux*)
    version_type=linux # correct to gnu/linux during the next big
    refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname$shared_ext'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    ;;

freebsd* | dragonfly*)
    # DragonFly does not have aout.  When/if they implement a new
    # versioning mechanism, adjust this.
    if test -x /usr/bin/objformat; then
        objformat=`/usr/bin/objformat`
    else
        case $host_os in
            freebsd[23].*) objformat=aout ;;
            *) objformat=elf ;;
        esac
    fi
    version_type=freebsd-$objformat
    case $version_type in
        freebsd-elf*)
            library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext} $libname${shared_ext}'
            need_version=no
            need_lib_prefix=no
            ;;
        freebsd-*)

```



```

        library_names_spec='${libname}${release}${shared_ext}$versuffix
$libname${shared_ext}$versuffix'
        need_version=yes
        ;;
    esac
    shlibpath_var=LD_LIBRARY_PATH
    case $host_os in
    freebsd2.*)
        shlibpath_overrides_runpath=yes
        ;;
    freebsd3.[01]* | freebsdelf3.[01]*)
        shlibpath_overrides_runpath=yes
        hardcode_into_libs=yes
        ;;
    freebsd3.[2-9]* | freebsdelf3.[2-9]* | \
    freebsd4.[0-5] | freebsdelf4.[0-5] | freebsd4.1.1 | freebsdelf4.1.1)
        shlibpath_overrides_runpath=no
        hardcode_into_libs=yes
        ;;
    *) # from 4.6 on, and DragonFly
        shlibpath_overrides_runpath=yes
        hardcode_into_libs=yes
        ;;
    esac
    ;;

gnu*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;

haiku*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    dynamic_linker="$host_os runtime_loader"
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}${major} ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    sys_lib_dlsearch_path_spec='/boot/home/config/lib /boot/common/lib
/boot/system/lib'

```

```

hardcode_into_libs=yes
;;

hpux9* | hpux10* | hpux11*)
  # Give a soname corresponding to the major version so that dld.sl
  refuses to
  # link against other versions.
  version_type=sunos
  need_lib_prefix=no
  need_version=no
  case $host_cpu in
  ia64*)
    shrext_cmds='.so'
    hardcode_into_libs=yes
    dynamic_linker="$host_os dld.so"
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    if test "X$HPUX_IA64_MODE" = X32; then
      sys_lib_search_path_spec="/usr/lib/hpux32 /usr/local/lib/hpux32
/usr/local/lib"
    else
      sys_lib_search_path_spec="/usr/lib/hpux64 /usr/local/lib/hpux64"
    fi
    sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
  hppa*64*)
    shrext_cmds='.sl'
    hardcode_into_libs=yes
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=LD_LIBRARY_PATH # How should we handle SHLIB_PATH
    shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    sys_lib_search_path_spec="/usr/lib/pa20_64 /usr/ccs/lib/pa20_64"
    sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
  *)
    shrext_cmds='.sl'
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=SHLIB_PATH
    shlibpath_overrides_runpath=no # +s is required to enable
SHLIB_PATH
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    ;;
  esac

```

```

# HP-UX runs *really* slowly unless shared libraries are mode 555,
...
postinstall_cmds='chmod 555 $lib'
# or fails outright, so override atomically:
install_override_mode=555
;;

interix[3-9]*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    dynamic_linker='Interix 3.x ld.so.1 (PE, like ELF)'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;

irix5* | irix6* | nonstopux*)
    case $host_os in
        nonstopux*) version_type=nonstopux ;;
        *)
            if test "$lt_cv_prog_gnu_ld" = yes; then
                version_type=linux # correct to gnu/linux during the next
big refactor
            else
                version_type=irix
            fi ;;
    esac
    need_lib_prefix=no
    need_version=no
    soname_spec='${libname}${release}${shared_ext}$major'
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major
${libname}${release}${shared_ext} $libname${shared_ext}'
    case $host_os in
        irix5* | nonstopux*)
            libsuff= shlibsuff=
            ;;
        *)
            case $LD in # libtool.m4 will add one of these switches to LD
*-32|*" -32 " | *-melf32bsmip|*" -melf32bsmip ")
                libsuff= shlibsuff= libmagic=32-bit;;
*-n32|*" -n32 " | *-melf32bmipn32|*" -melf32bmipn32 ")
                libsuff=32 shlibsuff=N32 libmagic=N32;;
*-64|*" -64 " | *-melf64bmip|*" -melf64bmip ")
                libsuff=64 shlibsuff=64 libmagic=64-bit;;
*) libsuff= shlibsuff= libmagic=never-match;;
            esac
    esac

```

```

    ;;
    esac
    shlibpath_var=LD_LIBRARY${shlibsuff}_PATH
    shlibpath_overrides_runpath=no
    sys_lib_search_path_spec="/usr/lib${libsuff} /lib${libsuff}
/usr/local/lib${libsuff}"
    sys_lib_dlsearch_path_spec="/usr/lib${libsuff} /lib${libsuff}"
    hardcode_into_libs=yes
    ;;

# No shared lib support for Linux oldld, aout, or coff.
linux*oldld* | linux*aout* | linux*coff*)
    dynamic_linker=no
    ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig -n $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no

    # Some binutils ld are patched to set DT_RUNPATH
    if ${lt_cv_shlibpath_overrides_runpath+:} false; then :
    $as_echo_n "(cached) " >&6
    else
    lt_cv_shlibpath_overrides_runpath=no
    save_LDFLAGS=$LDFLAGS
    save_libdir=$libdir
    eval "libdir=/foo; wl=\"\$lt_prog_compiler_wl_CXX\"; \
LDFLAGS=\"\$LDFLAGS $hardcode_libdir_flag_spec_CXX\""
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_cxx_try_link "$LINENO"; then :
    if `($OBJDUMP -p conftest$ac_exeext) 2>/dev/null | grep
"RUNPATH.*$libdir" >/dev/null; then :
        lt_cv_shlibpath_overrides_runpath=yes

```

```

fi
fi
rm -f core conftest.err conftest.$sac_objext \
    conftest$sac_exeext conftest.$sac_ext
    LDFLAGS=$save_LDFLAGS
    libdir=$save_libdir

fi

shlibpath_overrides_runpath=$lt_cv_shlibpath_overrides_runpath

# This implies no fast_install, which is unacceptable.
# Some rework will be needed to allow for fast_install
# before this can be enabled.
hardcode_into_libs=yes

# Append ld.so.conf contents to the search path
if test -f /etc/ld.so.conf; then
    lt_ld_extra=`awk '/^include / { system(sprintf("cd /etc; cat %s
2>/dev/null", \$$2)); skip = 1; } { if (!skip) print \$$0; skip = 0; }'
< /etc/ld.so.conf | $SED -e 's/#.*//;/^[		]*hwcap[		]/d;s/[:,/ ]//
/g;s/=[^=]*$///;s/=[^= ]* / /g;s/"//g;/^$/d' | tr '\n' ' '`
    sys_lib_dlsearch_path_spec="/lib /usr/lib $lt_ld_extra"
fi

# We used to test for /lib/ld.so.1 and disable shared libraries on
# powerpc, because MkLinux only supported shared libraries with the
# GNU dynamic linker.  Since this was broken with cross compilers,
# most powerpc-linux boxes support dynamic linking these days and
# people can always --disable-shared, the test was removed, and we
# assume the GNU/Linux dynamic linker is in use.
dynamic_linker='GNU/Linux ld.so'
;;

netbsd*)
version_type=sunos
need_lib_prefix=no
need_version=no
if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
    dynamic_linker='NetBSD (a.out) ld.so'
else
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    dynamic_linker='NetBSD ld.elf_so'
fi
shlibpath_var=LD_LIBRARY_PATH
shlibpath_overrides_runpath=yes
hardcode_into_libs=yes

```

```

;;

newsos6)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    ;;

*nto* | *qnx*)
    version_type=qnx
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    dynamic_linker='ldqnx.so'
    ;;

openbsd*)
    version_type=sunos
    sys_lib_dlsearch_path_spec="/usr/lib"
    need_lib_prefix=no
    # Some older versions of OpenBSD (3.3 at least) *do* need versioned
libs.
    case $host_os in
        openbsd3.3 | openbsd3.3.*)    need_version=yes ;;
        *)                            need_version=no  ;;
    esac
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`" || test
"$host_os-$host_cpu" = "openbsd2.8-powerpc"; then
        case $host_os in
            openbsd2.[89] | openbsd2.[89].*)
                shlibpath_overrides_runpath=no
                ;;
            *)
                shlibpath_overrides_runpath=yes
                ;;
        esac
    else
        shlibpath_overrides_runpath=yes
    fi
    ;;

```

```

os2*)
    libname_spec='$name'
    shrext_cmds=".dll"
    need_lib_prefix=no
    library_names_spec='$libname${shared_ext} $libname.a'
    dynamic_linker='OS/2 ld.exe'
    shlibpath_var=LIBPATH
    ;;

osf3* | osf4* | osf5*)
    version_type=osf
    need_lib_prefix=no
    need_version=no
    soname_spec='${libname}${release}${shared_ext}$major'
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    shlibpath_var=LD_LIBRARY_PATH
    sys_lib_search_path_spec="/usr/shlib /usr/ccs/lib /usr/lib/cmplrs/cc
/usr/lib /usr/local/lib /var/shlib"
    sys_lib_dlsearch_path_spec="$sys_lib_search_path_spec"
    ;;

rdos*)
    dynamic_linker=no
    ;;

solaris*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    # ldd complains unless libraries are executable
    postinstall_cmds='chmod +x $lib'
    ;;

sunos4*)
    version_type=sunos
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${shared_ext}$versuffix'
    finish_cmds='PATH="\$PATH:/usr/etc" ldconfig $libdir'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    if test "$with_gnu_ld" = yes; then
        need_lib_prefix=no
    fi

```

```

    need_version=yes
    ;;

sysv4 | sysv4.3*)
    version_type=linux # correct to gnu/linux during the next big
refactor
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    case $host_vendor in
        sni)
            shlibpath_overrides_runpath=no
            need_lib_prefix=no
            runpath_var=LD_RUN_PATH
            ;;
        siemens)
            need_lib_prefix=no
            ;;
        motorola)
            need_lib_prefix=no
            need_version=no
            shlibpath_overrides_runpath=no
            sys_lib_search_path_spec='/lib /usr/lib /usr/ccs/lib'
            ;;
    esac
    ;;

sysv4*MP*)
    if test -d /usr/nec ;then
        version_type=linux # correct to gnu/linux during the next big
refactor
        library_names_spec='$libname${shared_ext}.$versuffix
$libname${shared_ext}.$major $libname${shared_ext}'
        soname_spec='$libname${shared_ext}.$major'
        shlibpath_var=LD_LIBRARY_PATH
    fi
    ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
    version_type=freebsd-elf
    need_lib_prefix=no
    need_version=no
    library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext} $libname${shared_ext}'
    soname_spec='${libname}${release}${shared_ext}$major'
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    if test "$with_gnu_ld" = yes; then
        sys_lib_search_path_spec='/usr/local/lib /usr/gnu/lib /usr/ccs/lib
/usr/lib /lib'
    fi

```



```

else
  sys_lib_search_path_spec='/usr/ccs/lib /usr/lib'
  case $host_os in
    sco3.2v5*)
      sys_lib_search_path_spec="$sys_lib_search_path_spec /lib"
      ;;
    esac
  fi
  sys_lib_dlsearch_path_spec='/usr/lib'
  ;;

tpf*)
  # TPF is a cross-target only. Preferred cross-host = GNU/Linux.
  version_type=linux # correct to gnu/linux during the next big
  refactor
  need_lib_prefix=no
  need_version=no
  library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
  shlibpath_var=LD_LIBRARY_PATH
  shlibpath_overrides_runpath=no
  hardcode_into_libs=yes
  ;;

uts4*)
  version_type=linux # correct to gnu/linux during the next big
  refactor
  library_names_spec='${libname}${release}${shared_ext}$versuffix
${libname}${release}${shared_ext}$major $libname${shared_ext}'
  soname_spec='${libname}${release}${shared_ext}$major'
  shlibpath_var=LD_LIBRARY_PATH
  ;;

*)
  dynamic_linker=no
  ;;
esac
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dynamic_linker" >&5
$as_echo "$dynamic_linker" >&6; }
test "$dynamic_linker" = no && can_build_shared=no

variables_saved_for_relink="PATH $shlibpath_var $runpath_var"
if test "$GCC" = yes; then
  variables_saved_for_relink="$variables_saved_for_relink
GCC_EXEC_PREFIX COMPILER_PATH LIBRARY_PATH"
fi

if test "${lt_cv_sys_lib_search_path_spec+set}" = set; then
  sys_lib_search_path_spec="$lt_cv_sys_lib_search_path_spec"
fi
if test "${lt_cv_sys_lib_dlsearch_path_spec+set}" = set; then
  sys_lib_dlsearch_path_spec="$lt_cv_sys_lib_dlsearch_path_spec"

```

fi

```
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking how to hardcode
library paths into programs" >&5
$as_echo_n "checking how to hardcode library paths into programs... "
>&6; }
hardcode_action_CXX=
if test -n "$hardcode_libdir_flag_spec_CXX" ||
    test -n "$runpath_var_CXX" ||
    test "X$hardcode_automatic_CXX" = "Xyes" ; then

# We can hardcode non-existent directories.
if test "$hardcode_direct_CXX" != no &&
    # If the only mechanism to avoid hardcoding is shlibpath_var, we
```

```

        # have to relink, otherwise we might link with an installed
library
        # when we should be linking with a yet-to-be-installed one
        ## test "$_LT_TAGVAR(hardcode_shlibpath_var, CXX)" != no &&
        test "$hardcode_minus_L_CXX" != no; then
        # Linking always hardcodes the temporary library directory.
        hardcode_action_CXX=relink
    else
        # We can link without hardcoding, and we can hardcode nonexisting
dirs.
        hardcode_action_CXX=immediate
    fi
else
        # We cannot hardcode anything, or else we can only hardcode existing
        # directories.
        hardcode_action_CXX=unsupported
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $hardcode_action_CXX"
>&5
$as_echo "$hardcode_action_CXX" >&6; }

if test "$hardcode_action_CXX" = relink ||
    test "$inherit_rpath_CXX" = yes; then
    # Fast installation is not supported
    enable_fast_install=no
elif test "$shlibpath_overrides_runpath" = yes ||
    test "$enable_shared" = no; then
    # Fast installation is not necessary
    enable_fast_install=needless
fi

fi # test -n "$compiler"

CC=$lt_save_CC
CFLAGS=$lt_save_CFLAGS
LDCXX=$LD
LD=$lt_save_LD
GCC=$lt_save_GCC
with_gnu_ld=$lt_save_with_gnu_ld
lt_cv_path_LDCXX=$lt_cv_path_LD
lt_cv_path_LD=$lt_save_path_LD
lt_cv_prog_gnu_ldcxx=$lt_cv_prog_gnu_ld
lt_cv_prog_gnu_ld=$lt_save_with_gnu_ld
fi # test "$lt_caught_CXX_error" != yes

ac_ext=c

```

```
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu
```

```
ac_config_commands="$ac_config_commands libtool"
```

```
# Only expand once:
```

```
# Check whether --enable-compiler-coverage was given.
if test "${enable_compiler_coverage+set}" = set; then :
  enableval=$enable_compiler_coverage; if test
"x$enable_compiler_coverage" = "xyes"; then
    if test "x$GCC" = "xyes"; then
      CFLAGS="$CFLAGS -fprofile-arcs -ftest-coverage"
    fi
  fi
fi
```

```
# Check whether --enable-compiler-optimisations was given.
if test "${enable_compiler_optimisations+set}" = set; then :
  enableval=$enable_compiler_optimisations; if test
"x$enable_compiler_optimisations" = "xno"; then
    CFLAGS=`echo "$CFLAGS" | sed -e "s/ -O[1-9]*\b/ -O0/g"`
  fi
fi
```

```
if test "x$ac_cv_env_PKG_CONFIG_set" != "xset"; then
  if test -n "$ac_tool_prefix"; then
```

```

# Extract the first word of "${ac_tool_prefix}pkg-config", so it can
be a program name with args.
set dummy ${ac_tool_prefix}pkg-config; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_PKG_CONFIG+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $PKG_CONFIG in
  [\\/] * | ?:[\\/] *)
    ac_cv_path_PKG_CONFIG="$PKG_CONFIG" # Let the user override the test
with a path.
    ;;
  *)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
IFS=$as_save_IFS

    ;;
  esac
fi
PKG_CONFIG=$ac_cv_path_PKG_CONFIG
if test -n "$PKG_CONFIG"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $PKG_CONFIG" >&5
$as_echo "$PKG_CONFIG" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_path_PKG_CONFIG"; then
  ac_pt_PKG_CONFIG=$PKG_CONFIG
  # Extract the first word of "pkg-config", so it can be a program
name with args.
  set dummy pkg-config; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_path_ac_pt_PKG_CONFIG+:} false; then :

```

```

    $as_echo_n "(cached) " >&6
else
  case $ac_pt_PKG_CONFIG in
    [\\/* | ?:[\\/*]*)
      ac_cv_path_ac_pt_PKG_CONFIG="$ac_pt_PKG_CONFIG" # Let the user
      override the test with a path.
      ;;
    *)
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ''$ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_path_ac_pt_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      ;;
    esac
  fi
  ac_pt_PKG_CONFIG=$ac_cv_path_ac_pt_PKG_CONFIG
  if test -n "$ac_pt_PKG_CONFIG"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_pt_PKG_CONFIG"
    >&5
    $as_echo "$ac_pt_PKG_CONFIG" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
    $as_echo "no" >&6; }
  fi

  if test "x$ac_pt_PKG_CONFIG" = x; then
    PKG_CONFIG=""
  else
    case $cross_compiling:$ac_tool_warned in
    yes:)
      { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
      not prefixed with host triplet" >&5
      $as_echo "$as_me: WARNING: using cross tools not prefixed with host
      triplet" >&2;}
      ac_tool_warned=yes ;;
    esac
    PKG_CONFIG=$ac_pt_PKG_CONFIG
  fi
else
  PKG_CONFIG="$ac_cv_path_PKG_CONFIG"

```

```

fi

fi
if test -n "$PKG_CONFIG"; then
    _pkg_min_version=0.9.0
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking pkg-config is
at least version $_pkg_min_version" >&5
$as_echo_n "checking pkg-config is at least version
$_pkg_min_version... " >&6; }
    if $PKG_CONFIG --atleast-pkgconfig-version $_pkg_min_version;
then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        PKG_CONFIG=""
    fi
fi

fi

# Initialize libtool

if test -n "$ac_tool_prefix"; then
    # Extract the first word of "${ac_tool_prefix}windres", so it can be
a program name with args.
set dummy ${ac_tool_prefix}windres; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_RC+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -n "$RC"; then
        ac_cv_prog_RC="$RC" # Let the user override the test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_RC="${ac_tool_prefix}windres"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$as_save_IFS

fi
fi

```

```

fi
RC=$ac_cv_prog_RC
if test -n "$RC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $RC" >&5
$as_echo "$RC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_RC"; then
  ac_ct_RC=$RC
  # Extract the first word of "windres", so it can be a program name
  with args.
  set dummy windres; ac_word=$2
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
  if ${ac_cv_prog_ac_ct_RC+:} false; then :
    $as_echo_n "(cached) " >&6
  else
    if test -n "$ac_ct_RC"; then
      ac_cv_prog_ac_ct_RC="$ac_ct_RC" # Let the user override the test.
    else
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_prog_ac_ct_RC="windres"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS
    fi
  fi
  ac_ct_RC=$ac_cv_prog_ac_ct_RC
  if test -n "$ac_ct_RC"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_RC" >&5
$as_echo "$ac_ct_RC" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  fi
fi

```



```

    if test "x$ac_ct_RC" = x; then
        RC=""
    else
        case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
        RC=$ac_ct_RC
    fi
else
    RC="$ac_cv_prog_RC"
fi

# Source file extension for RC test sources.
ac_ext=rc

# Object file extension for compiled RC test sources.
objext=o
objext_RC=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code='sample MENU { MENUITEM "&Soup", 100,
CHECKED }'

# Code to be used in simple link tests
lt_simple_link_test_code="$lt_simple_compile_test_code"

# ltmain only uses $CC for tagged configurations so make sure $CC is
set.

# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC

```

```

# save warnings/boilerplate of simple test code
ac_outfile=confptest.$ac_objext
echo "$lt_simple_compile_test_code" >confptest.$ac_ext
eval "$ac_compile" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d'
>confptest.err
_lt_compiler_boilerplate=`cat confptest.err`
$RM confptest*

ac_outfile=confptest.$ac_objext
echo "$lt_simple_link_test_code" >confptest.$ac_ext
eval "$ac_link" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d' >confptest.err
_lt_linker_boilerplate=`cat confptest.err`
$RM -r confptest*

# Allow CC to be a program name with arguments.
lt_save_CC="$CC"
lt_save_CFLAGS=$CFLAGS
lt_save_GCC=$GCC
GCC=
CC=${RC-"windres"}
CFLAGS=
compiler=$CC
compiler_RC=$CC
for cc_temp in $compiler""; do
  case $cc_temp in
    compile | *[\//]compile | ccache | *[\//]ccache ) ;;
    distcc | *[\//]distcc | purify | *[\//]purify ) ;;
    \-*) ;;
    *) break;;
  esac
done
cc_basename=`$ECHO "$cc_temp" | $SED "s%.*/%%; s%^$host_alias-%%"`

lt_cv_prog_compiler_c_o_RC=yes

if test -n "$compiler"; then
  :

fi

GCC=$lt_save_GCC
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS confptest.$ac_ext >&5'
ac_link='$CC -o confptest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
confptest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

CC=$lt_save_CC

```

```
CFLAGS=$lt_save_CFLAGS
```

```
# Set some internal variables depending on the platform for later use.
```

```
dbus_win=no
```

```
dbus_cygwin=no
```

```
dbus_unix=no
```

```
case "${host}" in
```

```
  *-mingw32ce*)
```

```
    dbus_win=yes
```

```
    dbus_wince=yes
```

```
    ;;
```

```
  *-mingw32*)
```

```
    dbus_win=yes
```

```
    ;;
```

```
  *-cygwin*)
```

```
    dbus_cygwin=yes
```

```
    dbus_unix=yes
```

```
    ;;
```

```
  *)
```

```
    dbus_unix=yes
```

```
    ;;
```

```
esac
```

```
# Special defines for certain platforms
```

```
if test "$dbus_win" = yes; then
```

```
$as_echo "#define DBUS_WIN 1" >>confdefs.h
```

```
    BUILD_TIMESTAMP=`date --iso-8601=minutes`
```

```
    # Assume DBUS_VERSION is always three numbers
```

```
    BUILD_FILEVERSION=`echo "$DBUS_VERSION" | sed -e 's/\./,/g'`,0
```

```
    if test -n "$ac_tool_prefix"; then
```

```
    # Extract the first word of "${ac_tool_prefix}windres", so it can be  
a program name with args.
```

```
set dummy ${ac_tool_prefix}windres; ac_word=$2
```

```
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
```

```
$as_echo_n "checking for $ac_word... " >&6; }
```

```
if ${ac_cv_prog_WINDRES+:} false; then :
```

```
  $as_echo_n "(cached) " >&6
```

```
else
```

```
  if test -n "$WINDRES"; then
```

```
    ac_cv_prog_WINDRES="$WINDRES" # Let the user override the test.
```

```
else
```

```
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
```

```
for as_dir in $PATH
```

```
do
```

```
  IFS=$as_save_IFS
```

```
  test -z "$as_dir" && as_dir=.
```

```
  for ac_exec_ext in '' $ac_executable_extensions; do
```

```

    if as_fn_executable_p "$sas_dir/$ac_word$sac_exec_ext"; then
        ac_cv_prog_WINDRES="{ac_tool_prefix}windres"
        $sas_echo "$sas_me:${as_lineno-$LINENO}: found
$sas_dir/$ac_word$sac_exec_ext" >&5
        break 2
    fi
done
done
IFS=$sas_save_IFS

fi
fi
WINDRES=$ac_cv_prog_WINDRES
if test -n "$WINDRES"; then
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $WINDRES" >&5
    $sas_echo "$WINDRES" >&6; }
else
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
    $sas_echo "no" >&6; }
fi

fi
if test -z "$ac_cv_prog_WINDRES"; then
    ac_ct_WINDRES=$WINDRES
    # Extract the first word of "windres", so it can be a program name
    with args.
    set dummy windres; ac_word=$2
    { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
    $sas_echo_n "checking for $ac_word... " >&6; }
    if ${ac_cv_prog_ac_ct_WINDRES+:} false; then :
        $sas_echo_n "(cached) " >&6
    else
        if test -n "$ac_ct_WINDRES"; then
            ac_cv_prog_ac_ct_WINDRES="$ac_ct_WINDRES" # Let the user override
            the test.
        else
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$sas_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in ' ' $ac_executable_extensions; do
                    if as_fn_executable_p "$sas_dir/$ac_word$sac_exec_ext"; then
                        ac_cv_prog_ac_ct_WINDRES="windres"
                        $sas_echo "$sas_me:${as_lineno-$LINENO}: found
                        $sas_dir/$ac_word$sac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$sas_save_IFS

```

```

fi
fi
ac_ct_WINDRES=$ac_cv_prog_ac_ct_WINDRES
if test -n "$ac_ct_WINDRES"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_ct_WINDRES" >&5
$as_echo "$ac_ct_WINDRES" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  if test "x$ac_ct_WINDRES" = x; then
    WINDRES="no"
  else
    case $cross_compiling:$ac_tool_warned in
yes:)
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
not prefixed with host triplet" >&5
$as_echo "$as_me: WARNING: using cross tools not prefixed with host
triplet" >&2;}
ac_tool_warned=yes ;;
esac
    WINDRES=$ac_ct_WINDRES
  fi
else
  WINDRES="$ac_cv_prog_WINDRES"
fi

  if test "$WINDRES" = no; then
    as_fn_error $? "*** Could not find an implementation of windres
in your PATH." "$LINENO" 5
  fi
  if test "$dbus_wince" = yes; then

$as_echo "#define DBUS_WINCE 1" >>confdefs.h

$as_echo "#define _WIN32_WCE 0x0502" >>confdefs.h

  fi
else

$as_echo "#define DBUS_UNIX 1" >>confdefs.h

fi
if test "$dbus_cygwin" = yes; then

$as_echo "#define DBUS_CYGWIN 1" >>confdefs.h

fi

```

```

if test "$dbus_win" = yes; then
    DBUS_WIN_TRUE=
    DBUS_WIN_FALSE='#'
else
    DBUS_WIN_TRUE='#'
    DBUS_WIN_FALSE=
fi

if test "$dbus_wince" = yes; then
    DBUS_WINCE_TRUE=
    DBUS_WINCE_FALSE='#'
else
    DBUS_WINCE_TRUE='#'
    DBUS_WINCE_FALSE=
fi

if test "$dbus_unix" = yes; then
    DBUS_UNIX_TRUE=
    DBUS_UNIX_FALSE='#'
else
    DBUS_UNIX_TRUE='#'
    DBUS_UNIX_FALSE=
fi

if test "$dbus_cygwin" = yes; then
    DBUS_CYGWIN_TRUE=
    DBUS_CYGWIN_FALSE='#'
else
    DBUS_CYGWIN_TRUE='#'
    DBUS_CYGWIN_FALSE=
fi

# this must come first: other options use this to set their defaults
# Check whether --enable-developer was given.
if test "${enable_developer+set}" = set; then :
    enableval=$enable_developer;
else
    enable_developer=no
fi

DBUS_STATIC_BUILD_CPPFLAGS=
if test "x$enable_shared" = xno; then
    # On Windows, linking against the static library requires special
    effort
    # to turn off DLL import/export processing. We normally link some
    things
    # against the dynamic library, but if we're not building that,
    we'll
    # have to link everything statically.
    DBUS_STATIC_BUILD_CPPFLAGS=-DDBUS_STATIC_BUILD

```

```
fi

# Check whether --enable-ansi was given.
if test "${enable_ansi+set}" = set; then :
    enableval=$enable_ansi; enable_ansi=$enableval
else
    enable_ansi=no
fi

# Check whether --enable-verbose-mode was given.
if test "${enable_verbose_mode+set}" = set; then :
    enableval=$enable_verbose_mode; enable_verbose_mode=$enableval
else
    enable_verbose_mode=$enable_developer
fi

# Check whether --enable-asserts was given.
if test "${enable_asserts+set}" = set; then :
    enableval=$enable_asserts; enable_asserts=$enableval
else
    enable_asserts=$enable_developer
fi

# Check whether --enable-checks was given.
if test "${enable_checks+set}" = set; then :
    enableval=$enable_checks; enable_checks=$enableval
else
    enable_checks=yes
fi

# Check whether --enable-xml-docs was given.
if test "${enable_xml_docs+set}" = set; then :
    enableval=$enable_xml_docs; enable_xml_docs=$enableval
else
    enable_xml_docs=auto
fi

# Check whether --enable-doxygen-docs was given.
if test "${enable_doxygen_docs+set}" = set; then :
    enableval=$enable_doxygen_docs; enable_doxygen_docs=$enableval
else
    enable_doxygen_docs=auto
fi

# Check whether --enable-abstract-sockets was given.
if test "${enable_abstract_sockets+set}" = set; then :
    enableval=$enable_abstract_sockets;
enable_abstract_sockets=$enableval
else
    enable_abstract_sockets=auto
fi
```

```
# Check whether --enable-selinux was given.
if test "${enable_selinux+set}" = set; then :
    enableval=$enable_selinux; enable_selinux=$enableval
else
    enable_selinux=auto
fi

# Check whether --enable-libaudit was given.
if test "${enable_libaudit+set}" = set; then :
    enableval=$enable_libaudit; enable_libaudit=$enableval
else
    enable_libaudit=auto
fi

# Check whether --enable-dnotify was given.
if test "${enable_dnotify+set}" = set; then :
    enableval=$enable_dnotify; enable_dnotify=$enableval
else
    enable_dnotify=auto
fi

# Check whether --enable-inotify was given.
if test "${enable_inotify+set}" = set; then :
    enableval=$enable_inotify; enable_inotify=$enableval
else
    enable_inotify=auto
fi

# Check whether --enable-kqueue was given.
if test "${enable_kqueue+set}" = set; then :
    enableval=$enable_kqueue; enable_kqueue=$enableval
else
    enable_kqueue=auto
fi

# Check whether --enable-console-owner-file was given.
if test "${enable_console_owner_file+set}" = set; then :
    enableval=$enable_console_owner_file;
enable_console_owner_file=$enableval
else
    enable_console_owner_file=auto
fi

# Check whether --enable-userdb-cache was given.
if test "${enable_userdb_cache+set}" = set; then :
    enableval=$enable_userdb_cache; enable_userdb_cache=$enableval
else
    enable_userdb_cache=yes
fi

# Check whether --enable-launchd was given.
```



```
if test "${enable_launchd+set}" = set; then :
    enableval=$enable_launchd; enable_launchd=$enableval
else
    enable_launchd=auto
fi
```

```
# Check whether --enable-systemd was given.
if test "${enable_systemd+set}" = set; then :
    enableval=$enable_systemd; enable_systemd=$enableval
else
    enable_systemd=auto
fi
```

```
# Check whether --with-xml was given.
if test "${with_xml+set}" = set; then :
    withval=$with_xml;
fi
```

```
# Check whether --with-init-scripts was given.
if test "${with_init_scripts+set}" = set; then :
    withval=$with_init_scripts;
fi
```

```
# Check whether --with-session-socket-dir was given.
if test "${with_session_socket_dir+set}" = set; then :
    withval=$with_session_socket_dir;
fi
```

```
# Check whether --with-test-socket-dir was given.
if test "${with_test_socket_dir+set}" = set; then :
    withval=$with_test_socket_dir;
fi
```

```
# Check whether --with-system-pid-file was given.
if test "${with_system_pid_file+set}" = set; then :
    withval=$with_system_pid_file;
fi
```

```
# Check whether --with-system-socket was given.
if test "${with_system_socket+set}" = set; then :
    withval=$with_system_socket;
fi
```

```
# Check whether --with-console-auth-dir was given.
```

```
if test "${with_console_auth_dir+set}" = set; then :
    withval=$with_console_auth_dir;
fi

# Check whether --with-console-owner-file was given.
if test "${with_console_owner_file+set}" = set; then :
    withval=$with_console_owner_file;
fi

# Check whether --with-launchd-agent-dir was given.
if test "${with_launchd_agent_dir+set}" = set; then :
    withval=$with_launchd_agent_dir;
fi

# Check whether --with-dbus_user was given.
if test "${with_dbus_user+set}" = set; then :
    withval=$with_dbus_user;
fi

# Check whether --with-dbus_daemon_dir was given.
if test "${with_dbus_daemon_dir+set}" = set; then :
    withval=$with_dbus_daemon_dir;
fi

# Check whether --with-dbus_session_bus_default_address was given.
if test "${with_dbus_session_bus_default_address+set}" = set; then :
    withval=$with_dbus_session_bus_default_address;
with_dbus_session_bus_default_address=$withval
else
    with_dbus_session_bus_default_address=nonce-tcp:
fi

# Check whether --enable-embedded-tests was given.
if test "${enable_embedded_tests+set}" = set; then :
    enableval=$enable_embedded_tests;
else
    enable_embedded_tests=$enable_developer
fi

# Check whether --enable-modular-tests was given.
if test "${enable_modular_tests+set}" = set; then :
    enableval=$enable_modular_tests;
else
    enable_modular_tests=auto
fi
```

```

# --enable-tests overrides both --enable-embedded-tests and
# --enable-modular-tests
# Check whether --enable-tests was given.
if test "${enable_tests+set}" = set; then :
  enableval=$enable_tests;
  if test "x$enableval" = xyes; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage was
requested with --enable-tests=yes" >&5
$as_echo "$as_me: Full test coverage was requested with --enable-
tests=yes" >&6;}
    { $as_echo "$as_me:${as_lineno-$LINENO}: This has many
dependencies (GLib, dbus-glib, Python)" >&5
$as_echo "$as_me: This has many dependencies (GLib, dbus-glib,
Python)" >&6;}
    fi
  enable_embedded_tests=$enableval
  enable_modular_tests=$enableval

fi

# DBUS_ENABLE_EMBEDDED_TESTS controls unit tests built in to .c files
# and also some stuff in the test/ subdir. DBUS_BUILD_TESTS was an
older
# name for this.
if test "x$enable_embedded_tests" = xyes; then
  DBUS_BUILD_TESTS_TRUE=
  DBUS_BUILD_TESTS_FALSE='#'
else
  DBUS_BUILD_TESTS_TRUE='#'
  DBUS_BUILD_TESTS_FALSE=
fi

if test "x$enable_embedded_tests" = xyes; then
  DBUS_ENABLE_EMBEDDED_TESTS_TRUE=
  DBUS_ENABLE_EMBEDDED_TESTS_FALSE='#'
else
  DBUS_ENABLE_EMBEDDED_TESTS_TRUE='#'
  DBUS_ENABLE_EMBEDDED_TESTS_FALSE=
fi

if test "x$enable_embedded_tests" = xyes; then

$as_echo "#define DBUS_ENABLE_EMBEDDED_TESTS 1" >>confdefs.h

$as_echo "#define DBUS_BUILD_TESTS 1" >>confdefs.h

fi

# DBUS_ENABLE_MODULAR_TESTS controls tests that work based on public
API.

```

```

# These use GTest, from GLib, because life's too short. They're
enabled by
# default (unless you don't have GLib), because they don't bloat the
library
# or binaries.

with_glib=yes

if test "x$enable_modular_tests" != xno; then

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for GLIB" >&5
$as_echo_n "checking for GLIB... " >&6; }

if test -n "$GLIB_CFLAGS"; then
    pkg_cv_GLIB_CFLAGS="$GLIB_CFLAGS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"glib-2.0 >= 2.24, gio-2.0 >= 2.24\""; } >&5
        ($PKG_CONFIG --exists --print-errors "glib-2.0 >= 2.24, gio-2.0 >=
2.24") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_GLIB_CFLAGS=`$PKG_CONFIG --cflags "glib-2.0 >= 2.24, gio-2.0
>= 2.24" 2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi
if test -n "$GLIB_LIBS"; then
    pkg_cv_GLIB_LIBS="$GLIB_LIBS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"glib-2.0 >= 2.24, gio-2.0 >= 2.24\""; } >&5
        ($PKG_CONFIG --exists --print-errors "glib-2.0 >= 2.24, gio-2.0 >=
2.24") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_GLIB_LIBS=`$PKG_CONFIG --libs "glib-2.0 >= 2.24, gio-2.0 >=
2.24" 2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi

```

```

if test $pkg_failed = yes; then

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        GLIB_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "glib-2.0 >= 2.24, gio-2.0 >= 2.24" 2>&1`
    else
        GLIB_PKG_ERRORS=`$PKG_CONFIG --print-errors "glib-2.0 >=
2.24, gio-2.0 >= 2.24" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$GLIB_PKG_ERRORS" >&5

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        if test "x$enable_modular_tests" = xyes; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage (--
enable-modular-tests=yes or --enable-tests=yes) requires GLib" >&5
$as_echo "$as_me: Full test coverage (--enable-modular-tests=yes or --
enable-tests=yes) requires GLib" >&6;}
            as_fn_error $? "$GLIB_ERRORS" "$LINENO" 5
        else # assumed to be "auto"
            with_glib=no
        fi
elif test $pkg_failed = untried; then
    if test "x$enable_modular_tests" = xyes; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage (--
enable-modular-tests=yes or --enable-tests=yes) requires GLib" >&5
$as_echo "$as_me: Full test coverage (--enable-modular-tests=yes or --
enable-tests=yes) requires GLib" >&6;}
        as_fn_error $? "$GLIB_ERRORS" "$LINENO" 5
    else # assumed to be "auto"
        with_glib=no
    fi
else
    GLIB_CFLAGS=$pkg_cv_GLIB_CFLAGS
    GLIB_LIBS=$pkg_cv_GLIB_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    :
fi
    # If dbus-gmain.[ch] returned to libdbus then we wouldn't need this

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for DBUS_GLIB" >&5

```

```

$as_echo_n "checking for DBUS_GLIB... " >&6; }

if test -n "$DBUS_GLIB_CFLAGS"; then
    pkg_cv_DBUS_GLIB_CFLAGS="$DBUS_GLIB_CFLAGS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"dbus-glib-1\""; } >&5
        ($PKG_CONFIG --exists --print-errors "dbus-glib-1") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_DBUS_GLIB_CFLAGS=`$PKG_CONFIG --cflags "dbus-glib-1"
2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi

if test -n "$DBUS_GLIB_LIBS"; then
    pkg_cv_DBUS_GLIB_LIBS="$DBUS_GLIB_LIBS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"dbus-glib-1\""; } >&5
        ($PKG_CONFIG --exists --print-errors "dbus-glib-1") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_DBUS_GLIB_LIBS=`$PKG_CONFIG --libs "dbus-glib-1" 2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi

if test $pkg_failed = yes; then

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        DBUS_GLIB_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "dbus-glib-1" 2>&1`
    else

```

```

        DBUS_GLIB_PKG_ERRORS=`$PKG_CONFIG --print-errors "dbus-
glib-1" 2>&1`
        fi
        # Put the nasty error message in config.log where it belongs
        echo "$DBUS_GLIB_PKG_ERRORS" >&5

        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        if test "x$enable_modular_tests" = xyes; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage (--
enable-modular-tests=yes or --enable-tests=yes) requires dbus-glib"
>&5
$as_echo "$as_me: Full test coverage (--enable-modular-tests=yes or --
enable-tests=yes) requires dbus-glib" >&6;}
            as_fn_error $? "$DBUS_GLIB_ERRORS" "$LINENO" 5
        else # assumed to be "auto"
            with_glib=no
        fi
    elif test $pkg_failed = untried; then
        if test "x$enable_modular_tests" = xyes; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage (--
enable-modular-tests=yes or --enable-tests=yes) requires dbus-glib"
>&5
$as_echo "$as_me: Full test coverage (--enable-modular-tests=yes or --
enable-tests=yes) requires dbus-glib" >&6;}
            as_fn_error $? "$DBUS_GLIB_ERRORS" "$LINENO" 5
        else # assumed to be "auto"
            with_glib=no
        fi
    else
        DBUS_GLIB_CFLAGS=$pkg_cv_DBUS_GLIB_CFLAGS
        DBUS_GLIB_LIBS=$pkg_cv_DBUS_GLIB_LIBS
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
        :
    fi
fi
fi
if test "x$enable_modular_tests" != xno; then

$as_echo "#define DBUS_ENABLE_MODULAR_TESTS 1" >>confdefs.h

fi
if test "x$enable_modular_tests" != xno; then
    DBUS_ENABLE_MODULAR_TESTS_TRUE=
    DBUS_ENABLE_MODULAR_TESTS_FALSE='#'
else
    DBUS_ENABLE_MODULAR_TESTS_TRUE='#'
    DBUS_ENABLE_MODULAR_TESTS_FALSE=
fi

if test "x$with_glib" != xno; then

```

```

$as_echo "#define DBUS_WITH_GLIB 1" >>confdefs.h

fi
if test "x$with_glib" != xno; then
    DBUS_WITH_GLIB_TRUE=
    DBUS_WITH_GLIB_FALSE='#'
else
    DBUS_WITH_GLIB_TRUE='#'
    DBUS_WITH_GLIB_FALSE=
fi

# Check whether --enable-installed-tests was given.
if test "${enable_installed_tests+set}" = set; then :
    enableval=$enable_installed_tests;
else
    enable_installed_tests=no
fi

if test "x$enable_installed_tests" = xyes; then
    DBUS_ENABLE_INSTALLED_TESTS_TRUE=
    DBUS_ENABLE_INSTALLED_TESTS_FALSE='#'
else
    DBUS_ENABLE_INSTALLED_TESTS_TRUE='#'
    DBUS_ENABLE_INSTALLED_TESTS_FALSE=
fi

if test "x$enable_tests" = xyes; then
    # full test coverage is required, Python is a hard dependency
    { $as_echo "$as_me:${as_lineno-$LINENO}: Full test coverage (--
enable-tests=yes) requires Python, dbus-python, pygobject" >&5
$as_echo "$as_me: Full test coverage (--enable-tests=yes) requires
Python, dbus-python, pygobject" >&6;}

        if test -n "$PYTHON"; then
            # If the user set $PYTHON, use it and don't search something
else.
            { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether
$PYTHON version is >= 2.6" >&5
$as_echo_n "checking whether $PYTHON version is >= 2.6... " >&6; }
            prog="import sys
# split strings by '.' and convert to numeric. Append some zeros
# because we need at least 4 digits for the hex conversion.
# map returns an iterator in Python 3.0 and a list in 2.x
minver = list(map(int, '2.6'.split('.'))) + [0, 0, 0]

```



```

minverhex = 0
# xrange is not present in Python 3.0 and range returns an iterator
for i in list(range(0, 4)): minverhex = (minverhex << 8) + minver[i]
sys.exit(sys.hexversion < minverhex)"
  if { echo "$as_me:$LINENO: $PYTHON -c "$prog"" >&5
    ($PYTHON -c "$prog") >&5 2>&5
    ac_status=$?
    echo "$as_me:$LINENO: \ $? = $ac_status" >&5
    (exit $ac_status); }; then :
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
          as_fn_error $? "Python interpreter is too old"
"$LINENO" 5
fi
    am_display_PYTHON=$PYTHON
  else
    # Otherwise, try each interpreter until we find one that
satisfies
    # VERSION.
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for a Python
interpreter with version >= 2.6" >&5
$as_echo_n "checking for a Python interpreter with version >= 2.6... "
>&6; }
if ${am_cv_pathless_PYTHON+:} false; then :
  $as_echo_n "(cached) " >&6
else
    for am_cv_pathless_PYTHON in python python2 python3 python3.3
python3.2 python3.1 python3.0 python2.7 python2.6 python2.5 python2.4
python2.3 python2.2 python2.1 python2.0 none; do
    test "$am_cv_pathless_PYTHON" = none && break
    prog="import sys
# split strings by '.' and convert to numeric. Append some zeros
# because we need at least 4 digits for the hex conversion.
# map returns an iterator in Python 3.0 and a list in 2.x
minver = list(map(int, '2.6'.split('.'))) + [0, 0, 0]
minverhex = 0
# xrange is not present in Python 3.0 and range returns an iterator
for i in list(range(0, 4)): minverhex = (minverhex << 8) + minver[i]
sys.exit(sys.hexversion < minverhex)"
    if { echo "$as_me:$LINENO: $am_cv_pathless_PYTHON -c "$prog"" >&5
      ($am_cv_pathless_PYTHON -c "$prog") >&5 2>&5
      ac_status=$?
      echo "$as_me:$LINENO: \ $? = $ac_status" >&5
      (exit $ac_status); }; then :
    break
fi
    done
fi

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_pathless_PYTHON" >&5
$as_echo "$am_cv_pathless_PYTHON" >&6; }
    # Set $PYTHON to the absolute path of $am_cv_pathless_PYTHON.
    if test "$am_cv_pathless_PYTHON" = none; then
        PYTHON=:
    else
        # Extract the first word of "$am_cv_pathless_PYTHON", so it
        can be a program name with args.
        set dummy $am_cv_pathless_PYTHON; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
        $as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_path_PYTHON+:} false; then :
            $as_echo_n "(cached) " >&6
        else
            case $PYTHON in
            [[\/* | ?:[\/*]*)
            ac_cv_path_PYTHON="$PYTHON" # Let the user override the test with a
            path.
            ;;
            *)
            as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
            for as_dir in $PATH
            do
                IFS=$as_save_IFS
                test -z "$as_dir" && as_dir=.
                for ac_exec_ext in '$ac_executable_extensions; do
                    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                        ac_cv_path_PYTHON="$as_dir/$ac_word$ac_exec_ext"
                        $as_echo "$as_me:${as_lineno-$LINENO}: found
                        $as_dir/$ac_word$ac_exec_ext" >&5
                        break 2
                    fi
                done
            done
            IFS=$as_save_IFS

            ;;
            esac
        fi
        PYTHON=$ac_cv_path_PYTHON
        if test -n "$PYTHON"; then
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: $PYTHON" >&5
            $as_echo "$PYTHON" >&6; }
        else
            { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
            $as_echo "no" >&6; }
        fi

        fi

        am_display_PYTHON=$am_cv_pathless_PYTHON

```

```

fi

if test "$PYTHON" = ;; then
    as_fn_error $? "no suitable Python interpreter found" "$LINENO"
5
else

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON version" >&5
$as_echo_n "checking for $am_display_PYTHON version... " >&6; }
if ${am_cv_python_version+:} false; then :
    $as_echo_n "(cached) " >&6
else
    am_cv_python_version=`$PYTHON -c "import sys;
sys.stdout.write(sys.version[:3])"`
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_version" >&5
$as_echo "$am_cv_python_version" >&6; }
PYTHON_VERSION=$am_cv_python_version

PYTHON_PREFIX='${prefix}'

PYTHON_LIB_PREFIX='${libdir}'

PYTHON_EXEC_PREFIX='${exec_prefix}'

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON platform" >&5
$as_echo_n "checking for $am_display_PYTHON platform... " >&6; }
if ${am_cv_python_platform+:} false; then :
    $as_echo_n "(cached) " >&6
else
    am_cv_python_platform=`$PYTHON -c "import sys;
sys.stdout.write(sys.platform)"`
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_platform" >&5
$as_echo "$am_cv_python_platform" >&6; }
PYTHON_PLATFORM=$am_cv_python_platform

# Just factor out some code duplication.
am_python_setup_sysconfig="\
import sys
# Prefer sysconfig over distutils.sysconfig, for better compatibility

```

```

# with python 3.x. See automake bug#10227.
try:
    import sysconfig
except ImportError:
    can_use_sysconfig = 0
else:
    can_use_sysconfig = 1
# Can't use sysconfig in CPython 2.7, since it's broken in
virtualenvs:
# <https://github.com/pypa/virtualenv/issues/118>
try:
    from platform import python_implementation
    if python_implementation() == 'CPython' and sys.version[:3] ==
'2.7':
        can_use_sysconfig = 0
except ImportError:
    pass"

        { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON script directory" >&5
$as_echo_n "checking for $am_display_PYTHON script directory... " >&6;
}
if ${am_cv_python_pythondir+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test "x$prefix" = xNONE
    then
        am_py_prefix=$ac_default_prefix
    else
        am_py_prefix=$prefix
    fi
    am_cv_python_pythondir=`$PYTHON -c "
$am_python_setup_sysconfig
if can_use_sysconfig:
    sitedir = sysconfig.get_path('purelib',
vars={'base': '$am_py_prefix'})
else:
    from distutils import sysconfig
    sitedir = sysconfig.get_python_lib(0, 0, prefix='$am_py_prefix')
sys.stdout.write(sitedir)" ||
echo "$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-packages" `
    case $am_cv_python_pythondir in
    $am_py_prefix*)
        am__strip_prefix=`echo "$am_py_prefix" | sed 's|.|.|g'`
        am_cv_python_pythondir=`echo "$am_cv_python_pythondir" | sed
"s,^$am__strip_prefix,$PYTHON_PREFIX," `
        ;;
    *)
        case $am_py_prefix in
        /usr|/System*) ;;
        *)

```

```

am_cv_python_pythondir=$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-
packages
    ;;
    esac
    ;;
    esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_pythondir" >&5
$as_echo "$am_cv_python_pythondir" >&6; }
    pythondir=$am_cv_python_pythondir

pkgpythondir=\${pythondir}/$PACKAGE

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON extension module directory" >&5
$as_echo_n "checking for $am_display_PYTHON extension module
directory... " >&6; }
if ${am_cv_python_pyexecdir+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test "x$exec_prefix" = xNONE
    then
        am_py_exec_prefix=$am_py_prefix
    else
        am_py_exec_prefix=$exec_prefix
    fi
    am_cv_python_pyexecdir=`$PYTHON -c "
$am_python_setup_sysconfig
if can_use_sysconfig:
    sitedir = sysconfig.get_path('platlib',
vars={'platbase': '$am_py_prefix'})
else:
    from distutils import sysconfig
    sitedir = sysconfig.get_python_lib(1, 0, prefix='$am_py_prefix')
sys.stdout.write(sitedir)" ||
echo "$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-packages" `
    case $am_cv_python_pyexecdir in
    $am_py_exec_prefix*)
        am__strip_prefix=`echo "$am_py_exec_prefix" | sed 's|.|.|g'`
        am_cv_python_pyexecdir=`echo "$am_cv_python_pyexecdir" | sed
"s,^$am__strip_prefix,$PYTHON_EXEC_PREFIX," `
        ;;
    *)
        case $am_py_exec_prefix in
        /usr|/System*) ;;
        *)

```

```

am_cv_python_pyexecdir=$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-
packages
    ;;
    esac
    ;;
    esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_pyexecdir" >&5
$as_echo "$am_cv_python_pyexecdir" >&6; }
pyexecdir=$am_cv_python_pyexecdir

pkgpyexecdir=\${pyexecdir}/$PACKAGE

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for Python modules
for full test coverage" >&5
$as_echo_n "checking for Python modules for full test coverage... "
>&6; }
if "$PYTHON" -c "import dbus, gobject, dbus.mainloop.glib"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
    as_fn_error $? "cannot import dbus, gobject, dbus.mainloop.glib
Python modules" "$LINENO" 5
fi
else
# --enable-tests not given: do not abort if Python is missing

    if test -n "$PYTHON"; then
# If the user set $PYTHON, use it and don't search something
else.
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether
$PYTHON version is >= 2.6" >&5
$as_echo_n "checking whether $PYTHON version is >= 2.6... " >&6; }
    prog="import sys
# split strings by '.' and convert to numeric. Append some zeros

```

```

# because we need at least 4 digits for the hex conversion.
# map returns an iterator in Python 3.0 and a list in 2.x
minver = list(map(int, '2.6'.split('.'))) + [0, 0, 0]
minverhex = 0
# xrange is not present in Python 3.0 and range returns an iterator
for i in list(range(0, 4)): minverhex = (minverhex << 8) + minver[i]
sys.exit(sys.hexversion < minverhex)"
  if { echo "$as_me:$LINENO: $PYTHON -c "$prog"" >&5
    ($PYTHON -c "$prog") >&5 2>&5
    ac_status=$?
    echo "$as_me:$LINENO: \ $? = $ac_status" >&5
    (exit $ac_status); }; then :
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
          as_fn_error $? "Python interpreter is too old"
"$LINENO" 5
fi
    am_display_PYTHON=$PYTHON
  else
    # Otherwise, try each interpreter until we find one that
satisfies
    # VERSION.
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for a Python
interpreter with version >= 2.6" >&5
$as_echo_n "checking for a Python interpreter with version >= 2.6... "
>&6; }
if ${am_cv_pathless_PYTHON+:} false; then :
  $as_echo_n "(cached) " >&6
else
    for am_cv_pathless_PYTHON in python python2 python3 python3.3
python3.2 python3.1 python3.0 python2.7 python2.6 python2.5 python2.4
python2.3 python2.2 python2.1 python2.0 none; do
    test "$am_cv_pathless_PYTHON" = none && break
    prog="import sys
# split strings by '.' and convert to numeric. Append some zeros
# because we need at least 4 digits for the hex conversion.
# map returns an iterator in Python 3.0 and a list in 2.x
minver = list(map(int, '2.6'.split('.'))) + [0, 0, 0]
minverhex = 0
# xrange is not present in Python 3.0 and range returns an iterator
for i in list(range(0, 4)): minverhex = (minverhex << 8) + minver[i]
sys.exit(sys.hexversion < minverhex)"
    if { echo "$as_me:$LINENO: $am_cv_pathless_PYTHON -c "$prog"" >&5
      ($am_cv_pathless_PYTHON -c "$prog") >&5 2>&5
      ac_status=$?
      echo "$as_me:$LINENO: \ $? = $ac_status" >&5
      (exit $ac_status); }; then :
    break

```

```

fi
    done
fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$am_cv_pathless_PYTHON" >&5
$sas_echo "$am_cv_pathless_PYTHON" >&6; }
    # Set $PYTHON to the absolute path of $am_cv_pathless_PYTHON.
    if test "$am_cv_pathless_PYTHON" = none; then
        PYTHON=:
    else
        # Extract the first word of "$am_cv_pathless_PYTHON", so it
        can be a program name with args.
        set dummy $am_cv_pathless_PYTHON; ac_word=$2
        { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
        $sas_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_path_PYTHON+:} false; then :
            $sas_echo_n "(cached) " >&6
        else
            case $PYTHON in
            [\\/]*)
                ac_cv_path_PYTHON="$PYTHON" # Let the user override the test with a
                path.
                ;;
            *)
                as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
                for as_dir in $PATH
                do
                    IFS=$as_save_IFS
                    test -z "$as_dir" && as_dir=.
                    for ac_exec_ext in ' $ac_executable_extensions; do
                        if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                            ac_cv_path_PYTHON="$as_dir/$ac_word$ac_exec_ext"
                            $sas_echo "$sas_me:${as_lineno-$LINENO}: found
                            $as_dir/$ac_word$ac_exec_ext" >&5
                            break 2
                        fi
                    done
                done
                IFS=$as_save_IFS

                ;;
            esac
        fi
        PYTHON=$ac_cv_path_PYTHON
        if test -n "$PYTHON"; then
            { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $PYTHON" >&5
            $sas_echo "$PYTHON" >&6; }
        else
            { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
            $sas_echo "no" >&6; }
        fi
    fi

```



```

        fi
        am_display_PYTHON=$am_cv_pathless_PYTHON
    fi

if test "$PYTHON" = :; then
    :
else

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON version" >&5
$as_echo_n "checking for $am_display_PYTHON version... " >&6; }
if ${am_cv_python_version+:} false; then :
    $as_echo_n "(cached) " >&6
else
    am_cv_python_version=`$PYTHON -c "import sys;
sys.stdout.write(sys.version[:3])"`
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_version" >&5
$as_echo "$am_cv_python_version" >&6; }
PYTHON_VERSION=$am_cv_python_version

PYTHON_PREFIX='${prefix}'

PYTHON_LIB_PREFIX='${libdir}'

PYTHON_EXEC_PREFIX='${exec_prefix}'

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON platform" >&5
$as_echo_n "checking for $am_display_PYTHON platform... " >&6; }
if ${am_cv_python_platform+:} false; then :
    $as_echo_n "(cached) " >&6
else
    am_cv_python_platform=`$PYTHON -c "import sys;
sys.stdout.write(sys.platform)"`
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_platform" >&5
$as_echo "$am_cv_python_platform" >&6; }
PYTHON_PLATFORM=$am_cv_python_platform

# Just factor out some code duplication.
am_python_setup_sysconfig="\

```

```

import sys
# Prefer sysconfig over distutils.sysconfig, for better compatibility
# with python 3.x. See automake bug#10227.
try:
    import sysconfig
except ImportError:
    can_use_sysconfig = 0
else:
    can_use_sysconfig = 1
# Can't use sysconfig in CPython 2.7, since it's broken in
virtualenvs:
# <https://github.com/pypa/virtualenv/issues/118>
try:
    from platform import python_implementation
    if python_implementation() == 'CPython' and sys.version[:3] ==
'2.7':
        can_use_sysconfig = 0
except ImportError:
    pass"

        { $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON script directory" >&5
$sas_echo_n "checking for $am_display_PYTHON script directory... " >&6;
}
if ${am_cv_python_pythondir+:} false; then :
    $sas_echo_n "(cached) " >&6
else
    if test "x$prefix" = xNONE
    then
        am_py_prefix=$ac_default_prefix
    else
        am_py_prefix=$prefix
    fi
    am_cv_python_pythondir=`$PYTHON -c "
$am_python_setup_sysconfig
if can_use_sysconfig:
    sitedir = sysconfig.get_path('purelib',
vars={'base': '$am_py_prefix'})
else:
    from distutils import sysconfig
    sitedir = sysconfig.get_python_lib(0, 0, prefix='$am_py_prefix')
sys.stdout.write(sitedir) " ||
    echo "$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-packages" `
    case $am_cv_python_pythondir in
    $am_py_prefix*)
        am__strip_prefix=`echo "$am_py_prefix" | sed 's|.|.|g'`
        am_cv_python_pythondir=`echo "$am_cv_python_pythondir" | sed
"s,^$am__strip_prefix,$PYTHON_PREFIX," `
        ;;
    *)
        case $am_py_prefix in

```

```

        /usr|/System*) ;;
    *)

am_cv_python_pythondir=$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-
packages
    ;;
    esac
    ;;
    esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_pythondir" >&5
$as_echo "$am_cv_python_pythondir" >&6; }
pythondir=$am_cv_python_pythondir

pkgpythondir=\${pythondir}/$PACKAGE

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
$am_display_PYTHON extension module directory" >&5
$as_echo_n "checking for $am_display_PYTHON extension module
directory... " >&6; }
if ${am_cv_python_pyexecdir+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "x$exec_prefix" = xNONE
  then
    am_py_exec_prefix=$am_py_prefix
  else
    am_py_exec_prefix=$exec_prefix
  fi
  am_cv_python_pyexecdir=`$PYTHON -c "
$am_python_setup_sysconfig
if can_use_sysconfig:
  sitedir = sysconfig.get_path('platlib',
vars={'platbase': '$am_py_prefix'})
else:
  from distutils import sysconfig
  sitedir = sysconfig.get_python_lib(1, 0, prefix='$am_py_prefix')
sys.stdout.write(sitedir)" ||
echo "$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-packages" `
  case $am_cv_python_pyexecdir in
  $am_py_exec_prefix*)
    am__strip_prefix=`echo "$am_py_exec_prefix" | sed 's|.|.|g'`
    am_cv_python_pyexecdir=`echo "$am_cv_python_pyexecdir" | sed
"s,^$am__strip_prefix,$PYTHON_EXEC_PREFIX,"`
    ;;
  *)
    case $am_py_exec_prefix in

```

```

        /usr|/System*) ;;
        *)

am_cv_python_pyexecdir=$PYTHON_LIB_PREFIX/python$PYTHON_VERSION/site-
packages
        ;;
        esac
        ;;
        esac

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$am_cv_python_pyexecdir" >&5
$as_echo "$am_cv_python_pyexecdir" >&6; }
pyexecdir=$am_cv_python_pyexecdir

pkgpyexecdir=\${pyexecdir}/$PACKAGE

fi

fi

if test x$enable_verbose_mode = xyes; then
$as_echo "#define DBUS_ENABLE_VERBOSE_MODE 1" >>confdefs.h
fi

if test x$enable_asserts = xno; then
$as_echo "#define DBUS_DISABLE_ASSERT 1" >>confdefs.h

    DISABLE_UNUSED_WARNINGS="unused-label"
    R_DYNAMIC_LDFLAG=""
else
    # -rdynamic is needed for glibc's backtrace_symbols to work.
    # No clue how much overhead this adds, but it's useful
    # to do this on any assertion failure,
    # so for now it's enabled anytime asserts are (currently not
    # in production builds).

    # To get -rdynamic you pass -export-dynamic to libtool.

$as_echo "#define DBUS_BUILT_R_DYNAMIC 1" >>confdefs.h

    R_DYNAMIC_LDFLAG=-export-dynamic
fi

```

```

if test x$enable_checks = xno; then

$as_echo "#define DBUS_DISABLE_CHECKS 1" >>confdefs.h

$as_echo "#define G_DISABLE_CHECKS 1" >>confdefs.h

    DISABLE_UNUSED_WARNINGS="unused-label"
fi

if test x$enable_userdb_cache = xyes; then

$as_echo "#define DBUS_ENABLE_USERDB_CACHE 1" >>confdefs.h

fi

if test x$enable_compiler_coverage = xyes; then
    ## so that config.h changes when you toggle gcov support

cat >>confdefs.h <<_ACEOF
#define DBUS_GCOV_ENABLED 1
_ACEOF

fi

# glibc21.m4 serial 3

# Test for the GNU C Library, version 2.1 or newer.
# From Bruno Haible.

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether we are using
the GNU C Library 2.1 or newer" >&5
$as_echo_n "checking whether we are using the GNU C Library 2.1 or
newer... " >&6; }
if ${ac_cv_gnu_library_2_1+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <features.h>
#ifdef __GNU_LIBRARY__
    #if ((__GLIBC__ == 2 && __GLIBC_MINOR__ >= 1) || (__GLIBC__ > 2))
        Lucky GNU user
    #endif
#endif

_ACEOF
if (eval "$ac_cpp conftest.$ac_ext" 2>&5 |
    $EGREP "Lucky GNU user" >/dev/null 2>&1; then :

```

```

    ac_cv_gnu_library_2_1=yes
else
    ac_cv_gnu_library_2_1=no
fi
rm -f conftest*

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_gnu_library_2_1" >&5
$as_echo "$ac_cv_gnu_library_2_1" >&6; }

#### Integer sizes

# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[[(sizeof (unsigned char)) >= 0]];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of char" >&5
$as_echo_n "checking size of char... " >&6; }
if ${ac_cv_sizeof_char+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (char))"
"ac_cv_sizeof_char" "$ac_includes_default"; then :
else
    if test "$ac_cv_type_char" = yes; then
        { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`$ac_pwd':"
>&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (char)
See \`config.log' for more details" "$LINENO" 5; }
        else
            ac_cv_sizeof_char=0
        fi
    fi
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_sizeof_char"
>&5
$as_echo "$ac_cv_sizeof_char" >&6; }

cat >>confdefs.h <<_ACEOF
#define SIZEOF_CHAR $ac_cv_sizeof_char
_ACEOF

# The cast to long int works around a bug in the HP C Compiler

```

```

# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[sizeof (unsigned char) >= 0];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of short" >&5
$as_echo_n "checking size of short... " >&6; }
if ${ac_cv_sizeof_short+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (short))"
"ac_cv_sizeof_short" "$ac_includes_default"; then :

else
  if test "$ac_cv_type_short" = yes; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`${ac_pwd':"
>&5
$as_echo "$as_me: error: in \`${ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (short)
See `config.log' for more details" "$LINENO" 5; }
    else
      ac_cv_sizeof_short=0
    fi
  fi

fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_sizeof_short"
>&5
$as_echo "$ac_cv_sizeof_short" >&6; }

```

```

cat >>confdefs.h <<_ACEOF
#define SIZEOF_SHORT $ac_cv_sizeof_short
_ACEOF

```

```

# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[sizeof (unsigned char) >= 0];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of long" >&5
$as_echo_n "checking size of long... " >&6; }
if ${ac_cv_sizeof_long+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (long))"
"ac_cv_sizeof_long" "$ac_includes_default"; then :

else
  if test "$ac_cv_type_long" = yes; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`${ac_pwd':"
>&5
$as_echo "$as_me: error: in \`${ac_pwd':" >&2;}

```

```

as_fn_error 77 "cannot compute sizeof (long)
See `config.log' for more details" "$LINENO" 5; }
    else
        ac_cv_sizeof_long=0
    fi
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_sizeof_long"
>&5
$as_echo "$ac_cv_sizeof_long" >&6; }

cat >>confdefs.h <<_ACEOF
#define SIZEOF_LONG $ac_cv_sizeof_long
_ACEOF

# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[[(sizeof (unsigned char)) >= 0]];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of int" >&5
$as_echo_n "checking size of int... " >&6; }
if ${ac_cv_sizeof_int+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (int))"
"ac_cv_sizeof_int" "$ac_includes_default"; then :
else
    if test "$ac_cv_type_int" = yes; then
        { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `'$ac_pwd':"
>&5
$as_echo "$as_me: error: in `'$ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (int)
See `config.log' for more details" "$LINENO" 5; }
    else
        ac_cv_sizeof_int=0
    fi
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_sizeof_int"
>&5
$as_echo "$ac_cv_sizeof_int" >&6; }

cat >>confdefs.h <<_ACEOF
#define SIZEOF_INT $ac_cv_sizeof_int

```


_ACEOF

```
# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[sizeof (unsigned char) >= 0];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of void *" >&5
$as_echo_n "checking size of void *... " >&6; }
if ${ac_cv_sizeof_void_p+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (void *))"
"ac_cv_sizeof_void_p" "$ac_includes_default"; then :

else
  if test "$ac_cv_type_void_p" = yes; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (void *)"
See `config.log' for more details" "$LINENO" 5; }
    else
      ac_cv_sizeof_void_p=0
    fi
  fi

fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_sizeof_void_p"
>&5
$as_echo "$ac_cv_sizeof_void_p" >&6; }
```

```
cat >>confdefs.h <<_ACEOF
#define SIZEOF_VOID_P $ac_cv_sizeof_void_p
_ACEOF
```

```
# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[sizeof (unsigned char) >= 0];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of long long"
>&5
$as_echo_n "checking size of long long... " >&6; }
if ${ac_cv_sizeof_long_long+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (long long))"
"ac_cv_sizeof_long_long" "$ac_includes_default"; then :
```

```

else
  if test "$ac_cv_type_long_long" = yes; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`$ac_pwd':"
>&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (long long)
See \`config.log' for more details" "$LINENO" 5; }
    else
      ac_cv_sizeof_long_long=0
    fi
  fi

fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_sizeof_long_long" >&5
$as_echo "$ac_cv_sizeof_long_long" >&6; }

cat >>confdefs.h <<_ACEOF
#define SIZEOF_LONG_LONG $ac_cv_sizeof_long_long
_ACEOF

# The cast to long int works around a bug in the HP C Compiler
# version HP92453-01 B.11.11.23709.GP, which incorrectly rejects
# declarations like `int a3[[(sizeof (unsigned char)) >= 0]];'.
# This bug is HP SR number 8606223364.
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking size of __int64" >&5
$as_echo_n "checking size of __int64... " >&6; }
if ${ac_cv_sizeof__int64+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if ac_fn_c_compute_int "$LINENO" "(long int) (sizeof (__int64))"
"ac_cv_sizeof__int64" "$ac_includes_default"; then :

else
  if test "$ac_cv_type__int64" = yes; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in \`$ac_pwd':"
>&5
$as_echo "$as_me: error: in \`$ac_pwd':" >&2;}
as_fn_error 77 "cannot compute sizeof (__int64)
See \`config.log' for more details" "$LINENO" 5; }
    else
      ac_cv_sizeof__int64=0
    fi
  fi

fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_sizeof__int64" >&5
$as_echo "$ac_cv_sizeof__int64" >&6; }

```

```

cat >>confdefs.h <<_ACEOF
#define SIZEOF___INT64 $ac_cv_sizeof___int64
_ACEOF

# Check whether --with-64-bit was given.
if test "${with_64_bit+set}" = set; then :
  withval=$with_64_bit;
else
  with_64_bit=yes
fi

### See what our 64 bit type is called
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking 64-bit integer type"
>&5
$as_echo_n "checking 64-bit integer type... " >&6; }

case 8 in
$ac_cv_sizeof_int)
  dbusint64=int
  dbusint64_constant='(val) '
  dbusuint64_constant='(val) '
  dbusint64_printf_modifier='""'
  ;;
$ac_cv_sizeof_long)
  dbusint64=long
  dbusint64_constant='(val##L) '
  dbusuint64_constant='(val##UL) '
  dbusint64_printf_modifier='"l"'
  ;;
$ac_cv_sizeof_long_long)
  dbusint64='long long'
  dbusint64_constant='(val##LL) '
  dbusuint64_constant='(val##ULL) '
  # Ideally we discover what the format is, but this is
  # only used in verbose mode, so eh...
  if test x"$ac_cv_gnu_library_2_1" = xyes; then
    dbusint64_printf_modifier='"ll"'
  fi
  ;;
$ac_cv_sizeof___int64)
  dbusint64=___int64
  dbusint64_constant='(val##i64) '
  dbusuint64_constant='(val##ui64) '
  # See above case
  if test x"$ac_cv_gnu_library_2_1" = xyes; then

```

```

        dbusint64_printf_modifier='"l1"'
    fi
    ;;
esac

if test "x$with_64_bit" = xno; then :

    DBUS_INT64_TYPE="no_int64_type_detected"
    DBUS_HAVE_INT64=0
    DBUS_INT64_CONSTANT=
    DBUS_UINT64_CONSTANT=
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: disabled via
--without-64-bit" >&5
$as_echo "disabled via --without-64-bit" >&6; }

elif test -z "$dbusint64"; then :
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: not found" >&5
$as_echo "not found" >&6; }
    as_fn_error $? "Could not find a 64-bit integer type.

```

Please report a bug here with details of your platform and compiler:

http://bugs.freedesktop.org/enter_bug.cgi?product=DBus&component=core

To compile D-Bus with all 64-bit integer types removed (not recommended), use the option "--without-64-bit".

This option is likely to be removed in future, unless you report that your platform needs it." "\$LINENO" 5

else

```

        DBUS_INT64_TYPE="$dbusint64"
        DBUS_HAVE_INT64=1
        DBUS_INT64_CONSTANT="$dbusint64_constant"
        DBUS_UINT64_CONSTANT="$dbusuint64_constant"
        if test x"$dbusint64_printf_modifier" != x; then

cat >>confdefs.h <<_ACEOF
#define DBUS_INT64_PRINTF_MODIFIER $dbusint64_printf_modifier
_ACEOF

        fi
        { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$DBUS_INT64_TYPE" >&5
$as_echo "$DBUS_INT64_TYPE" >&6; }

fi

```

```

### see what 32-bit int is called
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking 32-bit integer type"
>&5
$as_echo_n "checking 32-bit integer type... " >&6; }

case 4 in
$ac_cv_sizeof_short)
    dbusint32=short
    ;;
$ac_cv_sizeof_int)
    dbusint32=int
    ;;
$ac_cv_sizeof_long)
    dbusint32=long
    ;;
esac

if test -z "$dbusint32" ; then
    DBUS_INT32_TYPE="no_int32_type_detected"
    as_fn_error $? "No 32-bit integer type found" "$LINENO" 5
else
    DBUS_INT32_TYPE="$dbusint32"
    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$DBUS_INT32_TYPE" >&5
$as_echo "$DBUS_INT32_TYPE" >&6; }
fi

### see what 16-bit int is called
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking 16-bit integer type"
>&5
$as_echo_n "checking 16-bit integer type... " >&6; }

case 2 in
$ac_cv_sizeof_short)
    dbusint16=short
    ;;
$ac_cv_sizeof_int)
    dbusint16=int
    ;;
esac

if test -z "$dbusint16" ; then
    DBUS_INT16_TYPE="no_int16_type_detected"
    as_fn_error $? "No 16-bit integer type found" "$LINENO" 5
else

```

```

        DBUS_INT16_TYPE="$dbusint16"
        { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$DBUS_INT16_TYPE" >&5
$as_echo "$DBUS_INT16_TYPE" >&6; }
fi

## byte order
case $host_os in
  darwin*)
    # check at compile-time, so that it is possible to build
universal
    # (with multiple architectures at once on the compile line)

    ;;
  *)
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether
byte ordering is bigendian" >&5
$as_echo_n "checking whether byte ordering is bigendian... " >&6; }
if ${ac_cv_c_bigendian+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_cv_c_bigendian=unknown
  # See if we're dealing with a universal compiler.
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#endif __APPLE_CC__
    not a universal capable compiler
#endif
    typedef int dummy;

_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :

    # Check for potential -arch flags. It is not universal unless
    # there are at least two -arch flags with different values.
    ac_arch=
    ac_prev=
    for ac_word in $CC $CFLAGS $CPPFLAGS $LDFLAGS; do
      if test -n "$ac_prev"; then
        case $ac_word in
          i?86 | x86_64 | ppc | ppc64)
            if test -z "$ac_arch" || test "$ac_arch" = "$ac_word";
then
              ac_arch=$ac_word
            else
              ac_cv_c_bigendian=universal
            break
          fi
        ;;
      esac

```

```

        ac_prev=
    elif test "x$sac_word" = "x-arch"; then
        ac_prev=arch
    fi
done
fi
rm -f core conftest.err conftest.$sac_objext conftest.$sac_ext
if test $ac_cv_c_bigendian = unknown; then
    # See if sys/param.h defines the BYTE_ORDER macro.
    cat confdefs.h - <<_ACEOF >conftest.$sac_ext
/* end confdefs.h. */
#include <sys/types.h>
        #include <sys/param.h>

int
main ()
{
#if ! (defined BYTE_ORDER && defined BIG_ENDIAN \
        && defined LITTLE_ENDIAN && BYTE_ORDER && BIG_ENDIAN \
        && LITTLE_ENDIAN)
        bogus endian macros
#endif

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    # It does; now see whether it defined to BIG_ENDIAN or not.
    cat confdefs.h - <<_ACEOF >conftest.$sac_ext
/* end confdefs.h. */
#include <sys/types.h>
        #include <sys/param.h>

int
main ()
{
#if BYTE_ORDER != BIG_ENDIAN
        not big endian
#endif

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    ac_cv_c_bigendian=yes
else
    ac_cv_c_bigendian=no
fi
rm -f core conftest.err conftest.$sac_objext conftest.$sac_ext
fi

```

```

rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
if test $ac_cv_c_bigendian = unknown; then
  # See if <limits.h> defines _LITTLE_ENDIAN or _BIG_ENDIAN (e.g.,
  Solaris).
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <limits.h>

int
main ()
{
#if ! (defined _LITTLE_ENDIAN || defined _BIG_ENDIAN)
  bogus endian macros
#endif

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  # It does; now see whether it defined to _BIG_ENDIAN or not.
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <limits.h>

int
main ()
{
#ifdef _BIG_ENDIAN
  not big endian
#endif

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  ac_cv_c_bigendian=yes
else
  ac_cv_c_bigendian=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
if test $ac_cv_c_bigendian = unknown; then
  # Compile a test program.
  if test "$cross_compiling" = yes; then :
    # Try to guess by grepping values from an object file.
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```



```

short int ascii_mm[] =
    { 0x4249, 0x4765, 0x6E44, 0x6961, 0x6E53, 0x7953, 0 };
short int ascii_ii[] =
    { 0x694C, 0x5454, 0x656C, 0x6E45, 0x6944, 0x6E61, 0 };
int use_ascii (int i) {
    return ascii_mm[i] + ascii_ii[i];
}
short int ebcdic_ii[] =
    { 0x89D3, 0xE3E3, 0x8593, 0x95C5, 0x89C4, 0x9581, 0 };
short int ebcdic_mm[] =
    { 0xC2C9, 0xC785, 0x95C4, 0x8981, 0x95E2, 0xA8E2, 0 };
int use_ebcdic (int i) {
    return ebcdic_mm[i] + ebcdic_ii[i];
}
extern int foo;

int
main ()
{
return use_ascii (foo) == use_ebcdic (foo);
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    if grep BIGenDianSyS conftest.$ac_objext >/dev/null; then
        ac_cv_c_bigendian=yes
    fi
    if grep LiTTleEnDian conftest.$ac_objext >/dev/null ; then
        if test "$ac_cv_c_bigendian" = unknown; then
            ac_cv_c_bigendian=no
        else
            # finding both strings is unlikely to happen, but who
knows?
            ac_cv_c_bigendian=unknown
        fi
    fi
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
$ac_includes_default
int
main ()
{

    /* Are we little or big endian? From Harbison&Steele. */
union
{
    long int l;
    char c[sizeof (long int)];
}

```

```

        } u;
        u.l = 1;
        return u.c[sizeof (long int) - 1] == 1;

;
return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    ac_cv_c_bigendian=no
else
    ac_cv_c_bigendian=yes
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$sac_exeext \
    conftest.$sac_objext conftest.beam conftest.$sac_ext
fi

    fi
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_c_bigendian"
>&5
$as_echo "$ac_cv_c_bigendian" >&6; }
case $ac_cv_c_bigendian in #(
    yes)
    $as_echo "#define WORDS_BIGENDIAN 1" >>confdefs.h
;; #(
    no)
    ;; #(
    universal)

$as_echo "#define AC_APPLE_UNIVERSAL_BUILD 1" >>confdefs.h

    ;; #(
    *)
    as_fn_error $? "unknown endianness
    presetting ac_cv_c_bigendian=no (or yes) will help" "$LINENO" 5 ;;
esac

;;

esac

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for an
implementation of va_copy()" >&5
$as_echo_n "checking for an implementation of va_copy()... " >&6; }
if ${dbus_cv_va_copy+:} false; then :
    $as_echo_n "(cached) " >&6
else

    cat confdefs.h - <<_ACEOF >>conftest.$sac_ext
/* end confdefs.h. */
#include <stdarg.h>
#include <stdlib.h>

```

```

static void f (int i, ...) {
va_list args1, args2;
va_start (args1, i);
va_copy (args2, args1);
if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
    exit (1);
va_end (args1); va_end (args2);
}
int main() {
    f (0, 42);
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    dbus_cv_va_copy=yes
else
    dbus_cv_va_copy=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_cv_va_copy" >&5
$as_echo "$dbus_cv_va_copy" >&6; }
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for an
implementation of __va_copy()" >&5
$as_echo_n "checking for an implementation of __va_copy()... " >&6; }
if ${dbus_cv__va_copy+:} false; then :
    $as_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <stdarg.h>
#include <stdlib.h>
static void f (int i, ...) {
va_list args1, args2;
va_start (args1, i);
__va_copy (args2, args1);
if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
    exit (1);
va_end (args1); va_end (args2);
}
int main() {
    f (0, 42);
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    dbus_cv__va_copy=yes
else
    dbus_cv__va_copy=no

```

```

fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_cv___va_copy"
>&5
$as_echo "$dbus_cv___va_copy" >&6; }

if test "x$dbus_cv_va_copy" = "xyes"; then
    dbus_va_copy_func=va_copy
else if test "x$dbus_cv___va_copy" = "xyes"; then
    dbus_va_copy_func=__va_copy
fi
fi

if test -n "$dbus_va_copy_func"; then

cat >>confdefs.h <<_ACEOF
#define DBUS_VA_COPY $dbus_va_copy_func
_ACEOF

fi

ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether va_lists can
be copied by value" >&5
$as_echo_n "checking whether va_lists can be copied by value... " >&6;
}
if ${dbus_cv_va_val_copy+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test "$cross_compiling" = yes; then :
        dbus_cv_va_val_copy=yes
    else
        cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

        #include <stdarg.h>
        #include <stdlib.h>

int
main ()
{

    static void f (int i, ...) {

```

```

        va_list args1, args2;
        va_start (args1, i);
        args2 = args1;
        if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
            exit (1);
        va_end (args1); va_end (args2);
    }
    int main() {
        f (0, 42);
        return 0;
    }

;
return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    dbus_cv_va_val_copy=yes
else
    dbus_cv_va_val_copy=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$sac_exeext \
    conftest.$sac_objext conftest.beam conftest.$sac_ext
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_cv_va_val_copy"
>&5
$as_echo "$dbus_cv_va_val_copy" >&6; }
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$sac_ext >&5'
ac_link='$CC -o conftest$sac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$sac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

if test "x$dbus_cv_va_val_copy" = "xno"; then

$as_echo "#define DBUS_VA_COPY_AS_ARRAY 1" >>confdefs.h

fi

#### Atomic integers

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CC knows
__sync_sub_and_fetch()" >&5
$as_echo_n "checking whether $CC knows __sync_sub_and_fetch()... "
>&6; }
if ${dbus_cv_sync_sub_and_fetch+:} false; then :

```

```

    $sas_echo_n "(cached) " >&6
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{
int a = 4; int b = __sync_sub_and_fetch(&a, 4); exit(b);
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    dbus_cv_sync_sub_and_fetch=yes
else
    dbus_cv_sync_sub_and_fetch=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext

fi
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$dbus_cv_sync_sub_and_fetch" >&5
$sas_echo_n "$dbus_cv_sync_sub_and_fetch" >&6; }

if test "x$dbus_cv_sync_sub_and_fetch" = "xyes" ; then
    have_sync=1
else
    have_sync=0
fi

cat >>confdefs.h <<_ACEOF
#define DBUS_USE_SYNC $have_sync
_ACEOF

#### Various functions
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for library
containing socket" >&5
$sas_echo_n "checking for library containing socket... " >&6; }
if ${ac_cv_search_socket+:} false; then :
    $sas_echo_n "(cached) " >&6
else
    ac_func_search_save_LIBS=$LIBS
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC

```

```

    builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char socket ();
int
main ()
{
return socket ();
;
return 0;
}
_ACEOF
for ac_lib in ' ' socket network; do
if test -z "$ac_lib"; then
ac_res="none required"
else
ac_res=-l$ac_lib
LIBS="-l$ac_lib $ac_func_search_save_LIBS"
fi
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_search_socket=$ac_res
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext
if ${ac_cv_search_socket+:} false; then :
break
fi
done
if ${ac_cv_search_socket+:} false; then :

else
ac_cv_search_socket=no
fi
rm conftest.$ac_ext
LIBS=$ac_func_search_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_search_socket"
>&5
$as_echo "$ac_cv_search_socket" >&6; }
ac_res=$ac_cv_search_socket
if test "$ac_res" != no; then :
test "$ac_res" = "none required" || LIBS="$ac_res $LIBS"

fi

ac_fn_c_check_func "$LINENO" "gethostbyname"
"ac_cv_func_gethostbyname"
if test "x$ac_cv_func_gethostbyname" = xyes; then :

else

```

```

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for gethostbyname
in -lnsl" >&5
$as_echo_n "checking for gethostbyname in -lnsl... " >&6; }
if ${ac_cv_lib_nsl_gethostbyname+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lnsl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char gethostbyname ();
int
main ()
{
return gethostbyname ();
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_nsl_gethostbyname=yes
else
    ac_cv_lib_nsl_gethostbyname=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_nsl_gethostbyname" >&5
$as_echo "$ac_cv_lib_nsl_gethostbyname" >&6; }
if test "x$ac_cv_lib_nsl_gethostbyname" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_LIBNSL 1
_ACEOF

    LIBS="-lnsl $LIBS"

fi

fi

```



```

for ac_func in vsnprintf vasprintf nanosleep usleep setenv clearenv
unsetenv socketpair getgrouplist fpathconf setrlimit poll setlocale
localeconv strtoll strtoull issetugid getresuid
do :
    as_ac_var=`$as_echo "ac_cv_func_$ac_func" | $as_tr_sh`
ac_fn_c_check_func "$LINENO" "$ac_func" "$as_ac_var"
if eval test \"x\${$as_ac_var}\" = x"yes"; then :
    cat >>confdefs.h <<_ACEOF
#define `"$as_echo "HAVE_$ac_func" | $as_tr_cpp` 1
    _ACEOF

fi
done

for ac_header in syslog.h
do :
    ac_fn_c_check_header_mongrel "$LINENO" "syslog.h"
"ac_cv_header_syslog_h" "$ac_includes_default"
if test "x$ac_cv_header_syslog_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_SYSLOG_H 1
    _ACEOF

fi

done

if test "x$ac_cv_header_syslog_h" = "xyes"; then
    ac_fn_c_check_decl "$LINENO" "LOG_PERROR"
"ac_cv_have_decl_LOG_PERROR" "#include <syslog.h>
"
if test "x$ac_cv_have_decl_LOG_PERROR" = xyes; then :
    ac_have_decl=1
else
    ac_have_decl=0
fi

cat >>confdefs.h <<_ACEOF
#define HAVE_DECL_LOG_PERROR $ac_have_decl
    _ACEOF

fi

#### Check for broken poll; taken from Glib's configure

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for broken poll" >&5
$as_echo_n "checking for broken poll... " >&6; }
if test "$cross_compiling" = yes; then :
    broken_poll="no (cross compiling)"
else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext

```

```

/* end confdefs.h.  */

#include <stdlib.h>
#include <fcntl.h>
#include <poll.h>
#ifdef HAVE_SYS_POLL_H
#include <sys/poll.h>
#endif
int main(void) {
    struct pollfd fds[1];
    int fd;
    fd = open("/dev/null", 1);
    fds[0].fd = fd;
    fds[0].events = POLLIN;
    fds[0].revents = 0;
    if (poll(fds, 1, 0) < 0 || (fds[0].revents & POLLNVAL) != 0) {
        exit(1); /* Does not work for devices -- fail */
    }
    exit(0);
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    broken_poll=no
else
    broken_poll=yes

$as_echo "#define BROKEN_POLL 1" >>confdefs.h

fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$ac_exeext \
    conftest.$ac_objext conftest.beam conftest.$ac_ext
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $broken_poll" >&5
$as_echo "$broken_poll" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for dirfd" >&5
$as_echo_n "checking for dirfd... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

#include <sys/types.h>
#include <dirent.h>

int
main ()
{

DIR *dirp;
dirp = opendir(".");
dirfd(dirp);
closedir(dirp);

```

```

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    dbus_have_dirfd=yes
else
    dbus_have_dirfd=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_have_dirfd" >&5
$as_echo "$dbus_have_dirfd" >&6; }
if test "$dbus_have_dirfd" = yes; then

$as_echo "#define HAVE_DIRFD 1" >>confdefs.h

else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for DIR *dirp-
>dd_fd" >&5
$as_echo_n "checking for DIR *dirp->dd_fd... " >&6; }
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <sys/types.h>
#include <dirent.h>

int
main ()
{

DIR *dirp;
int fd;
dirp = opendir(".");
fd = dirp->dd_fd;
closedir(dirp);

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    dbus_have_ddfd=yes
else
    dbus_have_ddfd=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_have_ddfd"
>&5
$as_echo "$dbus_have_ddfd" >&6; }

```

```

        if test "$dbus_have_ddfd" = yes; then

$as_echo "#define HAVE_DDFD 1" >>confdefs.h

        fi
fi

for ac_header in sys/resource.h
do :
    ac_fn_c_check_header_mongrel "$LINENO" "sys/resource.h"
"ac_cv_header_sys_resource_h" "$ac_includes_default"
if test "x$ac_cv_header_sys_resource_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_SYS_RESOURCE_H 1
    _ACEOF

fi

done

for ac_header in dirent.h
do :
    ac_fn_c_check_header_mongrel "$LINENO" "dirent.h"
"ac_cv_header_dirent_h" "$ac_includes_default"
if test "x$ac_cv_header_dirent_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_DIRENT_H 1
    _ACEOF

fi

done

for ac_header in execinfo.h
do :
    ac_fn_c_check_header_mongrel "$LINENO" "execinfo.h"
"ac_cv_header_execinfo_h" "$ac_includes_default"
if test "x$ac_cv_header_execinfo_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_EXECINFO_H 1
    _ACEOF
    for ac_func in backtrace
do :
        ac_fn_c_check_func "$LINENO" "backtrace" "ac_cv_func_backtrace"
if test "x$ac_cv_func_backtrace" = xyes; then :
            cat >>confdefs.h <<_ACEOF
#define HAVE_BACKTRACE 1
                _ACEOF

fi

```

done

fi

done

for ac_header in errno.h

do :

ac_fn_c_check_header_mongrel "\$LINENO" "errno.h"

"ac_cv_header_errno_h" "\$ac_includes_default"

if test "x\$ac_cv_header_errno_h" = xyes; then :

cat >>confdefs.h <<_ACEOF

#define HAVE_ERRNO_H 1

_ACEOF

fi

done

for ac_header in signal.h

do :

ac_fn_c_check_header_mongrel "\$LINENO" "signal.h"

"ac_cv_header_signal_h" "\$ac_includes_default"

if test "x\$ac_cv_header_signal_h" = xyes; then :

cat >>confdefs.h <<_ACEOF

#define HAVE_SIGNAL_H 1

_ACEOF

fi

done

for ac_header in locale.h

do :

ac_fn_c_check_header_mongrel "\$LINENO" "locale.h"

"ac_cv_header_locale_h" "\$ac_includes_default"

if test "x\$ac_cv_header_locale_h" = xyes; then :

cat >>confdefs.h <<_ACEOF

#define HAVE_LOCALE_H 1

_ACEOF

fi

done

for ac_header in byteswap.h

do :

```
    ac_fn_c_check_header_mongrel "$LINENO" "byteswap.h"  
"ac_cv_header_byteswap_h" "$ac_includes_default"  
if test "x$ac_cv_header_byteswap_h" = xyes; then :  
    cat >>confdefs.h <<_ACEOF  
#define HAVE_BYTESWAP_H 1  
_ACEOF  
  
fi  
  
done
```

```
for ac_header in unistd.h  
do :  
    ac_fn_c_check_header_mongrel "$LINENO" "unistd.h"  
"ac_cv_header_unistd_h" "$ac_includes_default"  
if test "x$ac_cv_header_unistd_h" = xyes; then :  
    cat >>confdefs.h <<_ACEOF  
#define HAVE_UNISTD_H 1  
_ACEOF  
  
fi  
  
done
```

```
for ac_header in ws2tcpip.h  
do :  
    ac_fn_c_check_header_mongrel "$LINENO" "ws2tcpip.h"  
"ac_cv_header_ws2tcpip_h" "$ac_includes_default"  
if test "x$ac_cv_header_ws2tcpip_h" = xyes; then :  
    cat >>confdefs.h <<_ACEOF  
#define HAVE_WS2TCPIP_H 1  
_ACEOF  
  
fi  
  
done
```

```
for ac_header in wspiapi.h  
do :  
    ac_fn_c_check_header_mongrel "$LINENO" "wspiapi.h"  
"ac_cv_header_wspiapi_h" "$ac_includes_default"  
if test "x$ac_cv_header_wspiapi_h" = xyes; then :  
    cat >>confdefs.h <<_ACEOF  
#define HAVE_WSPIAPI_H 1  
_ACEOF  
  
fi  
  
done
```

```

# Add -D_POSIX_PTHREAD_SEMANTICS if on Solaris
#
case $host_os in
    solaris*)
        CFLAGS="$CFLAGS -D_POSIX_PTHREAD_SEMANTICS" ;;
esac

# checking for a posix version of getpwnam_r
# if we are cross compiling and can not run the test
# assume getpwnam_r is the posix version
# it is up to the person cross compiling to change
# this behavior if desired
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for posix
getpwnam_r" >&5
$as_echo_n "checking for posix getpwnam_r... " >&6; }
if ${ac_cv_func_posix_getpwnam_r+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :
    ac_cv_func_posix_getpwnam_r=yes

else
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <errno.h>
#include <pwd.h>

int
main ()
{

    char buffer[10000];
    struct passwd pwd, *pwptr = &pwd;
    int error;
    errno = 0;
    error = getpwnam_r ("", &pwd, buffer,
                      sizeof (buffer), &pwptr);
    return (error < 0 && errno == ENOSYS)
        || error == ENOSYS;

;
return 0;

```

```

}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
  ac_cv_func_posix_getpwnam_r=yes
else
  ac_cv_func_posix_getpwnam_r=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$ac_exeext \
  conftest.$ac_objext conftest.beam conftest.$ac_ext
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:"
$ac_cv_func_posix_getpwnam_r" >&5
$as_echo "$ac_cv_func_posix_getpwnam_r" >&6; }
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

if test "$ac_cv_func_posix_getpwnam_r" = yes; then

$as_echo "#define HAVE_POSIX_GETPWNAM_R 1" >>confdefs.h

else
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for nonposix
getpwnam_r" >&5
$as_echo_n "checking for nonposix getpwnam_r... " >&6; }
if ${ac_cv_func_nonposix_getpwnam_r+:} false; then :
  $as_echo_n "(cached) " >&6
else
  cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */
#include <pwd.h>
int
main ()
{
char buffer[10000];

                                struct passwd pwd;
                                getpwnam_r ("", &pwd, buffer,
                                                sizeof (buffer));

;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_func_nonposix_getpwnam_r=yes
else
  ac_cv_func_nonposix_getpwnam_r=no

```



```

fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_func_nonposix_getpwnam_r" >&5
$as_echo "$ac_cv_func_nonposix_getpwnam_r" >&6; }
    if test "$ac_cv_func_nonposix_getpwnam_r" = yes; then

$as_echo "#define HAVE_NONPOSIX_GETPWNAM_R 1" >>confdefs.h

        fi
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether socklen_t is
defined" >&5
$as_echo_n "checking whether socklen_t is defined... " >&6; }
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h.  */

#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>

int
main ()
{

socklen_t foo;
foo = 1;

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    dbus_have_socklen_t=yes
else
    dbus_have_socklen_t=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $dbus_have_socklen_t"
>&5
$as_echo "$dbus_have_socklen_t" >&6; }

if test "x$dbus_have_socklen_t" = "xyes"; then

$as_echo "#define HAVE_SOCKLEN_T 1" >>confdefs.h

fi

for ac_header in sys/uio.h

```

```

do :
  ac_fn_c_check_header_mongrel "$LINENO" "sys/uio.h"
  "ac_cv_header_sys_uio_h" "$ac_includes_default"
  if test "x$ac_cv_header_sys_uio_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_SYS_UIO_H 1
  _ACEOF
  for ac_func in writev
do :
  ac_fn_c_check_func "$LINENO" "writev" "ac_cv_func_writev"
  if test "x$ac_cv_func_writev" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_WRITEV 1
  _ACEOF

fi
done

fi

done

for ac_header in sys/syslimits.h
do :
  ac_fn_c_check_header_mongrel "$LINENO" "sys/syslimits.h"
  "ac_cv_header_sys_syslimits_h" "$ac_includes_default"
  if test "x$ac_cv_header_sys_syslimits_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_SYS_SYSLIMITS_H 1
  _ACEOF

fi

done

ac_fn_c_check_decl "$LINENO" "MSG_NOSIGNAL"
"ac_cv_have_decl_MSG_NOSIGNAL" " #include <sys/types.h>
#include <sys/socket.h>
"
if test "x$ac_cv_have_decl_MSG_NOSIGNAL" = xyes; then :
  ac_have_decl=1
else
  ac_have_decl=0
fi

cat >>confdefs.h <<_ACEOF
#define HAVE_DECL_MSG_NOSIGNAL $ac_have_decl
  _ACEOF

```

```

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for ISO C99 varargs
macros in C" >&5
$sas_echo_n "checking for ISO C99 varargs macros in C... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

int a(int p1, int p2, int p3);
#define call_a(...) a(1, __VA_ARGS__)
call_a(2,3);

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
dbus_have_iso_c_varargs=yes
else
dbus_have_iso_c_varargs=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: result:
$dbus_have_iso_c_varargs" >&5
$sas_echo "$dbus_have_iso_c_varargs" >&6; }

{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for GNUC varargs
macros" >&5
$sas_echo_n "checking for GNUC varargs macros... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

int a(int p1, int p2, int p3);
#define call_a(params...) a(1,params)
call_a(2,3);

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
dbus_have_gnuc_varargs=yes
else
dbus_have_gnuc_varargs=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$dbus_have_gnuc_varargs" >&5
$as_echo "$dbus_have_gnuc_varargs" >&6; }

if test x$dbus_have_iso_c_varargs = xyes; then

$as_echo "#define HAVE_ISO_VARARGS 1" >>confdefs.h

fi
if test x$dbus_have_gnuc_varargs = xyes; then

$as_echo "#define HAVE_GNUC_VARARGS 1" >>confdefs.h

fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for struct cmsgcred"
>&5
$as_echo_n "checking for struct cmsgcred... " >&6; }
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h.  */

#include <sys/types.h>
#include <sys/socket.h>

int
main ()
{

struct cmsgcred cred;

cred.cmcred_pid = 0;

;
return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
dbus_have_struct_cmsgcred=yes
else
dbus_have_struct_cmsgcred=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$dbus_have_struct_cmsgcred" >&5
$as_echo "$dbus_have_struct_cmsgcred" >&6; }

if test x$dbus_have_struct_cmsgcred = xyes; then

$as_echo "#define HAVE_CMSGCRED 1" >>confdefs.h

fi

```

```

for ac_func in getpeerucred getpeereid
do :
  as_ac_var=`$as_echo "ac_cv_func_$ac_func" | $as_tr_sh`
ac_fn_c_check_func "$LINENO" "$ac_func" "$as_ac_var"
if eval test \"x\$$as_ac_var\" = x"yes"; then :
  cat >>confdefs.h <<_ACEOF
#define ` $as_echo "HAVE_$ac_func" | $as_tr_cpp` 1
_ACEOF

fi
done

for ac_func in pipe2 accept4
do :
  as_ac_var=`$as_echo "ac_cv_func_$ac_func" | $as_tr_sh`
ac_fn_c_check_func "$LINENO" "$ac_func" "$as_ac_var"
if eval test \"x\$$as_ac_var\" = x"yes"; then :
  cat >>confdefs.h <<_ACEOF
#define ` $as_echo "HAVE_$ac_func" | $as_tr_cpp` 1
_ACEOF

fi
done

#### Abstract sockets

if test x$enable_abstract_sockets = xauto; then
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$ac_ext >&5'
ac_link='$CC -o conftest$ac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$ac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

warn_on_xcompile=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking abstract socket
namespace" >&5
$as_echo_n "checking abstract socket namespace... " >&6; }
if ${ac_cv_have_abstract_sockets+:} false; then :
  $as_echo_n "(cached) " >&6
else
  if test "$cross_compiling" = yes; then :

      ac_cv_have_abstract_sockets=no
      warn_on_xcompile=yes

  else
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

```

```

#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <errno.h>

int
main ()
{

    size_t slen;
    int listen_fd;
    struct sockaddr_un addr;

    listen_fd = socket (PF_UNIX, SOCK_STREAM, 0);

    if (listen_fd < 0)
    {
        fprintf (stderr, "socket() failed: %s\n", strerror (errno));
        exit (1);
    }

    memset (&addr, '\0', sizeof (addr));
    addr.sun_family = AF_UNIX;
    strcpy (addr.sun_path, "X/tmp/dbus-fake-socket-path-used-in-
configure-test");
    /* SUN_LEN uses strlen() so need to calculate it before adding \0 at
the
* beginning.
*/
    slen = SUN_LEN(&addr);
    addr.sun_path[0] = '\0'; /* this is what makes it abstract */

    if (bind (listen_fd, (struct sockaddr*) &addr, slen) < 0)
    {
        fprintf (stderr, "Abstract socket namespace bind() failed:
%s\n",
                strerror (errno));
        exit (1);
    }
    else
        exit (0);

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_run "$LINENO"; then :
    ac_cv_have_abstract_sockets=yes

```

```

else
  ac_cv_have_abstract_sockets=no
fi
rm -f core *.core core.conftest.* gmon.out bb.out conftest$sac_exeext \
  conftest.$sac_objext conftest.beam conftest.$sac_ext
fi

fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$sac_cv_have_abstract_sockets" >&5
$as_echo "$sac_cv_have_abstract_sockets" >&6; }
if test x$warn_on_xcompile = xyes ; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: Cannot check for
abstract sockets when cross-compiling, please use --enable-abstract-
sockets" >&5
$as_echo "$as_me: WARNING: Cannot check for abstract sockets when
cross-compiling, please use --enable-abstract-sockets" >&2;}
fi
ac_ext=c
ac_cpp='$CPP $CPPFLAGS'
ac_compile='$CC -c $CFLAGS $CPPFLAGS conftest.$sac_ext >&5'
ac_link='$CC -o conftest$sac_exeext $CFLAGS $CPPFLAGS $LDFLAGS
conftest.$sac_ext $LIBS >&5'
ac_compiler_gnu=$ac_cv_c_compiler_gnu

fi

if test x$enable_abstract_sockets = xyes; then
  if test x$sac_cv_have_abstract_sockets = xno; then
    as_fn_error $? "Abstract sockets explicitly required, and support
not detected." "$LINENO" 5
  fi
fi

if test x$enable_abstract_sockets = xno; then
  ac_cv_have_abstract_sockets=no;
fi

if test x$sac_cv_have_abstract_sockets = xyes ; then
  DBUS_PATH_OR_ABSTRACT=abstract

$as_echo "#define HAVE_ABSTRACT_SOCKETS 1" >>confdefs.h

else
  DBUS_PATH_OR_ABSTRACT=path
fi

# this is used in addresses to prefer abstract, e.g.
# unix:path=/foo or unix:abstract=/foo

```

```

if test "x${ac_cv_env_PKG_CONFIG_set}" != "xset"; then
    if test -n "${ac_tool_prefix}"; then
        # Extract the first word of "${ac_tool_prefix}pkg-config", so it can
        be a program name with args.
        set dummy ${ac_tool_prefix}pkg-config; ac_word=$2
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
        $as_echo_n "checking for $ac_word... " >&6; }
        if ${ac_cv_path_PKG_CONFIG+:} false; then :
            $as_echo_n "(cached) " >&6
        else
            case $PKG_CONFIG in
                [\\/] * | ?:[\\/] *)
                    ac_cv_path_PKG_CONFIG="$PKG_CONFIG" # Let the user override the test
                    with a path.
                    ;;
                *)
                    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
                    for as_dir in $PATH
                    do
                        IFS=$as_save_IFS
                        test -z "$as_dir" && as_dir=.
                        for ac_exec_ext in ' ${ac_executable_extensions}; do
                            if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
                                ac_cv_path_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
                                $as_echo "$as_me:${as_lineno-$LINENO}: found
                                $as_dir/$ac_word$ac_exec_ext" >&5
                                break 2
                            fi
                        done
                    done
                    IFS=$as_save_IFS

                    ;;
                esac
            fi
            PKG_CONFIG=$ac_cv_path_PKG_CONFIG
            if test -n "$PKG_CONFIG"; then
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: $PKG_CONFIG" >&5
                $as_echo "$PKG_CONFIG" >&6; }
            else
                { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
                $as_echo "no" >&6; }
            fi

        fi

        if test -z "${ac_cv_path_PKG_CONFIG}"; then
            ac_pt_PKG_CONFIG=$PKG_CONFIG
            # Extract the first word of "pkg-config", so it can be a program
            name with args.
            set dummy pkg-config; ac_word=$2

```



```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_ac_pt_PKG_CONFIG+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $ac_pt_PKG_CONFIG in
    [\\/* | ?:[\\/*]*)
      ac_cv_path_ac_pt_PKG_CONFIG="$ac_pt_PKG_CONFIG" # Let the user
      override the test with a path.
      ;;
    *)
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
      for as_dir in $PATH
      do
        IFS=$as_save_IFS
        test -z "$as_dir" && as_dir=.
        for ac_exec_ext in ' $ac_executable_extensions; do
          if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
            ac_cv_path_ac_pt_PKG_CONFIG="$as_dir/$ac_word$ac_exec_ext"
            $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
            break 2
          fi
        done
      done
      IFS=$as_save_IFS

      ;;
    esac
  fi
  ac_pt_PKG_CONFIG=$ac_cv_path_ac_pt_PKG_CONFIG
  if test -n "$ac_pt_PKG_CONFIG"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_pt_PKG_CONFIG"
    >&5
    $as_echo "$ac_pt_PKG_CONFIG" >&6; }
  else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
    $as_echo "no" >&6; }
  fi

  if test "x$ac_pt_PKG_CONFIG" = x; then
    PKG_CONFIG=""
  else
    case $cross_compiling:$ac_tool_warned in
    yes:)
      { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: using cross tools
      not prefixed with host triplet" >&5
      $as_echo "$as_me: WARNING: using cross tools not prefixed with host
      triplet" >&2;}
      ac_tool_warned=yes ;;
    esac
    PKG_CONFIG=$ac_pt_PKG_CONFIG

```

```

    fi
else
    PKG_CONFIG="$ac_cv_path_PKG_CONFIG"
fi

fi
if test -n "$PKG_CONFIG"; then
    _pkg_min_version=0.9.0
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking pkg-config is
at least version $_pkg_min_version" >&5
$as_echo_n "checking pkg-config is at least version
$_pkg_min_version... " >&6; }
    if $PKG_CONFIG --atleast-pkgconfig-version $_pkg_min_version;
then
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    else
        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        PKG_CONFIG=""
    fi
fi

fi

#### Sort out XML library

# see what we have
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for
XML_ParserCreate_MM in -lexpat" >&5
$as_echo_n "checking for XML_ParserCreate_MM in -lexpat... " >&6; }
if ${ac_cv_lib_expat_XML_ParserCreate_MM+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lexpat $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char XML_ParserCreate_MM ();
int
main ()
{
return XML_ParserCreate_MM ();
;
return 0;
}

```

```

_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_expat_XML_ParserCreate_MM=yes
else
  ac_cv_lib_expat_XML_ParserCreate_MM=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_expat_XML_ParserCreate_MM" >&5
$as_echo "$ac_cv_lib_expat_XML_ParserCreate_MM" >&6; }
if test "x$ac_cv_lib_expat_XML_ParserCreate_MM" = xyes; then :
  for ac_header in expat.h
  do :
    ac_fn_c_check_header_mongrel "$LINENO" "expat.h"
    "ac_cv_header_expat_h" "$ac_includes_default"
    if test "x$ac_cv_header_expat_h" = xyes; then :
      cat >>confdefs.h <<_ACEOF
#define HAVE_EXPAT_H 1
_ACEOF
      have_expat=true
    else
      have_expat=false
    fi
  done

  else
    have_expat=false
  fi

# see what we want to use
dbus_use_libxml=false
dbus_use_expat=false
if test x$with_xml = xexpat; then
  if ! $have_expat ; then
    as_fn_error $? "Explicitly requested expat but expat not
found" "$LINENO" 5
  fi
  dbus_use_expat=true
elif test x$with_xml = xlibxml; then

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for LIBXML" >&5
$as_echo_n "checking for LIBXML... " >&6; }

if test -n "$LIBXML_CFLAGS"; then
  pkg_cv_LIBXML_CFLAGS="$LIBXML_CFLAGS"
  elif test -n "$PKG_CONFIG"; then

```

```

    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"libxml-2.0 >= 2.6.0\""; } >&5
        ($PKG_CONFIG --exists --print-errors "libxml-2.0 >= 2.6.0") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_LIBXML_CFLAGS=`$PKG_CONFIG --cflags "libxml-2.0 >= 2.6.0"
2>/dev/null`
    else
        pkg_failed=yes
    fi
    else
        pkg_failed=untried
    fi
if test -n "$LIBXML_LIBS"; then
    pkg_cv_LIBXML_LIBS="$LIBXML_LIBS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"libxml-2.0 >= 2.6.0\""; } >&5
        ($PKG_CONFIG --exists --print-errors "libxml-2.0 >= 2.6.0") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_LIBXML_LIBS=`$PKG_CONFIG --libs "libxml-2.0 >= 2.6.0"
2>/dev/null`
    else
        pkg_failed=yes
    fi
    else
        pkg_failed=untried
    fi

if test $pkg_failed = yes; then

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        LIBXML_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "libxml-2.0 >= 2.6.0" 2>&1`
    else
        LIBXML_PKG_ERRORS=`$PKG_CONFIG --print-errors "libxml-2.0
>= 2.6.0" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$LIBXML_PKG_ERRORS" >&5

```

```

        { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        have_libxml=false
elif test $pkg_failed = untried; then
    have_libxml=false
else
    LIBXML_CFLAGS=$pkg_cv_LIBXML_CFLAGS
    LIBXML_LIBS=$pkg_cv_LIBXML_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    have_libxml=true
fi
    if ! $have_libxml ; then
        as_fn_error $? "Explicitly requested libxml but libxml not
found" "$LINENO" 5
    fi
    dbus_use_libxml=true
else
    ### expat is the default because libxml can't currently
survive
    ### our brutal OOM-handling unit test setup.
    ### http://bugzilla.gnome.org/show_bug.cgi?id=109368
    if test x$have_expat = xfalse; then
        as_fn_error $? "Could not find expat.h, check
config.log for failed attempts" "$LINENO" 5
    fi
    ### By default, only use Expat since it's tested and known to
work.  If you're a
    ### general-purpose OS vendor, please don't enable libxml.  For
embedded use
    ### if your OS is built around libxml, that's another case.
    dbus_use_expat=true
fi

if $dbus_use_expat; then
    DBUS_USE_EXPAT_TRUE=
    DBUS_USE_EXPAT_FALSE='#'
else
    DBUS_USE_EXPAT_TRUE='#'
    DBUS_USE_EXPAT_FALSE=
fi

if $dbus_use_libxml; then
    DBUS_USE_LIBXML_TRUE=
    DBUS_USE_LIBXML_FALSE='#'
else
    DBUS_USE_LIBXML_TRUE='#'
    DBUS_USE_LIBXML_FALSE=
fi

```

```

if $dbus_use_expat; then
    XML_LIBS=-lexpat
    XML_CFLAGS=
fi
if $dbus_use_libxml; then
    XML_LIBS=$LIBXML_LIBS
    XML_CFLAGS=$LIBXML_CFLAGS
fi

# Thread lib detection
ac_fn_c_check_func "$LINENO" "pthread_cond_timedwait"
"ac_cv_func_pthread_cond_timedwait"
if test "x$ac_cv_func_pthread_cond_timedwait" = xyes; then :
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
pthread_cond_timedwait in -lpthread" >&5
$as_echo_n "checking for pthread_cond_timedwait in -lpthread... " >&6;
}
if ${ac_cv_lib_pthread_pthread_cond_timedwait+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lpthread $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char pthread_cond_timedwait ();
int
main ()
{
return pthread_cond_timedwait ();
    ;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_pthread_pthread_cond_timedwait=yes
else
    ac_cv_lib_pthread_pthread_cond_timedwait=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_pthread_pthread_cond_timedwait" >&5
$as_echo "$ac_cv_lib_pthread_pthread_cond_timedwait" >&6; }
if test "x$ac_cv_lib_pthread_pthread_cond_timedwait" = xyes; then :
  THREAD_LIBS="-lpthread"
fi

fi

save_libs="$LIBS"
LIBS="$LIBS $THREAD_LIBS"
ac_fn_c_check_func "$LINENO" "pthread_condattr_setclock"
"ac_cv_func_pthread_condattr_setclock"
if test "x$ac_cv_func_pthread_condattr_setclock" = xyes; then :
  have_pthread_condattr_setclock=true
else
  have_pthread_condattr_setclock=false
fi

if test x$have_pthread_condattr_setclock = xtrue; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for library
containing clock_getres" >&5
$as_echo_n "checking for library containing clock_getres... " >&6; }
if ${ac_cv_search_clock_getres+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_func_search_save_LIBS=$LIBS
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char clock_getres ();
int
main ()
{
return clock_getres ();
  ;
  return 0;
}
_ACEOF
for ac_lib in ' rt; do
  if test -z "$ac_lib"; then
    ac_res="none required"
  else
    ac_res=-l$ac_lib
    LIBS="-l$ac_lib $ac_func_search_save_LIBS"
  fi

```

```

    if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_search_clock_getres=$ac_res
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext
    if ${ac_cv_search_clock_getres+:} false; then :
    break
fi
done
if ${ac_cv_search_clock_getres+:} false; then :

else
    ac_cv_search_clock_getres=no
fi
rm conftest.$ac_ext
LIBS=$ac_func_search_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_search_clock_getres" >&5
$as_echo "$ac_cv_search_clock_getres" >&6; }
ac_res=$ac_cv_search_clock_getres
if test "$ac_res" != no; then :
    test "$ac_res" = "none required" || LIBS="$ac_res $LIBS"
    THREAD_LIBS="$THREAD_LIBS -lrt"
fi

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
CLOCK_MONOTONIC" >&5
$as_echo_n "checking for CLOCK_MONOTONIC... " >&6; }
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <time.h>
#include <pthread.h>

int
main ()
{

struct timespec monotonic_timer;
pthread_condattr_t attr;
pthread_condattr_init (&attr);
pthread_condattr_setclock (&attr, CLOCK_MONOTONIC);
clock_getres (CLOCK_MONOTONIC,&monotonic_timer);

;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    have_clock_monotonic=true
else
    have_clock_monotonic=false

```



```

fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
if test x$have_clock_monotonic = xtrue; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: found" >&5
$as_echo "found" >&6; }

$as_echo "#define HAVE_MONOTONIC_CLOCK 1" >>confdefs.h

else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: not found" >&5
$as_echo "not found" >&6; }
fi
fi
LIBS="$save_libs"

# SELinux detection
if test x$enable_selinux = xno ; then
  have_selinux=no;
else
  # See if we have SELinux library
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
is_selinux_enabled in -lselinux" >&5
$as_echo_n "checking for is_selinux_enabled in -lselinux... " >&6; }
if ${ac_cv_lib_selinux_is_selinux_enabled+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-lselinux $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char is_selinux_enabled ();
int
main ()
{
return is_selinux_enabled ();
  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_selinux_is_selinux_enabled=yes
else
  ac_cv_lib_selinux_is_selinux_enabled=no

```

```

fi
rm -f core conftest.err conftest.$_sac_objext \
    conftest$_sac_exeext conftest.$_sac_ext
LIBS=$_sac_check_lib_save_LIBS
fi
{ $_sac_echo "$_sac_me:${as_lineno-$LINENO}: result:
$_sac_cv_lib_selinux_is_selinux_enabled" >&5
$_sac_echo "$_sac_cv_lib_selinux_is_selinux_enabled" >&6; }
if test "x$_sac_cv_lib_selinux_is_selinux_enabled" = xyes; then :
    have_selinux=yes
else
    have_selinux=no
fi

# see if we have the SELinux header with the new D-Bus stuff in it
if test x$have_selinux = xyes ; then
    { $_sac_echo "$_sac_me:${as_lineno-$LINENO}: checking for DBUS
Flask permissions in selinux/av_permissions.h" >&5
$_sac_echo_n "checking for DBUS Flask permissions in
selinux/av_permissions.h... " >&6; }
    cat confdefs.h - <<_ACEOF >>conftest.$_sac_ext
/* end confdefs.h. */
#include <selinux/av_permissions.h>
int
main ()
{
#ifdef DBUS__ACQUIRE_SVC return 0;
    #else
    #error DBUS__ACQUIRE_SVC not defined
    #endif
;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    have_selinux=yes
else
    have_selinux=no
fi
rm -f core conftest.err conftest.$_sac_objext conftest.$_sac_ext
    { $_sac_echo "$_sac_me:${as_lineno-$LINENO}: result: $have_selinux"
>&5
$_sac_echo "$have_selinux" >&6; }
fi

if test x$enable_selinux = xauto ; then
    if test x$have_selinux = xno ; then
        { $_sac_echo "$_sac_me:${as_lineno-$LINENO}: WARNING:
Sufficiently new SELinux library not found" >&5
$_sac_echo "$_sac_me: WARNING: Sufficiently new SELinux library not found"
>&2;}

```

```

        fi
    else
        if test x$have_selinux = xno ; then
            as_fn_error $? "SElinux explicitly required, and
SELinux library not found" "$LINENO" 5
        fi
    fi
fi

if test x$have_selinux = xyes; then
    HAVE_SELINUX_TRUE=
    HAVE_SELINUX_FALSE='#'
else
    HAVE_SELINUX_TRUE='#'
    HAVE_SELINUX_FALSE=
fi

if test x$have_selinux = xyes ; then
    # the selinux code creates threads
    # which requires libpthread even on linux
    ac_fn_c_check_func "$LINENO" "pthread_create"
"ac_cv_func_pthread_create"
if test "x$ac_cv_func_pthread_create" = xyes; then :

else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for pthread_create
in -lpthread" >&5
$as_echo_n "checking for pthread_create in -lpthread... " >&6; }
if ${ac_cv_lib_pthread_pthread_create+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lpthread $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char pthread_create ();
int
main ()
{
return pthread_create ();
;
return 0;
}
_ACEOF

```

```

if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_pthread_pthread_create=yes
else
  ac_cv_lib_pthread_pthread_create=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_pthread_pthread_create" >&5
$as_echo "$ac_cv_lib_pthread_pthread_create" >&6; }
if test "x$ac_cv_lib_pthread_pthread_create" = xyes; then :
  SELINUX_THREAD_LIBS="-lpthread"
fi

fi

  SELINUX_LIBS="-lselinux $SELINUX_THREAD_LIBS"

$as_echo "#define HAVE_SELINUX 1" >>confdefs.h

else
  SELINUX_LIBS=
fi

# inotify checks
if test x$enable_inotify = xno ; then
  have_inotify=no;
else
  for ac_header in sys/inotify.h
do :
  ac_fn_c_check_header_mongrel "$LINENO" "sys/inotify.h"
"ac_cv_header_sys_inotify_h" "$ac_includes_default"
if test "x$ac_cv_header_sys_inotify_h" = xyes; then :
  cat >>confdefs.h <<_ACEOF
#define HAVE_SYS_INOTIFY_H 1
_ACEOF
  have_inotify=yes
else
  have_inotify=no
fi
done

fi

if test x$have_inotify = xyes; then

$as_echo "#define DBUS_BUS_ENABLE_INOTIFY 1" >>confdefs.h

```

```

    for ac_func in inotify_init1
do :
    ac_fn_c_check_func "$LINENO" "inotify_init1"
"ac_cv_func_inotify_init1"
if test "x$ac_cv_func_inotify_init1" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_INOTIFY_INIT1 1
_ACEOF

fi
done

fi

    if test x$have_inotify = xyes; then
        DBUS_BUS_ENABLE_INOTIFY_TRUE=
        DBUS_BUS_ENABLE_INOTIFY_FALSE='#'
    else
        DBUS_BUS_ENABLE_INOTIFY_TRUE='#'
        DBUS_BUS_ENABLE_INOTIFY_FALSE=
    fi

# dnotify checks
if test x$enable_dnotify = xno ; then
    have_dnotify=no;
else
    if test x$have_inotify = xno -a x$host_os = xlinux-gnu -o
x$host_os = xlinux; then
        have_dnotify=yes;
    else
        have_dnotify=no;
    fi
fi

if test x$have_dnotify = xyes; then

$as_echo "#define DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX 1" >>confdefs.h

fi

    if test x$have_dnotify = xyes; then
        DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE=
        DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE='#'
    else
        DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE='#'
        DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE=
    fi

# For simplicity, we require the userland API for epoll_create1 at
# compile-time (glibc 2.9), but we'll run on kernels that turn out

```

```

# not to have it at runtime.
# Check whether --enable-epoll was given.
if test "${enable_epoll+set}" = set; then :
    enableval=$enable_epoll; enable_epoll=$enableval
else
    enable_epoll=auto
fi

if test x$enable_epoll = xno; then
    have_linux_epoll=no
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for Linux
epoll(4)" >&5
$as_echo_n "checking for Linux epoll(4)... " >&6; }
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

    #ifndef __linux__
    #error This is not Linux
    #endif
    #include <sys/epoll.h>

int
main ()
{
epoll_create1 (EPOLL_CLOEXEC);
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    have_linux_epoll=yes
else
    have_linux_epoll=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
    { $as_echo "$as_me:${as_lineno-$LINENO}: result:
$have_linux_epoll" >&5
$as_echo "$have_linux_epoll" >&6; }
fi
if test x$enable_epoll,$have_linux_epoll = xyes,no; then
    as_fn_error $? "epoll support explicitly enabled but not
available" "$LINENO" 5
fi
if test x$have_linux_epoll = xyes; then

$as_echo "#define DBUS_HAVE_LINUX_EPOLL 1" >>confdefs.h

fi
if test x$have_linux_epoll = xyes; then
    HAVE_LINUX_EPOLL_TRUE=

```

```

    HAVE_LINUX_EPOLL_FALSE='#'
else
    HAVE_LINUX_EPOLL_TRUE='#'
    HAVE_LINUX_EPOLL_FALSE=
fi

# kqueue checks
if test x$enable_kqueue = xno ; then
    have_kqueue=no
else
    have_kqueue=yes
    ac_fn_c_check_header_mongrel "$LINENO" "sys/event.h"
"ac_cv_header_sys_event_h" "$ac_includes_default"
if test "x$ac_cv_header_sys_event_h" = xyes; then :

else
    have_kqueue=no
fi

    ac_fn_c_check_func "$LINENO" "kqueue" "ac_cv_func_kqueue"
if test "x$ac_cv_func_kqueue" = xyes; then :

else
    have_kqueue=no
fi

    if test x$enable_kqueue = xyes -a x$have_kqueue = xno; then
        as_fn_error $? "kqueue support explicitly enabled but not
available" "$LINENO" 5
    fi
fi

if test x$have_kqueue = xyes; then

$as_echo "#define DBUS_BUS_ENABLE_KQUEUE 1" >>confdefs.h

fi

if test x$have_kqueue = xyes; then
    DBUS_BUS_ENABLE_KQUEUE_TRUE=
    DBUS_BUS_ENABLE_KQUEUE_FALSE='#'
else
    DBUS_BUS_ENABLE_KQUEUE_TRUE='#'
    DBUS_BUS_ENABLE_KQUEUE_FALSE=
fi

# launchd checks
if test x$enable_launchd = xno ; then

```

```

    have_launchd=no
else
    have_launchd=yes
    ac_fn_c_check_header_mongrel "$LINENO" "launch.h"
"ac_cv_header_launch_h" "$ac_includes_default"
if test "x$ac_cv_header_launch_h" = xyes; then :

else
    have_launchd=no
fi

    # Extract the first word of "launchctl", so it can be a program
name with args.
set dummy launchctl; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_LAUNCHCTL+:} false; then :
    $as_echo_n "(cached) " >&6
else
    case $LAUNCHCTL in
    [\\/* | ?:[\\/*]*)
    ac_cv_path_LAUNCHCTL="$LAUNCHCTL" # Let the user override the test
with a path.
    ;;
    *)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in ' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_LAUNCHCTL="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
    fi
done
done
IFS=$as_save_IFS

    ;;
esac
fi
LAUNCHCTL=$ac_cv_path_LAUNCHCTL
if test -n "$LAUNCHCTL"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $LAUNCHCTL" >&5
$as_echo "$LAUNCHCTL" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }

```



```

fi

    if test "x$LAUNCHCTL" = "x"; then
        have_launchd=no
    fi

    if test x$enable_launchd = xyes && test x$have_launchd = xno ;
then
        as_fn_error $? "launchd support explicitly enabled but not
available" "$LINENO" 5
    fi
fi

if test x$have_launchd = xyes; then

$as_echo "#define DBUS_ENABLE_LAUNCHD 1" >>confdefs.h

fi

    if test x$have_launchd = xyes; then
        DBUS_ENABLE_LAUNCHD_TRUE=
        DBUS_ENABLE_LAUNCHD_FALSE='#'
    else
        DBUS_ENABLE_LAUNCHD_TRUE='#'
        DBUS_ENABLE_LAUNCHD_FALSE=
    fi

#### Directory to place launchd agent file
if test "x$with_launchd_agent_dir" = "x"; then
    LAUNCHD_AGENT_DIR="/Library/LaunchAgents"
else
    LAUNCHD_AGENT_DIR="$with_launchd_agent_dir"
fi

if test x$enable_console_owner_file = xno ; then
    have_console_owner_file=no;
else
    case $host_os in
    solaris*)
        have_console_owner_file=yes;

$as_echo "#define HAVE_CONSOLE_OWNER_FILE 1" >>confdefs.h

        ;;
    *)
        have_console_owner_file=no;;
    esac
fi

```

```

if test x$have_console_owner_file = xyes; then
    HAVE_CONSOLE_OWNER_FILE_TRUE=
    HAVE_CONSOLE_OWNER_FILE_FALSE='#'
else
    HAVE_CONSOLE_OWNER_FILE_TRUE='#'
    HAVE_CONSOLE_OWNER_FILE_FALSE=
fi

if test x$enable_systemd = xno ; then
    have_systemd=no;
else

pkg_failed=no
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for SYSTEMD" >&5
$as_echo_n "checking for SYSTEMD... " >&6; }

if test -n "$SYSTEMD_CFLAGS"; then
    pkg_cv_SYSTEMD_CFLAGS="$SYSTEMD_CFLAGS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"libsistemd-login >= 32, libsistemd-daemon >= 32\""; }
>&5
        ($PKG_CONFIG --exists --print-errors "libsistemd-login >= 32,
libsistemd-daemon >= 32") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_SYSTEMD_CFLAGS=`$PKG_CONFIG --cflags "libsistemd-login >= 32,
libsistemd-daemon >= 32" 2>/dev/null`
    else
        pkg_failed=yes
    fi
else
    pkg_failed=untried
fi

if test -n "$SYSTEMD_LIBS"; then
    pkg_cv_SYSTEMD_LIBS="$SYSTEMD_LIBS"
elif test -n "$PKG_CONFIG"; then
    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"libsistemd-login >= 32, libsistemd-daemon >= 32\""; }
>&5
        ($PKG_CONFIG --exists --print-errors "libsistemd-login >= 32,
libsistemd-daemon >= 32") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
        test $ac_status = 0; }; then
        pkg_cv_SYSTEMD_LIBS=`$PKG_CONFIG --libs "libsistemd-login >= 32,
libsistemd-daemon >= 32" 2>/dev/null`

```

```

else
    pkg_failed=yes
fi
else
    pkg_failed=untried
fi

if test $pkg_failed = yes; then

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
    _pkg_short_errors_supported=yes
else
    _pkg_short_errors_supported=no
fi
    if test $_pkg_short_errors_supported = yes; then
        SYSTEMD_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "libsystemd-login >= 32, libsystemd-daemon >= 32" 2>&1`
    else
        SYSTEMD_PKG_ERRORS=`$PKG_CONFIG --print-errors
"libsystemd-login >= 32, libsystemd-daemon >= 32" 2>&1`
    fi
    # Put the nasty error message in config.log where it belongs
    echo "$SYSTEMD_PKG_ERRORS" >&5

    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
        have_systemd=no
elif test $pkg_failed = untried; then
    have_systemd=no
else
    SYSTEMD_CFLAGS=$pkg_cv_SYSTEMD_CFLAGS
    SYSTEMD_LIBS=$pkg_cv_SYSTEMD_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    have_systemd=yes
fi
fi

if test x$have_systemd = xyes; then

$as_echo "#define HAVE_SYSTEMD 1" >>confdefs.h

fi

if test x$enable_systemd = xyes -a x$have_systemd != xyes ; then
    as_fn_error $? "Explicitly requested systemd support, but systemd
not found" "$LINENO" 5
fi

# libaudit detection

```

```

if test x$enable_libaudit = xno ; then
    have_libaudit=no;
else
    # See if we have audit daemon & capabilities library
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
audit_log_user_avc_message in -laudit" >&5
$as_echo_n "checking for audit_log_user_avc_message in -laudit... "
>&6; }
if ${ac_cv_lib_audit_audit_log_user_avc_message+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-laudit $LIBS"
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char audit_log_user_avc_message ();
int
main ()
{
return audit_log_user_avc_message ();
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_audit_audit_log_user_avc_message=yes
else
    ac_cv_lib_audit_audit_log_user_avc_message=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_audit_audit_log_user_avc_message" >&5
$as_echo "$ac_cv_lib_audit_audit_log_user_avc_message" >&6; }
if test "x$ac_cv_lib_audit_audit_log_user_avc_message" = xyes; then :
    have_libaudit=yes
else
    have_libaudit=no
fi

    if test x$have_libaudit = xyes ; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
capng_clear in -lcap-ng" >&5

```

```

$as_echo_n "checking for capng_clear in -lcap-ng... " >&6; }
if ${ac_cv_lib_cap_ng_capng_clear+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-lcap-ng $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char capng_clear ();
int
main ()
{
return capng_clear ();
  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_cap_ng_capng_clear=yes
else
  ac_cv_lib_cap_ng_capng_clear=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_cap_ng_capng_clear" >&5
$as_echo "$ac_cv_lib_cap_ng_capng_clear" >&6; }
if test "x$ac_cv_lib_cap_ng_capng_clear" = xyes; then :
  have_libaudit=yes
else
  have_libaudit=no
fi

  fi
fi

  if test x$have_libaudit = xyes; then
    HAVE_LIBAUDIT_TRUE=
    HAVE_LIBAUDIT_FALSE='#'
  else
    HAVE_LIBAUDIT_TRUE='#'
    HAVE_LIBAUDIT_FALSE=
  fi

```

```

if test x$have_libaudit = xyes ; then
    SELINUX_LIBS="$SELINUX_LIBS -laudit -lcap-ng"

$as_echo "#define HAVE_LIBAUDIT 1" >>confdefs.h

fi

# Check for ADT API (Solaris Basic Security Mode auditing)
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for ADT API" >&5
$as_echo_n "checking for ADT API... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <bsm/adt.h>
adt_user_context = ADT_USER;

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    check_adt_audit=yes
else
    check_adt_audit=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

if test ${check_adt_audit} = yes
then

$as_echo "#define HAVE_ADT /**/" >>confdefs.h

    ADT_LIBS="-lbsm"
    LIBS="-lbsm $LIBS"
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

# Check for SCM_RIGHTS
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for SCM_RIGHTS" >&5

```

```

$as_echo_n "checking for SCM_RIGHTS... " >&6; }
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
static int x = SCM_RIGHTS;

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: supported" >&5
$as_echo "supported" >&6; }

$as_echo "#define HAVE_UNIX_FD_PASSING 1" >>confdefs.h

else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: not supported" >&5
$as_echo "not supported" >&6; }
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext

NETWORK_libs=
if test x$dbus_win = xyes ; then
    if test x$dbus_wince = xyes ; then
        NETWORK_libs="-lws2"
    else
        NETWORK_libs="-lws2_32"
    fi
fi
fi

# Check whether --with-valgrind was given.
if test "${with_valgrind+set}" = set; then :
    withval=$with_valgrind;
else
    with_valgrind=no
fi

if test x$with_valgrind != xno; then

pkg_failed=no

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for VALGRIND" >&5
$as_echo_n "checking for VALGRIND... " >&6; }

if test -n "$VALGRIND_CFLAGS"; then
  pkg_cv_VALGRIND_CFLAGS="$VALGRIND_CFLAGS"
elif test -n "$PKG_CONFIG"; then
  if test -n "$PKG_CONFIG" && \
    { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"valgrind >= 3.6\""; } >&5
    ($PKG_CONFIG --exists --print-errors "valgrind >= 3.6") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    pkg_cv_VALGRIND_CFLAGS=`$PKG_CONFIG --cflags "valgrind >= 3.6"
2>/dev/null`
  else
    pkg_failed=yes
  fi
  else
    pkg_failed=untried
  fi
if test -n "$VALGRIND_LIBS"; then
  pkg_cv_VALGRIND_LIBS="$VALGRIND_LIBS"
elif test -n "$PKG_CONFIG"; then
  if test -n "$PKG_CONFIG" && \
    { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG} --exists -
-print-errors \"valgrind >= 3.6\""; } >&5
    ($PKG_CONFIG --exists --print-errors "valgrind >= 3.6") 2>&5
    ac_status=$?
    $as_echo "$as_me:${as_lineno-$LINENO}: \${?} = $ac_status" >&5
    test $ac_status = 0; }; then
    pkg_cv_VALGRIND_LIBS=`$PKG_CONFIG --libs "valgrind >= 3.6"
2>/dev/null`
  else
    pkg_failed=yes
  fi
  else
    pkg_failed=untried
  fi
fi

if test $pkg_failed = yes; then

if $PKG_CONFIG --atleast-pkgconfig-version 0.20; then
  _pkg_short_errors_supported=yes
else
  _pkg_short_errors_supported=no
fi
  if test $_pkg_short_errors_supported = yes; then
    VALGRIND_PKG_ERRORS=`$PKG_CONFIG --short-errors --print-
errors "valgrind >= 3.6" 2>&1`

```



```

        else
            VALGRIND_PKG_ERRORS=`$PKG_CONFIG --print-errors "valgrind
>= 3.6" 2>&1`
            fi
            # Put the nasty error message in config.log where it belongs
            echo "$VALGRIND_PKG_ERRORS" >&5

            as_fn_error $? "Package requirements (valgrind >= 3.6) were not
met:

$VALGRIND_PKG_ERRORS

Consider adjusting the PKG_CONFIG_PATH environment variable if you
installed software in a non-standard prefix.

Alternatively, you may set the environment variables VALGRIND_CFLAGS
and VALGRIND_LIBS to avoid the need to call pkg-config.
See the pkg-config man page for more details.
" "$LINENO" 5
elif test $pkg_failed = untried; then
    { { $as_echo "$as_me:${as_lineno-$LINENO}: error: in `\$ac_pwd':"
>&5
$as_echo "$as_me: error: in `\$ac_pwd':" >&2;}
as_fn_error $? "The pkg-config script could not be found or is too
old. Make sure it
is in your PATH or set the PKG_CONFIG environment variable to the full
path to pkg-config.

Alternatively, you may set the environment variables VALGRIND_CFLAGS
and VALGRIND_LIBS to avoid the need to call pkg-config.
See the pkg-config man page for more details.

To get pkg-config, see <http://pkg-config.freedesktop.org/>.
See `config.log' for more details" "$LINENO" 5; }
else
    VALGRIND_CFLAGS=$pkg_cv_VALGRIND_CFLAGS
    VALGRIND_LIBS=$pkg_cv_VALGRIND_LIBS
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
    :
fi

$as_echo "#define WITH_VALGRIND 1" >>confdefs.h

fi

#### Set up final flags
LIBDBUS_LIBS="$THREAD_LIBS $NETWORK_libs"

### X11 detection
DBUS_X_LIBS=

```

```

DBUS_X_CFLAGS=

# Check whether --enable-x11-autolaunch was given.
if test "${enable_x11_autolaunch+set}" = set; then :
    enableval=$enable_x11_autolaunch;
else
    enable_x11_autolaunch=auto
fi

if test "x$dbus_win" = xyes; then
    if test "x$enable_x11_autolaunch" = xyes; then
        as_fn_error $? "X11 auto-launch is not supported on Windows"
"$LINENO" 5
    fi

    enable_x11_autolaunch=no
    have_x11=no
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for X" >&5
$as_echo_n "checking for X... " >&6; }

# Check whether --with-x was given.
if test "${with_x+set}" = set; then :
    withval=$with_x;
fi

# $have_x is `yes', `no', `disabled', or empty when we do not yet
know.
if test "x$with_x" = xno; then
    # The user explicitly disabled X.
    have_x=disabled
else
    case $x_includes,$x_libraries in #(
        *\'*) as_fn_error $? "cannot use X directory names containing '"
"$LINENO" 5;; #(
        *,NONE | NONE,*) if ${ac_cv_have_x+:} false; then :
    $as_echo_n "(cached) " >&6
else
    # One or both of the vars are not set, and there is no cached value.
ac_x_includes=no ac_x_libraries=no
# Standard set of common directories for X headers.
# Check X11 before X11Rn because it is often a symlink to the current
release.
ac_x_header_dirs=''

if test "$ac_x_includes" = no; then
    # Guess where to find include files, by looking for Xlib.h.
    # First, try using that file with no special directory specified.
    cat confdefs.h - <<_ACEOF >>confgtest.$ac_ext
/* end confdefs.h. */

```

```

#include <X11/Xlib.h>
_ACEOF
if ac_fn_c_try_cpp "$LINENO"; then :
    # We can compile using X headers with no special include directory.
ac_x_includes=
else
    for ac_dir in $ac_x_header_dirs; do
        if test -r "$ac_dir/X11/Xlib.h"; then
            ac_x_includes=$ac_dir
            break
        fi
    done
fi
rm -f conftest.err conftest.i conftest.$ac_ext
fi # $ac_x_includes = no

if test "$ac_x_libraries" = no; then
    # Check for the libraries.
    # See if we find them without any special options.
    # Don't add to $LIBS permanently.
    ac_save_LIBS=$LIBS
    LIBS="-lX11 $LIBS"
    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */
#include <X11/Xlib.h>
int
main ()
{
XrmInitialize ()
    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    LIBS=$ac_save_LIBS
    # We can link X programs with no special library path.
ac_x_libraries=
else
    LIBS=$ac_save_LIBS
    for ac_dir in `$as_echo "$ac_x_includes $ac_x_header_dirs" | sed
s/include/lib/g`
    do
        # Don't even attempt the hair of trying to link an X program!
        for ac_extension in a so sl dylib la dll; do
            if test -r "$ac_dir/libX11.$ac_extension"; then
                ac_x_libraries=$ac_dir
                break 2
            fi
        done
    done
done
fi
rm -f core conftest.err conftest.$ac_objext \

```

```

    conftest$ac_exeext conftest.$ac_ext
fi # $ac_x_libraries = no

case $ac_x_includes,$ac_x_libraries in #(
  no,* | *,no | *\')
    # Didn't find X, or a directory has "" in its name.
    ac_cv_have_x="have_x=no";; #(
*)
    # Record where we found X for the cache.
    ac_cv_have_x="have_x=yes\
    ac_x_includes='$ac_x_includes'\
    ac_x_libraries='$ac_x_libraries'"
esac
fi
;; #(
*) have_x=yes;;
esac
eval "$ac_cv_have_x"
fi # $with_x != no

if test "$have_x" != yes; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $have_x" >&5
$as_echo "$have_x" >&6; }
  no_x=yes
else
  # If each of the values was on the command line, it overrides each
  guess.
  test "x$x_includes" = xNONE && x_includes=$ac_x_includes
  test "x$x_libraries" = xNONE && x_libraries=$ac_x_libraries
  # Update the cache value to reflect the command line values.
  ac_cv_have_x="have_x=yes\
  ac_x_includes='$x_includes'\
  ac_x_libraries='$x_libraries'"
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: libraries
$x_libraries, headers $x_includes" >&5
$as_echo "libraries $x_libraries, headers $x_includes" >&6; }
fi

if test "$no_x" = yes; then
  # Not all programs may use this symbol, but it does not hurt to
  define it.

$as_echo "#define X_DISPLAY_MISSING 1" >>confdefs.h

  X_CFLAGS= X_PRE_LIBS= X_LIBS= X_EXTRA_LIBS=
else
  if test -n "$x_includes"; then
    X_CFLAGS="$X_CFLAGS -I$x_includes"
  fi

  # It would also be nice to do this for all -L options, not just this
  one.

```

```

if test -n "$x_libraries"; then
  X_LIBS="$X_LIBS -L$x_libraries"
  # For Solaris; some versions of Sun CC require a space after -R
and
  # others require no space.  Words are not sufficient . . . .
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether -R must
be followed by a space" >&5
$as_echo_n "checking whether -R must be followed by a space... " >&6;
}
  ac_xsave_LIBS=$LIBS; LIBS="$LIBS -R$x_libraries"
  ac_xsave_c_werror_flag=$ac_c_werror_flag
  ac_c_werror_flag=yes
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

```

```

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
  X_LIBS="$X_LIBS -R$x_libraries"
else
  LIBS="$ac_xsave_LIBS -R $x_libraries"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

```

```

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: yes" >&5
$as_echo "yes" >&6; }
  X_LIBS="$X_LIBS -R $x_libraries"
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: neither works" >&5
$as_echo "neither works" >&6; }
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
fi
rm -f core conftest.err conftest.$ac_objext \

```

```

    conftest$sac_exeext conftest.$sac_ext
    ac_c_werror_flag=$sac_xsave_c_werror_flag
    LIBS=$sac_xsave_LIBS
fi

# Check for system-dependent libraries X programs must link with.
# Do this before checking for the system-independent R6 libraries
# (-lICE), since we may need -lsocket or whatever for X linking.

if test "$ISC" = yes; then
    X_EXTRA_LIBS="$X_EXTRA_LIBS -lnsl_s -linet"
else
    # Martyn Johnson says this is needed for Ultrix, if the X
    # libraries were built with DECnet support.  And Karl Berry says
    # the Alpha needs dnet_stub (dnet does not exist).
    ac_xsave_LIBS="$LIBS"; LIBS="$LIBS $X_LIBS -lX11"
    cat confdefs.h - <<_ACEOF >conftest.$sac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char XOpenDisplay ();
int
main ()
{
return XOpenDisplay ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :

else
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for dnet_ntoa in -
ldnet" >&5
$as_echo_n "checking for dnet_ntoa in -ldnet... " >&6; }
if ${ac_cv_lib_dnet_dnet_ntoa+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-ldnet $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$sac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
   Use char because int might match the return type of a GCC
   builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus

```

```

extern "C"
#endif
char dnet_ntoa ();
int
main ()
{
return dnet_ntoa ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_dnet_dnet_ntoa=yes
else
ac_cv_lib_dnet_dnet_ntoa=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dnet_dnet_ntoa" >&5
$as_echo "$ac_cv_lib_dnet_dnet_ntoa" >&6; }
if test "x$ac_cv_lib_dnet_dnet_ntoa" = xyes; then :
X_EXTRA_LIBS="$X_EXTRA_LIBS -ldnet"
fi

if test $ac_cv_lib_dnet_dnet_ntoa = no; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for dnet_ntoa
in -ldnet_stub" >&5
$as_echo_n "checking for dnet_ntoa in -ldnet_stub... " >&6; }
if ${ac_cv_lib_dnet_stub_dnet_ntoa+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-ldnet_stub $LIBS"
cat confdefs.h - << _ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"
#endif
char dnet_ntoa ();
int
main ()
{
return dnet_ntoa ();
;
return 0;

```

```

}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
  ac_cv_lib_dnet_stub_dnet_ntoa=yes
else
  ac_cv_lib_dnet_stub_dnet_ntoa=no
fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_dnet_stub_dnet_ntoa" >&5
$as_echo "$ac_cv_lib_dnet_stub_dnet_ntoa" >&6; }
if test "x$ac_cv_lib_dnet_stub_dnet_ntoa" = xyes; then :
  X_EXTRA_LIBS="$X_EXTRA_LIBS -ldnet_stub"
fi

  fi

fi
rm -f core conftest.err conftest.$ac_objext \
  conftest$ac_exeext conftest.$ac_ext
LIBS="$ac_xsave_LIBS"

# msh@cis.ufl.edu says -lnsl (and -lsocket) are needed for his
386/AT,
# to get the SysV transport functions.
# Chad R. Larson says the Pyramis MIS-ES running DC/OSx (SVR4)
# needs -lnsl.
# The nsl library prevents programs from opening the X display
# on Irix 5.2, according to T.E. Dickey.
# The functions gethostbyname, getservbyname, and inet_addr are
# in -lbsd on LynxOS 3.0.1/i386, according to Lars Hecking.
ac_fn_c_check_func "$LINENO" "gethostbyname"
"ac_cv_func_gethostbyname"
if test "x$ac_cv_func_gethostbyname" = xyes; then :

fi

  if test $ac_cv_func_gethostbyname = no; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: checking for
gethostbyname in -lnsl" >&5
$as_echo_n "checking for gethostbyname in -lnsl... " >&6; }
if ${ac_cv_lib_nsl_gethostbyname+:} false; then :
  $as_echo_n "(cached) " >&6
else
  ac_check_lib_save_LIBS=$LIBS
LIBS="-lnsl $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.

```



```

    Use char because int might match the return type of a GCC
    builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char gethostbyname ();
int
main ()
{
return gethostbyname ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_nsl_gethostbyname=yes
else
ac_cv_lib_nsl_gethostbyname=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_nsl_gethostbyname" >&5
$as_echo "$ac_cv_lib_nsl_gethostbyname" >&6; }
if test "x$ac_cv_lib_nsl_gethostbyname" = xyes; then :
X_EXTRA_LIBS="$X_EXTRA_LIBS -lnsl"
fi

if test $ac_cv_lib_nsl_gethostbyname = no; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for
gethostbyname in -lbsd" >&5
$as_echo_n "checking for gethostbyname in -lbsd... " >&6; }
if ${ac_cv_lib_bsd_gethostbyname+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-lbsd $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply.  */
#ifdef __cplusplus
extern "C"
#endif
char gethostbyname ();
int
main ()
{

```

```

return gethostbyname ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
    ac_cv_lib_bsd_gethostbyname=yes
else
    ac_cv_lib_bsd_gethostbyname=no
fi
rm -f core conftest.err conftest.$ac_objext \
    conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_bsd_gethostbyname" >&5
$as_echo "$ac_cv_lib_bsd_gethostbyname" >&6; }
if test "x$ac_cv_lib_bsd_gethostbyname" = xyes; then :
    X_EXTRA_LIBS="$X_EXTRA_LIBS -lbsd"
fi

    fi
fi

# lieder@skyler.mavd.honeywell.com says without -lsocket,
# socket/setsockopt and other routines are undefined under SCO ODT
# 2.0. But -lsocket is broken on IRIX 5.2 (and is not necessary
# on later versions), says Simon Leinen: it contains gethostby*
# variants that don't use the name server (or something). -
lsocket
# must be given before -lnsl if both are needed. We assume that
# if connect needs -lnsl, so does gethostbyname.
ac_fn_c_check_func "$LINENO" "connect" "ac_cv_func_connect"
if test "x$ac_cv_func_connect" = xyes; then :

fi

    if test $ac_cv_func_connect = no; then
        { $as_echo "$as_me:${as_lineno-$LINENO}: checking for connect in
-lsocket" >&5
$as_echo_n "checking for connect in -lsocket... " >&6; }
if ${ac_cv_lib_socket_connect+:} false; then :
    $as_echo_n "(cached) " >&6
else
    ac_check_lib_save_LIBS=$LIBS
LIBS="-lsocket $X_EXTRA_LIBS $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */

```

```

#ifdef __cplusplus
extern "C"
#endif
char connect ();
int
main ()
{
return connect ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_socket_connect=yes
else
ac_cv_lib_socket_connect=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_socket_connect" >&5
$as_echo "$ac_cv_lib_socket_connect" >&6; }
if test "x$ac_cv_lib_socket_connect" = xyes; then :
X_EXTRA_LIBS="-lsocket $X_EXTRA_LIBS"
fi

fi

# Guillermo Gomez says -lposix is necessary on A/UX.
ac_fn_c_check_func "$LINENO" "remove" "ac_cv_func_remove"
if test "x$ac_cv_func_remove" = xyes; then :

fi

if test $ac_cv_func_remove = no; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for remove in
-lposix" >&5
$as_echo_n "checking for remove in -lposix... " >&6; }
if ${ac_cv_lib_posix_remove+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-lposix $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */
#endif __cplusplus

```

```

extern "C"
#endif
char remove ();
int
main ()
{
return remove ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_posix_remove=yes
else
ac_cv_lib_posix_remove=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_posix_remove" >&5
$as_echo "$ac_cv_lib_posix_remove" >&6; }
if test "x$ac_cv_lib_posix_remove" = xyes; then :
X_EXTRA_LIBS="$X_EXTRA_LIBS -lposix"
fi

fi

# BSDI BSD/OS 2.1 needs -lipc for XOpenDisplay.
ac_fn_c_check_func "$LINENO" "shmat" "ac_cv_func_shmat"
if test "x$ac_cv_func_shmat" = xyes; then :

fi

if test $ac_cv_func_shmat = no; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for shmat in -
lipc" >&5
$as_echo_n "checking for shmat in -lipc... " >&6; }
if ${ac_cv_lib_ipc_shmat+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-lipc $LIBS"
cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */
#ifdef __cplusplus
extern "C"

```

```

#endif
char shmat ();
int
main ()
{
return shmat ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_ipc_shmat=yes
else
ac_cv_lib_ipc_shmat=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_cv_lib_ipc_shmat"
>&5
$as_echo "$ac_cv_lib_ipc_shmat" >&6; }
if test "x$ac_cv_lib_ipc_shmat" = xyes; then :
X_EXTRA_LIBS="$X_EXTRA_LIBS -lipc"
fi

fi
fi

# Check for libraries that X11R6 Xt/Xaw programs need.
ac_save_LDFLAGS=$LDFLAGS
test -n "$x_libraries" && LDFLAGS="$LDFLAGS -L$x_libraries"
# SM needs ICE to (dynamically) link under SunOS 4.x (so we have to
# check for ICE first), but we must link in the order -lSM -lICE or
# we get undefined symbols. So assume we have SM if we have ICE.
# These have to be linked with before -lX11, unlike the other
# libraries we check for below, so use a different variable.
# John Interrante, Karl Berry
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for
IceConnectionNumber in -lICE" >&5
$as_echo_n "checking for IceConnectionNumber in -lICE... " >&6; }
if ${ac_cv_lib_ICE_IceConnectionNumber+:} false; then :
$as_echo_n "(cached) " >&6
else
ac_check_lib_save_LIBS=$LIBS
LIBS="-lICE $X_EXTRA_LIBS $LIBS"
cat confdefs.h - <<_ACEOF >>conftest.$ac_ext
/* end confdefs.h. */

/* Override any GCC internal prototype to avoid an error.
Use char because int might match the return type of a GCC
builtin and then its argument prototype would still apply. */

```

```

#ifdef __cplusplus
extern "C"
#endif
char IceConnectionNumber ();
int
main ()
{
return IceConnectionNumber ();
;
return 0;
}
_ACEOF
if ac_fn_c_try_link "$LINENO"; then :
ac_cv_lib_ICE_IceConnectionNumber=yes
else
ac_cv_lib_ICE_IceConnectionNumber=no
fi
rm -f core conftest.err conftest.$ac_objext \
conftest$ac_exeext conftest.$ac_ext
LIBS=$ac_check_lib_save_LIBS
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result:
$ac_cv_lib_ICE_IceConnectionNumber" >&5
$as_echo "$ac_cv_lib_ICE_IceConnectionNumber" >&6; }
if test "x$ac_cv_lib_ICE_IceConnectionNumber" = xyes; then :
X_PRE_LIBS="$X_PRE_LIBS -lSM -lICE"
fi

LDFLAGS=$ac_save_LDFLAGS

fi

if test "x$no_x" = xyes; then
have_x11=no
else
have_x11=yes
DBUS_X_LIBS="$X_LIBS $X_PRE_LIBS -lX11 $X_EXTRA_LIBS"
DBUS_X_CFLAGS="$X_CFLAGS"
fi
fi

if test "x$enable_x11_autolaunch,$have_x11" = xyes,no; then
as_fn_error $? "X11 auto-launch requires X headers/libraries"
"$LINENO" 5
else
# move from "auto" to "yes" or "no" if necessary
if test "x$enable_x11_autolaunch" != xno; then
enable_x11_autolaunch="$have_x11"
fi
fi

```

```

if test "x$have_x11" = xyes ; then

$as_echo "#define DBUS_BUILD_X11 1" >>confdefs.h

fi

if test "x$enable_x11_autolaunch" = xyes ; then

$as_echo "#define DBUS_ENABLE_X11_AUTOLAUNCH 1" >>confdefs.h

fi

#### gcc warning flags

cc_supports_flag() {
  { $as_echo "$as_me:${as_lineno-$LINENO}: checking whether $CC
supports \"\$*\\" >&5
$as_echo_n "checking whether $CC supports \"\$*\\"... " >&6; }
  save_CFLAGS="$CFLAGS"
  CFLAGS="\$*"
  cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
  rc=yes
else
  rc=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $rc" >&5
$as_echo "$rc" >&6; }
  test "x$rc" = xyes
}

  { $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands " >&5
$as_echo_n "checking to see if compiler understands ... " >&6; }

```

```

save_CFLAGS="$CFLAGS"
save_CXXFLAGS="$CXXFLAGS"
CFLAGS="$CFLAGS "
CXXFLAGS="$CXXFLAGS "

cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then

    true
else

    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

tp_warnings=""
for tp_flag in all \
extra \
char-subscripts \
missing-declarations \
missing-prototypes \
nested-externs \
pointer-arith \
cast-align \
no-address \
float-equal \
declaration-after-statement \
; do

```



```

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -W$tp_flag" >&5
$as_echo_n "checking to see if compiler understands -W$tp_flag... "
>&6; }

    save_CFLAGS="$CFLAGS"
    save_CXXFLAGS="$CXXFLAGS"
    CFLAGS="$CFLAGS -W$tp_flag"
    CXXFLAGS="$CXXFLAGS -W$tp_flag"

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    tp_warnings="$tp_warnings -W$tp_flag"
    true
else
    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

done

tp_error_flags="-Werror"

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -Werror" >&5
$as_echo_n "checking to see if compiler understands -Werror... " >&6;
}

    save_CFLAGS="$CFLAGS"
    save_CXXFLAGS="$CXXFLAGS"
    CFLAGS="$CFLAGS -Werror"

```

```

CXXFLAGS="$CXXFLAGS -Werror"

cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    tp_werror=yes
    true
else
    tp_werror=no
    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

for tp_flag in
missing-field-initializers \
unused-parameter \
sign-compare \
pointer-sign \
type-limits \
; do
                                $DISABLE_UNUSED_WARNINGS \

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -Wno-$tp_flag" >&5
$as_echo_n "checking to see if compiler understands -Wno-$tp_flag... "
>&6; }

save_CFLAGS="$CFLAGS"
save_CXXFLAGS="$CXXFLAGS"
CFLAGS="$CFLAGS -Wno-$tp_flag"
CXXFLAGS="$CXXFLAGS -Wno-$tp_flag"

cat confdefs.h - <<_ACEOF >conftest.$ac_ext

```

```

/* end confdefs.h.  */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    tp_warnings="$tp_warnings -Wno-$tp_flag"
    true
else

    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -Wno-error=$tp_flag" >&5
$as_echo_n "checking to see if compiler understands -Wno-
error=$tp_flag... " >&6; }

save_CFLAGS="$CFLAGS"
save_CXXFLAGS="$CXXFLAGS"
CFLAGS="$CFLAGS -Wno-error=$tp_flag"
CXXFLAGS="$CXXFLAGS -Wno-error=$tp_flag"

cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :

```

```

    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    tp_error_flags="$tp_error_flags -Wno-error=$tp_flag"
    true
else
    tp_werror=no
    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

done

# Check whether --enable-Werror was given.
if test "${enable_Werror+set}" = set; then :
    enableval=$enable_Werror; tp_werror=$enableval
else
    :
fi

if test "x$tp_werror" = xyes &&      test x$dbus_win != xyes -a
x$dbus_cygwin != xyes -a x$enable_developer = xyes; then
    WARNING_CFLAGS="$tp_error_flags $tp_warnings"
else
    WARNING_CFLAGS="$tp_warnings"
fi

if test "x$GCC" = "xyes"; then
    # We're treating -fno-common like a warning: it makes the linker
more
    # strict, because on some systems the linker is *always* this strict

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -fno-common" >&5
$as_echo_n "checking to see if compiler understands -fno-common... "
>&6; }

    save_CFLAGS="$CFLAGS"
    save_CXXFLAGS="$CXXFLAGS"
    CFLAGS="$CFLAGS -fno-common"
    CXXFLAGS="$CXXFLAGS -fno-common"

```

```

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    WARNING_CFLAGS="$WARNING_CFLAGS -fno-common"
    true
else

    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

# http://bugs.freedesktop.org/show_bug.cgi?id=10599

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -fno-strict-aliasing" >&5
$as_echo_n "checking to see if compiler understands -fno-strict-
aliasing... " >&6; }

save_CFLAGS="$CFLAGS"
save_CXXFLAGS="$CXXFLAGS"
CFLAGS="$CFLAGS -fno-strict-aliasing"
CXXFLAGS="$CXXFLAGS -fno-strict-aliasing"

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

```

```

    ;
    return 0;
}
_ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    WARNING_CFLAGS="$WARNING_CFLAGS -fno-strict-aliasing"
    true
else

    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

if test "x$enable_ansi" = "xyes"; then

    { $as_echo "$as_me:${as_lineno-$LINENO}: checking to see if compiler
understands -ansi -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -pedantic"
>&5
$as_echo_n "checking to see if compiler understands -ansi -
D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -pedantic... " >&6; }

    save_CFLAGS="$CFLAGS"
    save_CXXFLAGS="$CXXFLAGS"
    CFLAGS="$CFLAGS -ansi -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -
pedantic"
    CXXFLAGS="$CXXFLAGS -ansi -D_POSIX_C_SOURCE=199309L -D_BSD_SOURCE -
pedantic"

    cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h. */

int
main ()
{

    ;
    return 0;
}

```

```

ACEOF
if ac_fn_c_try_compile "$LINENO"; then :
    flag_ok=yes
else
    flag_ok=no
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
CFLAGS="$save_CFLAGS"
CXXFLAGS="$save_CXXFLAGS"

if test "X$flag_ok" = Xyes ; then
    WARNING_CFLAGS="$WARNING_CFLAGS -ansi -D_POSIX_C_SOURCE=199309L -
D_BSD_SOURCE -pedantic"
    true
else

    true
fi
{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $flag_ok" >&5
$as_echo "$flag_ok" >&6; }

fi
fi

CFLAGS="$WARNING_CFLAGS $CFLAGS"

case $host_os in
    solaris*)
        # Solaris' C library apparently needs these runes to be
        threadsafe...
        CFLAGS="$CFLAGS -D_POSIX_PTHREAD_SEMANTICS -D_REENTRANT"
        # ... this opt-in to get sockaddr_in6 and sockaddr_storage...
        CFLAGS="$CFLAGS -D__EXTENSIONS__"
        # ... and this opt-in to get file descriptor passing support
        CFLAGS="$CFLAGS -D_XOPEN_SOURCE=500"
        ;;
esac

### Doxygen Documentation

# Extract the first word of "doxygen", so it can be a program name
with args.
set dummy doxygen; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_DOXYGEN+:} false; then :
    $as_echo_n "(cached) " >&6
else
    case $DOXYGEN in
        [\\/] * | ? : [\\/] * )

```

```

    ac_cv_path_DOXYGEN="$DOXYGEN" # Let the user override the test with
a path.
    ;;
*)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '' $ac_executable_extensions; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_path_DOXYGEN="$as_dir/$ac_word$ac_exec_ext"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
    done
IFS=$as_save_IFS

    test -z "$ac_cv_path_DOXYGEN" && ac_cv_path_DOXYGEN="no"
    ;;
esac
fi
DOXYGEN=$ac_cv_path_DOXYGEN
if test -n "$DOXYGEN"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $DOXYGEN" >&5
$as_echo "$DOXYGEN" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build
Doxygen documentation" >&5
$as_echo_n "checking whether to build Doxygen documentation... " >&6;
}

if test x$DOXYGEN = xno ; then
    have_doxygen=no
else
    have_doxygen=yes
fi

if test x$enable_doxygen_docs = xauto ; then
    if test x$have_doxygen = xno ; then
        enable_doxygen_docs=no
    else
        enable_doxygen_docs=yes
    fi
fi

```



```

fi

if test x$enable_doxygen_docs = xyes; then
    if test x$have_doxygen = xno; then
        as_fn_error $? "Building Doxygen docs explicitly required, but
Doxygen not found" "$LINENO" 5
    fi
fi

    if test x$enable_doxygen_docs = xyes; then
        DBUS_DOXYGEN_DOCS_ENABLED_TRUE=
        DBUS_DOXYGEN_DOCS_ENABLED_FALSE='#'
    else
        DBUS_DOXYGEN_DOCS_ENABLED_TRUE='#'
        DBUS_DOXYGEN_DOCS_ENABLED_FALSE=
    fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_doxygen_docs"
>&5
$as_echo "$enable_doxygen_docs" >&6; }

for ac_prog in xsltproc
do
    # Extract the first word of "$ac_prog", so it can be a program name
with args.
set dummy $ac_prog; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_prog_XSLTPROC+:} false; then :
    $as_echo_n "(cached) " >&6
else
    if test -n "$XSLTPROC"; then
        ac_cv_prog_XSLTPROC="$XSLTPROC" # Let the user override the test.
    else
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
    IFS=$as_save_IFS
    test -z "$as_dir" && as_dir=.
    for ac_exec_ext in '$ac_executable_extensions'; do
    if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
        ac_cv_prog_XSLTPROC="$ac_prog"
        $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
        break 2
    fi
done
    done
IFS=$as_save_IFS

fi
fi

```

```

XSLTPROC=$ac_cv_prog_XSLTPROC
if test -n "$XSLTPROC"; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: $XSLTPROC" >&5
$as_echo "$XSLTPROC" >&6; }
else
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

  test -n "$XSLTPROC" && break
done

if test "x$XSLTPROC" != "x"; then
  DBUS_HAVE_XSLTPROC_TRUE=
  DBUS_HAVE_XSLTPROC_FALSE='#'
else
  DBUS_HAVE_XSLTPROC_TRUE='#'
  DBUS_HAVE_XSLTPROC_FALSE=
fi

### XML Documentation

# Extract the first word of "xmlto", so it can be a program name with
args.
set dummy xmlto; ac_word=$2
{ $as_echo "$as_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$as_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_XMLTO+:} false; then :
  $as_echo_n "(cached) " >&6
else
  case $XMLTO in
  [\\/*]* | ?:[\\/*]*)
    ac_cv_path_XMLTO="$XMLTO" # Let the user override the test with a
path.
    ;;
  *)
    as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_XMLTO="$as_dir/$ac_word$ac_exec_ext"
    $as_echo "$as_me:${as_lineno-$LINENO}: found
$as_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
done
done

```

```

IFS=$as_save_IFS

    test -z "$ac_cv_path_XMLTO" && ac_cv_path_XMLTO="no"
    ;;
esac
fi
XMLTO=$ac_cv_path_XMLTO
if test -n "$XMLTO"; then
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: $XMLTO" >&5
$as_echo "$XMLTO" >&6; }
else
    { $as_echo "$as_me:${as_lineno-$LINENO}: result: no" >&5
$as_echo "no" >&6; }
fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking whether to build XML
documentation" >&5
$as_echo_n "checking whether to build XML documentation... " >&6; }

if test x$XMLTO = xno ; then
    have_xmlto=no
else
    have_xmlto=yes
fi

if test x$enable_xml_docs = xauto ; then
    if test x$have_xmlto = xno ; then
        enable_xml_docs=no
    else
        enable_xml_docs=yes
    fi
fi

if test x$enable_xml_docs = xyes; then
    if test x$have_xmlto = xno; then
        as_fn_error $? "Building XML docs explicitly required, but xmlto
not found" "$LINENO" 5
    fi
fi

    if test x$enable_xml_docs = xyes; then
        DBUS_XML_DOCS_ENABLED_TRUE=
        DBUS_XML_DOCS_ENABLED_FALSE='#'
    else
        DBUS_XML_DOCS_ENABLED_TRUE='#'
        DBUS_XML_DOCS_ENABLED_FALSE=
    fi

{ $as_echo "$as_me:${as_lineno-$LINENO}: result: $enable_xml_docs" >&5
$as_echo "$enable_xml_docs" >&6; }

```

```

# Extract the first word of "man2html", so it can be a program name
with args.
set dummy man2html; ac_word=$2
{ $sas_echo "$sas_me:${as_lineno-$LINENO}: checking for $ac_word" >&5
$sas_echo_n "checking for $ac_word... " >&6; }
if ${ac_cv_path_MAN2HTML+:} false; then :
  $sas_echo_n "(cached) " >&6
else
  case $MAN2HTML in
    [\\/* | ?:[\\/*]*)
      ac_cv_path_MAN2HTML="$MAN2HTML" # Let the user override the test
with a path.
      ;;
    *)
      as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  for ac_exec_ext in ' $ac_executable_extensions; do
  if as_fn_executable_p "$as_dir/$ac_word$ac_exec_ext"; then
    ac_cv_path_MAN2HTML="$as_dir/$ac_word$ac_exec_ext"
    $sas_echo "$sas_me:${as_lineno-$LINENO}: found
$sas_dir/$ac_word$ac_exec_ext" >&5
    break 2
  fi
done
  done
IFS=$as_save_IFS

  ;;
esac
fi
MAN2HTML=$ac_cv_path_MAN2HTML
if test -n "$MAN2HTML"; then
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: $MAN2HTML" >&5
$sas_echo "$MAN2HTML" >&6; }
else
  { $sas_echo "$sas_me:${as_lineno-$LINENO}: result: no" >&5
$sas_echo "no" >&6; }
fi

if test x$MAN2HTML != x; then
  DBUS_HAVE_MAN2HTML_TRUE=
  DBUS_HAVE_MAN2HTML_FALSE='#'
else
  DBUS_HAVE_MAN2HTML_TRUE='#'
  DBUS_HAVE_MAN2HTML_FALSE=
fi

```

```

if test x$enable_doxygen_docs = xyes -a x$enable_xml_docs = xyes -a \
    x$MAN2HTML != x; then
    DBUS_CAN_UPLOAD_DOCS_TRUE=
    DBUS_CAN_UPLOAD_DOCS_FALSE='#'
else
    DBUS_CAN_UPLOAD_DOCS_TRUE='#'
    DBUS_CAN_UPLOAD_DOCS_FALSE=
fi

```

```

#### Have to go $localstatedir->$prefix/var->/usr/local/var

```

```

#### find the actual value for $prefix that we'll end up with
## (I know this is broken and should be done in the Makefile, but
## that's a major pain and almost nobody actually seems to care)

```

```

EXP_VAR=EXPANDED_PREFIX
FROM_VAR="$prefix"

```

```

    prefix_save=$prefix
    exec_prefix_save=$exec_prefix

```

```

    if test "x$prefix" = "xNONE"; then
        prefix="$ac_default_prefix"
    fi
    if test "x$exec_prefix" = "xNONE"; then
        exec_prefix=$prefix
    fi

```

```

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

```

```

    full_var=$new_full_var
    EXPANDED_PREFIX="$full_var"

```

```

    prefix=$prefix_save
    exec_prefix=$exec_prefix_save

```

```

EXP_VAR=EXPANDED_LOCALSTATEDIR
FROM_VAR="$localstatedir"

```

```

    prefix_save=$prefix
    exec_prefix_save=$exec_prefix

```

```

    if test "x$prefix" = "xNONE"; then
    prefix="$ac_default_prefix"
fi
    if test "x$exec_prefix" = "xNONE"; then
    exec_prefix=$prefix
fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

    full_var=$new_full_var
EXPANDED_LOCALSTATEDIR="$full_var"

    prefix=$prefix_save
exec_prefix=$exec_prefix_save

EXP_VAR=EXPANDED_SYSCONFDIR
FROM_VAR="$sysconfdir"

    prefix_save=$prefix
exec_prefix_save=$exec_prefix

    if test "x$prefix" = "xNONE"; then
    prefix="$ac_default_prefix"
fi
    if test "x$exec_prefix" = "xNONE"; then
    exec_prefix=$prefix
fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

    full_var=$new_full_var
EXPANDED_SYSCONFDIR="$full_var"

    prefix=$prefix_save
exec_prefix=$exec_prefix_save

EXP_VAR=EXPANDED_BINDIR
FROM_VAR="$bindir"

```

```

    prefix_save=$prefix
exec_prefix_save=$exec_prefix

    if test "x$prefix" = "xNONE"; then
    prefix="$ac_default_prefix"
fi
    if test "x$exec_prefix" = "xNONE"; then
    exec_prefix=$prefix
fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

    full_var=$new_full_var
EXPANDED_BINDIR="$full_var"

    prefix=$prefix_save
exec_prefix=$exec_prefix_save

EXP_VAR=EXPANDED_LIBDIR
FROM_VAR="$libdir"

    prefix_save=$prefix
exec_prefix_save=$exec_prefix

    if test "x$prefix" = "xNONE"; then
    prefix="$ac_default_prefix"
fi
    if test "x$exec_prefix" = "xNONE"; then
    exec_prefix=$prefix
fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

    full_var=$new_full_var
EXPANDED_LIBDIR="$full_var"

    prefix=$prefix_save
exec_prefix=$exec_prefix_save

```

```

EXP_VAR=EXPANDED_LIBEXECDIR
FROM_VAR="$libexecdir"

    prefix_save=$prefix
exec_prefix_save=$exec_prefix

    if test "x$prefix" = "xNONE"; then
        prefix="$ac_default_prefix"
    fi
    if test "x$exec_prefix" = "xNONE"; then
        exec_prefix=$prefix
    fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

full_var=$new_full_var
EXPANDED_LIBEXECDIR="$full_var"

    prefix=$prefix_save
exec_prefix=$exec_prefix_save

EXP_VAR=EXPANDED_DATADIR
FROM_VAR="$datadir"

    prefix_save=$prefix
exec_prefix_save=$exec_prefix

    if test "x$prefix" = "xNONE"; then
        prefix="$ac_default_prefix"
    fi
    if test "x$exec_prefix" = "xNONE"; then
        exec_prefix=$prefix
    fi

full_var="$FROM_VAR"
while true; do
    new_full_var=`eval echo $full_var`
    if test "x$new_full_var" = "x$full_var"; then break; fi
    full_var=$new_full_var
done

full_var=$new_full_var
EXPANDED_DATADIR="$full_var"

```



```

    prefix=$prefix_save
    exec_prefix=$exec_prefix_save

#### Check our operating system
operating_system=unknown
if test -f /etc/redhat-release || test -f $EXPANDED_SYSCONFDIR/redhat-
release ; then
    operating_system=redhat
fi

if test -f /etc/slackware-version || test -f
$EXPANDED_SYSCONFDIR/slackware-version ; then
    operating_system=slackware
fi

if test -f /usr/bin/cygwin1.dll || test -f
$EXPANDED_BINDIR/cygwin1.dll ; then
    operating_system=cygwin
fi

#### Sort out init scripts

if test x$with_init_scripts = x; then
    case x$operating_system in
        xredhat) with_init_scripts=redhat ;;
        xslackware) with_init_scripts=slackware ;;
        xcygwin) with_init_scripts=cygwin ;;
        *) with_init_scripts=none ;;
    esac
fi

if test x$with_init_scripts = xredhat; then
    DBUS_INIT_SCRIPTS_RED_HAT_TRUE=
    DBUS_INIT_SCRIPTS_RED_HAT_FALSE='#'
else
    DBUS_INIT_SCRIPTS_RED_HAT_TRUE='#'
    DBUS_INIT_SCRIPTS_RED_HAT_FALSE=
fi

if test x$with_init_scripts = xslackware; then
    DBUS_INIT_SCRIPTS_SLACKWARE_TRUE=
    DBUS_INIT_SCRIPTS_SLACKWARE_FALSE='#'
else
    DBUS_INIT_SCRIPTS_SLACKWARE_TRUE='#'
    DBUS_INIT_SCRIPTS_SLACKWARE_FALSE=
fi

if test x$with_init_scripts = xcygwin; then
    DBUS_INIT_SCRIPTS_CYGWIN_TRUE=

```

```

    DBUS_INIT_SCRIPTS_CYGWIN_FALSE='#'
else
    DBUS_INIT_SCRIPTS_CYGWIN_TRUE='#'
    DBUS_INIT_SCRIPTS_CYGWIN_FALSE=
fi

##### systemd unit files

# Check whether --with-systemdsystemunitdir was given.
if test "${with_systemdsystemunitdir+set}" = set; then :
    withval=$with_systemdsystemunitdir;
else

    if test -n "$PKG_CONFIG" && \
        { { $as_echo "$as_me:${as_lineno-$LINENO}: \${PKG_CONFIG --exists -
--print-errors \"systemd\""; } >&5
        ($PKG_CONFIG --exists --print-errors "systemd") 2>&5
        ac_status=$?
        $as_echo "$as_me:${as_lineno-$LINENO}: \${? = $ac_status" >&5
        test $ac_status = 0; }; then
        with_systemdsystemunitdir=${($PKG_CONFIG --
variable=systemdsystemunitdir systemd)
    else
        with_systemdsystemunitdir=no
    fi
fi

if test "x$with_systemdsystemunitdir" != xno; then
    systemdsystemunitdir=$with_systemdsystemunitdir

fi
    if test -n "$with_systemdsystemunitdir" -a
"x$with_systemdsystemunitdir" != xno ; then
        HAVE_SYSTEMD_TRUE=
        HAVE_SYSTEMD_FALSE='#'
    else
        HAVE_SYSTEMD_TRUE='#'
        HAVE_SYSTEMD_FALSE=
    fi
fi

##### Set up location for system bus socket
if ! test -z "$with_system_socket"; then
    DBUS_SYSTEM_SOCKET=$with_system_socket
else

DBUS_SYSTEM_SOCKET=${EXPANDED_LOCALSTATEDIR}/run/dbus/system_bus_socke
t
fi

```

```

cat >>confdefs.h <<_ACEOF
#define DBUS_SYSTEM_SOCKET "$DBUS_SYSTEM_SOCKET"
_ACEOF

## system bus only listens on local domain sockets, and never
## on an abstract socket (so only root can create the socket)
DBUS_SYSTEM_BUS_DEFAULT_ADDRESS="unix:path=$DBUS_SYSTEM_SOCKET"

cat >>confdefs.h <<_ACEOF
#define DBUS_SYSTEM_BUS_DEFAULT_ADDRESS
"$DBUS_SYSTEM_BUS_DEFAULT_ADDRESS"
_ACEOF

#### Set up the pid file
if ! test -z "$with_system_pid_file"; then
    DBUS_SYSTEM_PID_FILE=$with_system_pid_file
elif test x$with_init_scripts = xredhat ; then
    DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/messagebus.pid
else
    DBUS_SYSTEM_PID_FILE=${EXPANDED_LOCALSTATEDIR}/run/dbus/pid
fi

#### Directory to check for console ownership
if ! test -z "$with_console_auth_dir"; then
    DBUS_CONSOLE_AUTH_DIR=$with_console_auth_dir
else
    DBUS_CONSOLE_AUTH_DIR=/var/run/console/
fi

cat >>confdefs.h <<_ACEOF
#define DBUS_CONSOLE_AUTH_DIR "$DBUS_CONSOLE_AUTH_DIR"
_ACEOF

#### File to check for console ownership
if test x$have_console_owner_file = xyes; then
    if ! test -z "$with_console_owner_file"; then
        DBUS_CONSOLE_OWNER_FILE=$with_console_owner_file
    else
        DBUS_CONSOLE_OWNER_FILE=/dev/console
    fi
else
    DBUS_CONSOLE_OWNER_FILE=

```

```
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_CONSOLE_OWNER_FILE "$DBUS_CONSOLE_OWNER_FILE"
_ACEOF
```

```
#### User to start the system bus as
if test -z "$with_dbus_user" ; then
    DBUS_USER=messagebus
else
    DBUS_USER=$with_dbus_user
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_USER "$DBUS_USER"
_ACEOF
```

```
#### Prefix to install into
DBUS_PREFIX=$EXPANDED_PREFIX
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_PREFIX "$DBUS_PREFIX"
_ACEOF
```

```
#### Direcotry to install data files into
DBUS_DATADIR=$EXPANDED_DATADIR
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_DATADIR "$DBUS_DATADIR"
_ACEOF
```

```
#### Directory to install dbus-daemon
if test -z "$with_dbus_daemondir" ; then
    DBUS_DAEMONDIR=$EXPANDED_BINDIR
else
    DBUS_DAEMONDIR=$with_dbus_daemondir
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_DAEMONDIR "$DBUS_DAEMONDIR"
_ACEOF
```

```
#### Directory to install the other binaries
DBUS_BINDIR="$EXPANDED_BINDIR"
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_BINDIR "$DBUS_BINDIR"
_ACEOF
```

```
#### Directory to install the libexec binaries
DBUS_LIBEXECDIR="$EXPANDED_LIBEXECDIR"
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_LIBEXECDIR "$DBUS_LIBEXECDIR"
_ACEOF
```

```
#### Tell tests where to find certain stuff in builddir
```

```
DBUS_PWD=`pwd`
# Useful in a cross-compilation environment, where the tests are run
on the host system.
```

```
# Check whether --with-dbus-test-dir was given.
if test "${with_dbus_test_dir+set}" = set; then :
  withval=$with_dbus_test_dir; DBUS_PWD=$withval
fi
```

```
DBUS_TEST_EXEC="$DBUS_PWD/test"
DBUS_TEST_DATA="$DBUS_PWD/test/data"
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_TEST_EXEC "$DBUS_TEST_EXEC"
_ACEOF
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_EXEEXT "$EXEEXT"
_ACEOF
```

```
cat >>confdefs.h <<_ACEOF
#define TEST_BUS_BINARY "$DBUS_PWD/bus/dbus-daemon$EXEEXT"
_ACEOF
```

```
## Export the non-setuid external helper
TEST_LAUNCH_HELPER_BINARY="$DBUS_PWD/bus/dbus-daemon-launch-helper-
test$EXEEXT"
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_TEST_LAUNCH_HELPER_BINARY "$TEST_LAUNCH_HELPER_BINARY"
_ACEOF
```

```
DEFAULT_SOCKET_DIR=/tmp
```

```
DEFAULT_SOCKET_DIR=`echo $DEFAULT_SOCKET_DIR | sed 's/+/%/2B/g'`
```

```
if ! test -z "$with_test_socket_dir" ; then
    TEST_SOCKET_DIR="$with_test_socket_dir"
else
    TEST_SOCKET_DIR=$DEFAULT_SOCKET_DIR
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_TEST_SOCKET_DIR "$TEST_SOCKET_DIR"
_ACEOF
```

```
if test "x$dbus_unix" = xyes; then
    TEST_LISTEN="unix:tmpdir=$TEST_SOCKET_DIR"
else
    TEST_LISTEN="tcp:host=localhost"
fi
```

```
cat >>confdefs.h <<_ACEOF
#define TEST_LISTEN "$TEST_LISTEN"
_ACEOF
```

```
if ! test -z "$with_session_socket_dir" ; then
    DBUS_SESSION_SOCKET_DIR="$with_session_socket_dir"
else
    DBUS_SESSION_SOCKET_DIR=$DEFAULT_SOCKET_DIR
fi
```

```
cat >>confdefs.h <<_ACEOF
#define DBUS_SESSION_SOCKET_DIR "$DBUS_SESSION_SOCKET_DIR"
_ACEOF
```

```

if test x$dbus_win = xyes; then

DBUS_SESSION_BUS_DEFAULT_ADDRESS="$with_dbus_session_bus_default_address"
elif test x$have_launchd = xyes; then

DBUS_SESSION_BUS_DEFAULT_ADDRESS="launchd:env=DBUS_LAUNCHD_SESSION_BUS_SOCKET"
else

DBUS_SESSION_BUS_DEFAULT_ADDRESS="unix:tmpdir=$DBUS_SESSION_SOCKET_DIR"
fi

# darwin needs this to initialize the environment
for ac_header in crt_extrns.h
do :
    ac_fn_c_check_header_mongrel "$LINENO" "crt_extrns.h"
"ac_cv_header_crt_extrns_h" "$ac_includes_default"
if test "x$ac_cv_header_crt_extrns_h" = xyes; then :
    cat >>confdefs.h <<_ACEOF
#define HAVE_CRT_EXTRNS_H 1
_ACEOF
fi

done

ac_fn_c_check_func "$LINENO" "_NSGetEnviron"
"ac_cv_func__NSGetEnviron"
if test "x$ac_cv_func__NSGetEnviron" = xyes; then :

$as_echo "#define HAVE_NSGETENVIRON 1" >>confdefs.h

fi

# Check whether --enable-stats was given.
if test "${enable_stats+set}" = set; then :
    enableval=$enable_stats;
else
    enable_stats=no
fi

if test "x$enable_stats" = xyes; then

$as_echo "#define DBUS_ENABLE_STATS 1" >>confdefs.h

fi

```

```
ac_config_files="$ac_config_files Doxyfile dbus/versioninfo.rc
dbus/dbus-arch-deps.h bus/system.conf bus/session.conf bus/messagebus
bus/messagebus-config bus/org.freedesktop.dbus-session.plist
bus/rc.messagebus bus/dbus.service bus/dbus.socket Makefile
dbus/Makefile bus/Makefile tools/Makefile test/Makefile test/name-
test/Makefile doc/Makefile doc/dbus-daemon.1 dbus-1.pc dbus-1-
uninstalled.pc test/data/valid-config-files/debug-allow-all.conf
test/data/valid-config-files/debug-allow-all-sha1.conf
test/data/valid-config-files-system/debug-allow-all-pass.conf
test/data/valid-config-files-system/debug-allow-all-fail.conf
test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service
test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service test/data/invalid-
service-files-system/org.freedesktop.DBus.TestSuiteNoUser.service
test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service"
```

```
cat >confcache <<\_ACEOF
# This file is a shell script that caches the results of configure
# tests run on this system so they can be shared between configure
# scripts and configure runs, see configure's option --config-cache.
# It is not useful on other systems.  If it contains results you don't
# want to keep, you may remove or edit it.
#
# config.status only pays attention to the cache file if you give it
# the --recheck option to rerun configure.
#
# `ac_cv_env_foo' variables (set or unset) will be overridden when
# loading this file, other *unset* `ac_cv_foo' will be assigned the
# following values.
```

```
_ACEOF
```



```

# The following way of writing the cache mishandles newlines in
values,
# but we know of no workaround that is simple, portable, and
efficient.
# So, we kill variables containing newlines.
# Ultrix sh set writes to stderr and can't be redirected directly,
# and sets the high bit in the cache file unless we assign to the
vars.
(
  for ac_var in `(set) 2>&1 | sed -n 's/^\([a-zA-Z_][a-zA-Z0-
9_]*\)=.*\/\1/p'`; do
    eval ac_val=\${$ac_var}
    case $ac_val in #(
      *${as_nl}*)
        case $ac_var in #(
          *_cv_*) { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: cache
variable $ac_var contains a newline" >&5
$as_echo "$as_me: WARNING: cache variable $ac_var contains a newline"
>&2;} ;;
        esac
        case $ac_var in #(
          _ | IFS | as_nl) ;; #(
          BASH_ARGV | BASH_SOURCE) eval $ac_var= ;; #(
          *) { eval $ac_var=; unset $ac_var;} ;;
        esac ;;
      esac
    done

    (set) 2>&1 |
      case $as_nl `(ac_space=' '; set) 2>&1` in #(
        *${as_nl}ac_space=\ *)
          # `set' does not quote correctly, so add quotes: double-quote
          # substitution turns \\ \\ into \, and sed turns \ into \.
          sed -n \
            "s/'/'\\\''/g;

s/^\(\([_${as_cr_alnum}]*_cv_[_${as_cr_alnum}]*\)\)=\(.*\)\)/\1='\'2'/p"
          ;; #(
          *)
            # `set' quotes correctly as required by POSIX, so do not add
            quotes.
            sed -n "/^\([_${as_cr_alnum}]*_cv_[_${as_cr_alnum}]*\)=/p"
            ;;
          esac |
          sort
        ) |
        sed '
/^ac_cv_env_/b end
t clear
:clear
s/^\([^\=]*\)=\(.*\[{}].*\)$/test "${\1+set}" = set || &/

```

```

t end
s/^\([^=]*\)=\(.*\)$/\1=${\1=\2}/
:end' >>confcache
if diff "$cache_file" confcache >/dev/null 2>&1; then ;; else
  if test -w "$cache_file"; then
    if test "x$cache_file" != "x/dev/null"; then
      { $as_echo "$as_me:${as_lineno-$LINENO}: updating cache
$cache_file" >&5
$as_echo "$as_me: updating cache $cache_file" >&6;}
      if test ! -f "$cache_file" || test -h "$cache_file"; then
        cat confcache >"$cache_file"
      else
        case $cache_file in #(
          */* | ?:* )
            mv -f confcache "$cache_file"$$ &&
            mv -f "$cache_file"$$ "$cache_file" ;; #(
          *)
            mv -f confcache "$cache_file" ;;
        esac
      fi
    else
      { $as_echo "$as_me:${as_lineno-$LINENO}: not updating unwritable
cache $cache_file" >&5
$as_echo "$as_me: not updating unwritable cache $cache_file" >&6;}
    fi
  fi
fi
rm -f confcache

test "x$prefix" = xNONE && prefix=$ac_default_prefix
# Let make expand exec_prefix.
test "x$exec_prefix" = xNONE && exec_prefix='${prefix}'

DEFS=-DHAVE_CONFIG_H

ac_libobjs=
ac_ltlibobjs=
U=
for ac_i in : $LIBOBJS; do test "x$ac_i" = x: && continue
# 1. Remove the extension, and $U if already installed.
ac_script='s/\$U\././;s/\.o$//;s/\.obj$//'
ac_i=`$as_echo "$ac_i" | sed "$ac_script"`
# 2. Prepend LIBOBJDIR. When used with automake>=1.10 LIBOBJDIR
# will be set to the directory where LIBOBJS objects are built.
as_fn_append ac_libobjs " \${LIBOBJDIR}$ac_i\$U.$ac_objext"
as_fn_append ac_ltlibobjs " \${LIBOBJDIR}$ac_i"'\$U.lo'
done
LIBOBJS=$ac_libobjs

LTLIBOBJS=$ac_ltlibobjs

```

```

{ $as_echo "$as_me:${as_lineno-$LINENO}: checking that generated files
are newer than configure" >&5
$as_echo_n "checking that generated files are newer than configure...
" >&6; }
  if test -n "$am_sleep_pid"; then
    # Hide warnings about reused PIDs.
    wait $am_sleep_pid 2>/dev/null
  fi
  { $as_echo "$as_me:${as_lineno-$LINENO}: result: done" >&5
$as_echo "done" >&6; }
  if test -n "$EXEEXT"; then
    am_EXEEXT_TRUE=
    am_EXEEXT_FALSE='#'
  else
    am_EXEEXT_TRUE='#'
    am_EXEEXT_FALSE=
  fi

if test -z "${MAINTAINER_MODE_TRUE}" && test -z
"${MAINTAINER_MODE_FALSE}"; then
  as_fn_error $? "conditional \"MAINTAINER_MODE\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${AMDEP_TRUE}" && test -z "${AMDEP_FALSE}"; then
  as_fn_error $? "conditional \"AMDEP\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${am_fastdepCC_TRUE}" && test -z
"${am_fastdepCC_FALSE}"; then
  as_fn_error $? "conditional \"am_fastdepCC\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${am_fastdepCXX_TRUE}" && test -z
"${am_fastdepCXX_FALSE}"; then
  as_fn_error $? "conditional \"am_fastdepCXX\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_WIN_TRUE}" && test -z "${DBUS_WIN_FALSE}"; then
  as_fn_error $? "conditional \"DBUS_WIN\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_WINCE_TRUE}" && test -z "${DBUS_WINCE_FALSE}"; then
  as_fn_error $? "conditional \"DBUS_WINCE\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_UNIX_TRUE}" && test -z "${DBUS_UNIX_FALSE}"; then

```

```

    as_fn_error $? "conditional \"DBUS_UNIX\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_CYGWIN_TRUE}" && test -z "${DBUS_CYGWIN_FALSE}";
then
    as_fn_error $? "conditional \"DBUS_CYGWIN\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BUILD_TESTS_TRUE}" && test -z
"${DBUS_BUILD_TESTS_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_BUILD_TESTS\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_ENABLE_EMBEDDED_TESTS_TRUE}" && test -z
"${DBUS_ENABLE_EMBEDDED_TESTS_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_ENABLE_EMBEDDED_TESTS\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_ENABLE_MODULAR_TESTS_TRUE}" && test -z
"${DBUS_ENABLE_MODULAR_TESTS_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_ENABLE_MODULAR_TESTS\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_WITH_GLIB_TRUE}" && test -z
"${DBUS_WITH_GLIB_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_WITH_GLIB\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_ENABLE_INSTALLED_TESTS_TRUE}" && test -z
"${DBUS_ENABLE_INSTALLED_TESTS_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_ENABLE_INSTALLED_TESTS\" was
never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi

if test -z "${DBUS_USE_EXPAT_TRUE}" && test -z
"${DBUS_USE_EXPAT_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_USE_EXPAT\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_USE_LIBXML_TRUE}" && test -z
"${DBUS_USE_LIBXML_FALSE}"; then

```

```

    as_fn_error $? "conditional \"DBUS_USE_LIBXML\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${HAVE_SELINUX_TRUE}" && test -z "${HAVE_SELINUX_FALSE}";
then
    as_fn_error $? "conditional \"HAVE_SELINUX\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BUS_ENABLE_INOTIFY_TRUE}" && test -z
"${DBUS_BUS_ENABLE_INOTIFY_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_BUS_ENABLE_INOTIFY\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_TRUE}" && test -z
"${DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_BUS_ENABLE_DNOTIFY_ON_LINUX\" was
never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${HAVE_LINUX_EPOLL_TRUE}" && test -z
"${HAVE_LINUX_EPOLL_FALSE}"; then
    as_fn_error $? "conditional \"HAVE_LINUX_EPOLL\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_BUS_ENABLE_KQUEUE_TRUE}" && test -z
"${DBUS_BUS_ENABLE_KQUEUE_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_BUS_ENABLE_KQUEUE\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_ENABLE_LAUNCHD_TRUE}" && test -z
"${DBUS_ENABLE_LAUNCHD_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_ENABLE_LAUNCHD\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${HAVE_CONSOLE_OWNER_FILE_TRUE}" && test -z
"${HAVE_CONSOLE_OWNER_FILE_FALSE}"; then
    as_fn_error $? "conditional \"HAVE_CONSOLE_OWNER_FILE\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi

```

```
if test -z "${HAVE_LIBAUDIT_TRUE}" && test -z
"${HAVE_LIBAUDIT_FALSE}"; then
    as_fn_error $? "conditional \"HAVE_LIBAUDIT\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_DOXYGEN_DOCS_ENABLED_TRUE}" && test -z
"${DBUS_DOXYGEN_DOCS_ENABLED_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_DOXYGEN_DOCS_ENABLED\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_HAVE_XSLTPROC_TRUE}" && test -z
"${DBUS_HAVE_XSLTPROC_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_HAVE_XSLTPROC\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_XML_DOCS_ENABLED_TRUE}" && test -z
"${DBUS_XML_DOCS_ENABLED_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_XML_DOCS_ENABLED\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_HAVE_MAN2HTML_TRUE}" && test -z
"${DBUS_HAVE_MAN2HTML_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_HAVE_MAN2HTML\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_CAN_UPLOAD_DOCS_TRUE}" && test -z
"${DBUS_CAN_UPLOAD_DOCS_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_CAN_UPLOAD_DOCS\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_INIT_SCRIPTS_RED_HAT_TRUE}" && test -z
"${DBUS_INIT_SCRIPTS_RED_HAT_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_INIT_SCRIPTS_RED_HAT\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_INIT_SCRIPTS_SLACKWARE_TRUE}" && test -z
"${DBUS_INIT_SCRIPTS_SLACKWARE_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_INIT_SCRIPTS_SLACKWARE\" was
never defined.
```

```

Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${DBUS_INIT_SCRIPTS_CYGWIN_TRUE}" && test -z
"${DBUS_INIT_SCRIPTS_CYGWIN_FALSE}"; then
    as_fn_error $? "conditional \"DBUS_INIT_SCRIPTS_CYGWIN\" was never
defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi
if test -z "${HAVE_SYSTEMD_TRUE}" && test -z "${HAVE_SYSTEMD_FALSE}";
then
    as_fn_error $? "conditional \"HAVE_SYSTEMD\" was never defined.
Usually this means the macro was only invoked conditionally."
"$LINENO" 5
fi

: "${CONFIG_STATUS=./config.status}"
ac_write_fail=0
ac_clean_files_save=$ac_clean_files
ac_clean_files="$ac_clean_files $CONFIG_STATUS"
{ $as_echo "$as_me:${as_lineno-$LINENO}: creating $CONFIG_STATUS" >&5
$as_echo "$as_me: creating $CONFIG_STATUS" >&6;}
as_write_fail=0
cat >$CONFIG_STATUS <<_ASEOF || as_write_fail=1
#! $SHELL
# Generated by $as_me.
# Run this file to recreate the current configuration.
# Compiler output produced by configure, useful for debugging
# configure, is in config.log if it exists.

debug=false
ac_cs_recheck=false
ac_cs_silent=false

SHELL=${CONFIG_SHELL-$SHELL}
export SHELL
_ASEOF
cat >>$CONFIG_STATUS <<\_ASEOF || as_write_fail=1
## ----- ##
## M4sh Initialization. ##
## ----- ##

# Be more Bourne compatible
DUALCASE=1; export DUALCASE # for MKS sh
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
:
    emulate sh
    NULLCMD=:
    # Pre-4.2 versions of Zsh do word splitting on ${1+"$@"}, which
    # is contrary to our usage.  Disable this feature.
    alias -g '${1+"$@"}'='"$@"'

```



```

# The user is always right.
if test "${PATH_SEPARATOR+set}" != set; then
  PATH_SEPARATOR=:
  (PATH='/bin;/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 && {
    (PATH='/bin:/bin'; FPATH=$PATH; sh -c :) >/dev/null 2>&1 ||
      PATH_SEPARATOR=';'
  }
fi

# IFS
# We need space, tab and new line, in precisely that order. Quoting
is
# there to prevent editors from complaining about space-tab.
# (If _AS_PATH_WALK were called with IFS unset, it would disable word
# splitting by setting IFS to empty value.)
IFS=" " $as_nl

# Find who we are. Look in the path if we contain no directory
separator.
as_myself=
case $0 in #((
  *[\ \/]* ) as_myself=$0 ;;
  *) as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
  IFS=$as_save_IFS
  test -z "$as_dir" && as_dir=.
  test -r "$as_dir/$0" && as_myself=$as_dir/$0 && break
done
IFS=$as_save_IFS

  ;;
esac
# We did not find ourselves, most probably we were run as `sh COMMAND'
# in which case we are not to be found in the path.
if test "x$as_myself" = x; then
  as_myself=$0
fi
if test ! -f "$as_myself"; then
  $as_echo "$as_myself: error: cannot find myself; rerun with an
absolute file name" >&2
  exit 1
fi

# Unset variables that we do not need and which cause bugs (e.g. in
# pre-3.0 UWIN ksh). But do not cause bugs in bash 2.01; the "|| exit
1"
# suppresses any "Segmentation fault" message there. '(' could
# trigger a bug in pdksh 5.2.14.
for as_var in BASH_ENV ENV MAIL MAILPATH

```

```

do eval test x\${$as_var+set} = xset \
  && ( (unset $as_var) || exit 1) >/dev/null 2>&1 && unset $as_var ||
:
done
PS1='$ '
PS2='> '
PS4='+ '

# NLS nuisances.
LC_ALL=C
export LC_ALL
LANGUAGE=C
export LANGUAGE

# CDPATH.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

# as_fn_error STATUS ERROR [LINENO LOG_FD]
# -----
# Output "`basename $0`: error: ERROR" to stderr. If LINENO and LOG_FD
# are
# provided, also output the error to LOG_FD, referencing LINENO. Then
# exit the
# script with STATUS, using 1 if that was 0.
as_fn_error ()
{
  as_status=$1; test $as_status -eq 0 && as_status=1
  if test "$4"; then
    as_lineno=${as_lineno-"$3"}
as_lineno_stack=as_lineno_stack=$as_lineno_stack
    $as_echo "$as_me:${as_lineno-$LINENO}: error: $2" >&$4
  fi
  $as_echo "$as_me: error: $2" >&2
  as_fn_exit $as_status
} # as_fn_error

# as_fn_set_status STATUS
# -----
# Set $? to STATUS, without forking.
as_fn_set_status ()
{
  return $1
} # as_fn_set_status

# as_fn_exit STATUS
# -----
# Exit the shell with STATUS, even in a "trap 0" or "set -e" context.
as_fn_exit ()
{
  set +e

```

```

    as_fn_set_status $1
    exit $1
} # as_fn_exit

# as_fn_unset VAR
# -----
# Portably unset VAR.
as_fn_unset ()
{
    { eval $1=; unset $1;}
}
as_unset=as_fn_unset
# as_fn_append VAR VALUE
# -----
# Append the text in VALUE to the end of the definition contained in
VAR. Take
# advantage of any shell optimizations that allow amortized linear
growth over
# repeated appends, instead of the typical quadratic growth present in
naive
# implementations.
if (eval "as_var=1; as_var+=2; test x\${as_var} = x12") 2>/dev/null;
then :
    eval 'as_fn_append ()
        {
            eval $1+=\${2}
        }'
else
    as_fn_append ()
    {
        eval $1=\${$1}\${2}
    }
fi # as_fn_append

# as_fn_arith ARG...
# -----
# Perform arithmetic evaluation on the ARGs, and store the result in
the
# global $as_val. Take advantage of shells that can avoid forks. The
arguments
# must be portable across $(( )) and expr.
if (eval "test \${(( 1 + 1 ))} = 2") 2>/dev/null; then :
    eval 'as_fn_arith ()
        {
            as_val=$(( $* ))
        }'
else
    as_fn_arith ()
    {
        as_val=`expr "$@" || test $? -eq 1`
    }
fi # as_fn_arith

```

```

if expr a : '\(a\)' >/dev/null 2>&1 &&
    test "X`expr 00001 : '.*\(...\)`" = X001; then
    as_expr=expr
else
    as_expr=false
fi

if (basename -- /) >/dev/null 2>&1 && test "X`basename -- / 2>&1`" =
"X/"; then
    as_basename=basename
else
    as_basename=false
fi

if (as_dir=`dirname -- /` && test "X$as_dir" = X/) >/dev/null 2>&1;
then
    as_dirname=dirname
else
    as_dirname=false
fi

as_me=`$as_basename -- "$0" ||
$as_expr X/"$0" : '.*\/\([^\/]\*\)\/*$' \| \| \
X"$0" : 'X\(\(\)\)$' \| \| \
X"$0" : 'X\(\(\)\)' \| \| . 2>/dev/null ||
$as_echo X/"$0" |
sed '/^\.*\/\([^\/]\*\)\/*$/{
    s//\1/
    q
}
/^X\(\(\)\)$/{
    s//\1/
    q
}
/^X\(\(\)\)\.*$/{
    s//\1/
    q
}
s/.*\/./; q'`

# Avoid depending upon Character Ranges.
as_cr_letters='abcdefghijklmnopqrstuvwxyz'
as_cr_LETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
as_cr_Letters=$as_cr_letters$as_cr_LETTERS
as_cr_digits='0123456789'
as_cr_alnum=$as_cr_Letters$as_cr_digits

ECHO_C= ECHO_N= ECHO_T=
case `echo -n x` in #((((
-n*)

```

```

case `echo 'xy\c'` in
*c*) ECHO_T=' ';; # ECHO_T is single tab character.
xy)  ECHO_C='\c';;
*)   echo `echo ksh88 bug on AIX 6.1` > /dev/null
      ECHO_T=' ';;
esac;;
*)
  ECHO_N='-n';;
esac

rm -f conf$$ conf$$exe conf$$file
if test -d conf$$dir; then
  rm -f conf$$dir/conf$$file
else
  rm -f conf$$dir
  mkdir conf$$dir 2>/dev/null
fi
if (echo >conf$$file) 2>/dev/null; then
  if ln -s conf$$file conf$$ 2>/dev/null; then
    as_ln_s='ln -s'
    # ... but there are two gotchas:
    # 1) On MSYS, both `ln -s file dir' and `ln file dir' fail.
    # 2) DJGPP < 2.04 has no symlinks; `ln -s' creates a wrapper
    executable.
    # In both cases, we have to default to `cp -pR'.
    ln -s conf$$file conf$$dir 2>/dev/null && test ! -f conf$$exe
  ||
    as_ln_s='cp -pR'
  elif ln conf$$file conf$$ 2>/dev/null; then
    as_ln_s=ln
  else
    as_ln_s='cp -pR'
  fi
else
  as_ln_s='cp -pR'
fi
rm -f conf$$ conf$$exe conf$$dir/conf$$file conf$$file
rmdir conf$$dir 2>/dev/null

# as_fn_mkdir_p
# -----
# Create "$as_dir" as a directory, including parents if necessary.
as_fn_mkdir_p ()
{
  case $as_dir in #(
  -*) as_dir=./$as_dir;;
  esac
  test -d "$as_dir" || eval $as_mkdir_p || {
    as_dirs=
    while ;; do

```

```

        case $as_dir in #(
          *\`) as_qdir=`$as_echo "$as_dir" | sed "s/'/'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\/g"`;;
        #'(
          *) as_qdir=$as_dir;;
        esac
        as_dirs="'$as_qdir' $as_dirs"
        as_dir=`$as_dirname -- "$as_dir" ||
$as_expr X"$as_dir" : 'X\(.*/\)\/*[^/][^/]*/*$' \|| \
  X"$as_dir" : 'X\(/\)\[^/]' \|| \
  X"$as_dir" : 'X\(/\)\$' \|| \
  X"$as_dir" : 'X\(/\)' \|| . 2>/dev/null ||
$as_echo X"$as_dir" |
  sed '/^X\(.*/\)\*\/*[^/][^/]*\/*$/{
    s//\1/
    q
  }
/^X\(\.\.\.\)\[^/].*/{
  s//\1/
  q
}
/^X\(\.\.\.\)\$/{
  s//\1/
  q
}
/^X\(\.\.\)\.*/{
  s//\1/
  q
}
s/.*\/./; q'`
        test -d "$as_dir" && break
    done
    test -z "$as_dirs" || eval "mkdir $as_dirs"
  } || test -d "$as_dir" || as_fn_error $? "cannot create directory
$as_dir"

} # as_fn_mkdir_p
if mkdir -p . 2>/dev/null; then
  as_mkdir_p='mkdir -p "$as_dir"'
else
  test -d ./-p && rmdir ./-p
  as_mkdir_p=false
fi

# as_fn_executable_p FILE
# -----
# Test if FILE is an executable regular file.
as_fn_executable_p ()
{
  test -f "$1" && test -x "$1"
} # as_fn_executable_p

```

```

as_test_x='test -x'
as_executable_p=as_fn_executable_p

# Sed expression to map a string onto a valid CPP name.
as_tr_cpp="eval sed
'y%*$as_cr_letters%P$as_cr_LETTERS%;s%[^_$as_cr_alnum]%%_g'"

# Sed expression to map a string onto a valid variable name.
as_tr_sh="eval sed 'y%*+%pp%;s%[^_$as_cr_alnum]%%_g'"

exec 6>&1
## ----- ##
## Main body of $CONFIG_STATUS script. ##
## ----- ##
_ACEOF
test $as_write_fail = 0 && chmod +x $CONFIG_STATUS || ac_write_fail=1

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# Save the log message, to keep $0 and so on meaningful, and to
# report actual input values of CONFIG_FILES etc. instead of their
# values after options handling.
ac_log="
This file was extended by dbus $as_me 1.6.8, which was
generated by GNU Autoconf 2.69.  Invocation command line was

    CONFIG_FILES    = $CONFIG_FILES
    CONFIG_HEADERS  = $CONFIG_HEADERS
    CONFIG_LINKS    = $CONFIG_LINKS
    CONFIG_COMMANDS = $CONFIG_COMMANDS
$ $0 $@

on `(hostname || uname -n) 2>/dev/null | sed 1q`
"

_ACEOF

case $ac_config_files in *)
  *) set x $ac_config_files; shift; ac_config_files=$*;;
esac

case $ac_config_headers in *)
  *) set x $ac_config_headers; shift; ac_config_headers=$*;;
esac

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# Files that config.status was made for.
config_files="$ac_config_files"
config_headers="$ac_config_headers"
config_commands="$ac_config_commands"

```

_ACEOF

```
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
ac_cs_usage="\
\`$as_me' instantiates files and other configuration actions
from templates according to the current configuration. Unless the
files
and actions are specified as TAGs, all are instantiated by default.
```

Usage: \$0 [OPTION]... [TAG]...

```
-h, --help          print this help, then exit
-V, --version       print version number and configuration settings,
then exit
--config           print configuration, then exit
-q, --quiet, --silent
do not print progress messages
-d, --debug         don't remove temporary files
--recheck          update $as_me by reconfiguring in the same
conditions
--file=FILE[:TEMPLATE]
instantiate the configuration file FILE
--header=FILE[:TEMPLATE]
instantiate the configuration header FILE
```

Configuration files:
\$config_files

Configuration headers:
\$config_headers

Configuration commands:
\$config_commands

Report bugs to
<https://bugs.freedesktop.org/enter_bug.cgi?product=dbus>."

_ACEOF

```
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
ac_cs_config="\`$as_echo "$ac_configure_args" | sed 's/^ //;
s/[\\\"'`\\$]/\\\\\\&/g'`"
ac_cs_version="\`
dbus config.status 1.6.8
configured by $0, generated by GNU Autoconf 2.69,
with options \\\"$ac_cs_config\\\""
```

Copyright (C) 2012 Free Software Foundation, Inc.
This config.status script is free software; the Free Software
Foundation
gives unlimited permission to copy, distribute and modify it."

ac_pwd='\$ac_pwd'


```

        case $ac_optarg in
        *\'*) ac_optarg=`$as_echo "$ac_optarg" | sed "s/'/'\\\''/g"`
;;
        esac
        as_fn_append CONFIG_HEADERS " '$ac_optarg'"
        ac_need_defaults=false;;
--he | --h)
        # Conflict between --help and --header
        as_fn_error $? "ambiguous option: \`$1'
Try \`$0 --help' for more information.";;
--help | --hel | -h )
        $as_echo "$ac_cs_usage"; exit ;;
-q | -quiet | --quiet | --quie | --qui | --qu | --q \
| -silent | --silent | --silen | --sile | --sil | --si | --s)
        ac_cs_silent=: ;;

# This is an error.
-*) as_fn_error $? "unrecognized option: \`$1'
Try \`$0 --help' for more information." ;;

*) as_fn_append ac_config_targets " $1"
        ac_need_defaults=false ;;

        esac
        shift
done

ac_configure_extra_args=

if $ac_cs_silent; then
    exec 6>/dev/null
    ac_configure_extra_args="$ac_configure_extra_args --silent"
fi

_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
if $ac_cs_recheck; then
    set X $$SHELL '$0' $ac_configure_args $ac_configure_extra_args --no-
create --no-recursion
    shift
    \$as_echo "running CONFIG_SHELL=$SHELL \$*" >&6
    CONFIG_SHELL='$SHELL'
    export CONFIG_SHELL
    exec "\$@"
fi

_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
exec 5>>config.log
{
    echo
    sed 'h;s/./-/g;s/^\.../## /;s/...$/ ##/;p;x;p;x' <<_ASBOX

```

```

## Running $as_me. ##
_ASBOX
  $as_echo "$ac_log"
} >&5

_ACEOF
cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
#
# INIT-COMMANDS
#
AMDEP_TRUE="$AMDEP_TRUE" ac_aux_dir="$ac_aux_dir"

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

sed_quote_subst='$sed_quote_subst'
double_quote_subst='$double_quote_subst'
delay_variable_subst='$delay_variable_subst'
macro_version='`$ECHO "$macro_version" | $SED
"$delay_single_quote_subst"`'
macro_revision='`$ECHO "$macro_revision" | $SED
"$delay_single_quote_subst"`'
enable_shared='`$ECHO "$enable_shared" | $SED
"$delay_single_quote_subst"`'
enable_static='`$ECHO "$enable_static" | $SED
"$delay_single_quote_subst"`'
pic_mode='`$ECHO "$pic_mode" | $SED "$delay_single_quote_subst"`'
enable_fast_install='`$ECHO "$enable_fast_install" | $SED
"$delay_single_quote_subst"`'
SHELL='`$ECHO "$SHELL" | $SED "$delay_single_quote_subst"`'
ECHO='`$ECHO "$ECHO" | $SED "$delay_single_quote_subst"`'
PATH_SEPARATOR='`$ECHO "$PATH_SEPARATOR" | $SED
"$delay_single_quote_subst"`'
host_alias='`$ECHO "$host_alias" | $SED "$delay_single_quote_subst"`'
host='`$ECHO "$host" | $SED "$delay_single_quote_subst"`'
host_os='`$ECHO "$host_os" | $SED "$delay_single_quote_subst"`'
build_alias='`$ECHO "$build_alias" | $SED
"$delay_single_quote_subst"`'
build='`$ECHO "$build" | $SED "$delay_single_quote_subst"`'
build_os='`$ECHO "$build_os" | $SED "$delay_single_quote_subst"`'
SED='`$ECHO "$SED" | $SED "$delay_single_quote_subst"`'
Xsed='`$ECHO "$Xsed" | $SED "$delay_single_quote_subst"`'
GREP='`$ECHO "$GREP" | $SED "$delay_single_quote_subst"`'
EGREP='`$ECHO "$EGREP" | $SED "$delay_single_quote_subst"`'
FGREP='`$ECHO "$FGREP" | $SED "$delay_single_quote_subst"`'
LD='`$ECHO "$LD" | $SED "$delay_single_quote_subst"`'
NM='`$ECHO "$NM" | $SED "$delay_single_quote_subst"`'
LN_S='`$ECHO "$LN_S" | $SED "$delay_single_quote_subst"`'
max_cmd_len='`$ECHO "$max_cmd_len" | $SED
"$delay_single_quote_subst"`'

```

```
ac_objext='`$ECHO "$ac_objext" | $SED "$delay_single_quote_subst"``'
exeext='`$ECHO "$exeext" | $SED "$delay_single_quote_subst"``'
lt_unset='`$ECHO "$lt_unset" | $SED "$delay_single_quote_subst"``'
lt_SP2NL='`$ECHO "$lt_SP2NL" | $SED "$delay_single_quote_subst"``'
lt_NL2SP='`$ECHO "$lt_NL2SP" | $SED "$delay_single_quote_subst"``'
lt_cv_to_host_file_cmd='`$ECHO "$lt_cv_to_host_file_cmd" | $SED
"$delay_single_quote_subst"``'
lt_cv_to_tool_file_cmd='`$ECHO "$lt_cv_to_tool_file_cmd" | $SED
"$delay_single_quote_subst"``'
reload_flag='`$ECHO "$reload_flag" | $SED
"$delay_single_quote_subst"``'
reload_cmds='`$ECHO "$reload_cmds" | $SED
"$delay_single_quote_subst"``'
OBJDUMP='`$ECHO "$OBJDUMP" | $SED "$delay_single_quote_subst"``'
deplibs_check_method='`$ECHO "$deplibs_check_method" | $SED
"$delay_single_quote_subst"``'
file_magic_cmd='`$ECHO "$file_magic_cmd" | $SED
"$delay_single_quote_subst"``'
file_magic_glob='`$ECHO "$file_magic_glob" | $SED
"$delay_single_quote_subst"``'
want_nocaseglob='`$ECHO "$want_nocaseglob" | $SED
"$delay_single_quote_subst"``'
DLLTOOL='`$ECHO "$DLLTOOL" | $SED "$delay_single_quote_subst"``'
sharedlib_from_linklib_cmd='`$ECHO "$sharedlib_from_linklib_cmd" |
$SED "$delay_single_quote_subst"``'
AR='`$ECHO "$AR" | $SED "$delay_single_quote_subst"``'
AR_FLAGS='`$ECHO "$AR_FLAGS" | $SED "$delay_single_quote_subst"``'
archiver_list_spec='`$ECHO "$archiver_list_spec" | $SED
"$delay_single_quote_subst"``'
STRIP='`$ECHO "$STRIP" | $SED "$delay_single_quote_subst"``'
RANLIB='`$ECHO "$RANLIB" | $SED "$delay_single_quote_subst"``'
old_postinstall_cmds='`$ECHO "$old_postinstall_cmds" | $SED
"$delay_single_quote_subst"``'
old_postuninstall_cmds='`$ECHO "$old_postuninstall_cmds" | $SED
"$delay_single_quote_subst"``'
old_archive_cmds='`$ECHO "$old_archive_cmds" | $SED
"$delay_single_quote_subst"``'
lock_old_archive_extraction='`$ECHO "$lock_old_archive_extraction" |
$SED "$delay_single_quote_subst"``'
CC='`$ECHO "$CC" | $SED "$delay_single_quote_subst"``'
CFLAGS='`$ECHO "$CFLAGS" | $SED "$delay_single_quote_subst"``'
compiler='`$ECHO "$compiler" | $SED "$delay_single_quote_subst"``'
GCC='`$ECHO "$GCC" | $SED "$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_pipe='`$ECHO "$lt_cv_sys_global_symbol_pipe" |
$SED "$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_to_cdecl='`$ECHO
"$lt_cv_sys_global_symbol_to_cdecl" | $SED
"$delay_single_quote_subst"``'
lt_cv_sys_global_symbol_to_c_name_address='`$ECHO
"$lt_cv_sys_global_symbol_to_c_name_address" | $SED
"$delay_single_quote_subst"``'
```

```
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix=`$ECHO
"$lt_cv_sys_global_symbol_to_c_name_address_lib_prefix" | $SED
"$delay_single_quote_subst"`
nm_file_list_spec=`$ECHO "$nm_file_list_spec" | $SED
"$delay_single_quote_subst"`
lt_sysroot=`$ECHO "$lt_sysroot" | $SED "$delay_single_quote_subst"`
objdir=`$ECHO "$objdir" | $SED "$delay_single_quote_subst"`
MAGIC_CMD=`$ECHO "$MAGIC_CMD" | $SED "$delay_single_quote_subst"`
lt_prog_compiler_no_builtin_flag=`$ECHO
"$lt_prog_compiler_no_builtin_flag" | $SED
"$delay_single_quote_subst"`
lt_prog_compiler_pic=`$ECHO "$lt_prog_compiler_pic" | $SED
"$delay_single_quote_subst"`
lt_prog_compiler_wl=`$ECHO "$lt_prog_compiler_wl" | $SED
"$delay_single_quote_subst"`
lt_prog_compiler_static=`$ECHO "$lt_prog_compiler_static" | $SED
"$delay_single_quote_subst"`
lt_cv_prog_compiler_c_o=`$ECHO "$lt_cv_prog_compiler_c_o" | $SED
"$delay_single_quote_subst"`
need_locks=`$ECHO "$need_locks" | $SED "$delay_single_quote_subst"`
MANIFEST_TOOL=`$ECHO "$MANIFEST_TOOL" | $SED
"$delay_single_quote_subst"`
DSYMUTIL=`$ECHO "$DSYMUTIL" | $SED "$delay_single_quote_subst"`
NMEDIT=`$ECHO "$NMEDIT" | $SED "$delay_single_quote_subst"`
LIPO=`$ECHO "$LIPO" | $SED "$delay_single_quote_subst"`
OTOOL=`$ECHO "$OTOOL" | $SED "$delay_single_quote_subst"`
OTOOL64=`$ECHO "$OTOOL64" | $SED "$delay_single_quote_subst"`
libext=`$ECHO "$libext" | $SED "$delay_single_quote_subst"`
shrext_cmds=`$ECHO "$shrext_cmds" | $SED
"$delay_single_quote_subst"`
extract_expsyms_cmds=`$ECHO "$extract_expsyms_cmds" | $SED
"$delay_single_quote_subst"`
archive_cmds_need_lc=`$ECHO "$archive_cmds_need_lc" | $SED
"$delay_single_quote_subst"`
enable_shared_with_static_runtimes=`$ECHO
"$enable_shared_with_static_runtimes" | $SED
"$delay_single_quote_subst"`
export_dynamic_flag_spec=`$ECHO "$export_dynamic_flag_spec" | $SED
"$delay_single_quote_subst"`
whole_archive_flag_spec=`$ECHO "$whole_archive_flag_spec" | $SED
"$delay_single_quote_subst"`
compiler_needs_object=`$ECHO "$compiler_needs_object" | $SED
"$delay_single_quote_subst"`
old_archive_from_new_cmds=`$ECHO "$old_archive_from_new_cmds" | $SED
"$delay_single_quote_subst"`
old_archive_from_expsyms_cmds=`$ECHO "$old_archive_from_expsyms_cmds"
| $SED "$delay_single_quote_subst"`
archive_cmds=`$ECHO "$archive_cmds" | $SED
"$delay_single_quote_subst"`
archive_expsym_cmds=`$ECHO "$archive_expsym_cmds" | $SED
"$delay_single_quote_subst"`
```

```
module_cmds='`$ECHO "$module_cmds" | $SED
"$delay_single_quote_subst"`'
module_expsym_cmds='`$ECHO "$module_expsym_cmds" | $SED
"$delay_single_quote_subst"`'
with_gnu_ld='`$ECHO "$with_gnu_ld" | $SED
"$delay_single_quote_subst"`'
allow_undefined_flag='`$ECHO "$allow_undefined_flag" | $SED
"$delay_single_quote_subst"`'
no_undefined_flag='`$ECHO "$no_undefined_flag" | $SED
"$delay_single_quote_subst"`'
hardcode_libdir_flag_spec='`$ECHO "$hardcode_libdir_flag_spec" | $SED
"$delay_single_quote_subst"`'
hardcode_libdir_separator='`$ECHO "$hardcode_libdir_separator" | $SED
"$delay_single_quote_subst"`'
hardcode_direct='`$ECHO "$hardcode_direct" | $SED
"$delay_single_quote_subst"`'
hardcode_direct_absolute='`$ECHO "$hardcode_direct_absolute" | $SED
"$delay_single_quote_subst"`'
hardcode_minus_L='`$ECHO "$hardcode_minus_L" | $SED
"$delay_single_quote_subst"`'
hardcode_shlibpath_var='`$ECHO "$hardcode_shlibpath_var" | $SED
"$delay_single_quote_subst"`'
hardcode_automatic='`$ECHO "$hardcode_automatic" | $SED
"$delay_single_quote_subst"`'
inherit_rpath='`$ECHO "$inherit_rpath" | $SED
"$delay_single_quote_subst"`'
link_all_deplibs='`$ECHO "$link_all_deplibs" | $SED
"$delay_single_quote_subst"`'
always_export_symbols='`$ECHO "$always_export_symbols" | $SED
"$delay_single_quote_subst"`'
export_symbols_cmds='`$ECHO "$export_symbols_cmds" | $SED
"$delay_single_quote_subst"`'
exclude_expsyms='`$ECHO "$exclude_expsyms" | $SED
"$delay_single_quote_subst"`'
include_expsyms='`$ECHO "$include_expsyms" | $SED
"$delay_single_quote_subst"`'
prelink_cmds='`$ECHO "$prelink_cmds" | $SED
"$delay_single_quote_subst"`'
postlink_cmds='`$ECHO "$postlink_cmds" | $SED
"$delay_single_quote_subst"`'
file_list_spec='`$ECHO "$file_list_spec" | $SED
"$delay_single_quote_subst"`'
variables_saved_for_relink='`$ECHO "$variables_saved_for_relink" |
$SED "$delay_single_quote_subst"`'
need_lib_prefix='`$ECHO "$need_lib_prefix" | $SED
"$delay_single_quote_subst"`'
need_version='`$ECHO "$need_version" | $SED
"$delay_single_quote_subst"`'
version_type='`$ECHO "$version_type" | $SED
"$delay_single_quote_subst"`'
runpath_var='`$ECHO "$runpath_var" | $SED
"$delay_single_quote_subst"`'
```

```
shlibpath_var='`$ECHO "$shlibpath_var" | $SED
"$delay_single_quote_subst"`'
shlibpath_overrides_runpath='`$ECHO "$shlibpath_overrides_runpath" |
$SED "$delay_single_quote_subst"`'
libname_spec='`$ECHO "$libname_spec" | $SED
"$delay_single_quote_subst"`'
library_names_spec='`$ECHO "$library_names_spec" | $SED
"$delay_single_quote_subst"`'
soname_spec='`$ECHO "$soname_spec" | $SED
"$delay_single_quote_subst"`'
install_override_mode='`$ECHO "$install_override_mode" | $SED
"$delay_single_quote_subst"`'
postinstall_cmds='`$ECHO "$postinstall_cmds" | $SED
"$delay_single_quote_subst"`'
postuninstall_cmds='`$ECHO "$postuninstall_cmds" | $SED
"$delay_single_quote_subst"`'
finish_cmds='`$ECHO "$finish_cmds" | $SED
"$delay_single_quote_subst"`'
finish_eval='`$ECHO "$finish_eval" | $SED
"$delay_single_quote_subst"`'
hardcode_into_libs='`$ECHO "$hardcode_into_libs" | $SED
"$delay_single_quote_subst"`'
sys_lib_search_path_spec='`$ECHO "$sys_lib_search_path_spec" | $SED
"$delay_single_quote_subst"`'
sys_lib_dlsearch_path_spec='`$ECHO "$sys_lib_dlsearch_path_spec" |
$SED "$delay_single_quote_subst"`'
hardcode_action='`$ECHO "$hardcode_action" | $SED
"$delay_single_quote_subst"`'
enable_dlopen='`$ECHO "$enable_dlopen" | $SED
"$delay_single_quote_subst"`'
enable_dlopen_self='`$ECHO "$enable_dlopen_self" | $SED
"$delay_single_quote_subst"`'
enable_dlopen_self_static='`$ECHO "$enable_dlopen_self_static" | $SED
"$delay_single_quote_subst"`'
old_striplib='`$ECHO "$old_striplib" | $SED
"$delay_single_quote_subst"`'
striplib='`$ECHO "$striplib" | $SED "$delay_single_quote_subst"`'
compiler_lib_search_dirs='`$ECHO "$compiler_lib_search_dirs" | $SED
"$delay_single_quote_subst"`'
predep_objects='`$ECHO "$predep_objects" | $SED
"$delay_single_quote_subst"`'
postdep_objects='`$ECHO "$postdep_objects" | $SED
"$delay_single_quote_subst"`'
predeps='`$ECHO "$predeps" | $SED "$delay_single_quote_subst"`'
postdeps='`$ECHO "$postdeps" | $SED "$delay_single_quote_subst"`'
compiler_lib_search_path='`$ECHO "$compiler_lib_search_path" | $SED
"$delay_single_quote_subst"`'
LD_CXX='`$ECHO "$LD_CXX" | $SED "$delay_single_quote_subst"`'
LD_RC='`$ECHO "$LD_RC" | $SED "$delay_single_quote_subst"`'
reload_flag_CXX='`$ECHO "$reload_flag_CXX" | $SED
"$delay_single_quote_subst"`'
```

```
reload_flag_RC='`$ECHO "$reload_flag_RC" | $SED
"$delay_single_quote_subst"`'
reload_cmds_CXX='`$ECHO "$reload_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
reload_cmds_RC='`$ECHO "$reload_cmds_RC" | $SED
"$delay_single_quote_subst"`'
old_archive_cmds_CXX='`$ECHO "$old_archive_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
old_archive_cmds_RC='`$ECHO "$old_archive_cmds_RC" | $SED
"$delay_single_quote_subst"`'
compiler_CXX='`$ECHO "$compiler_CXX" | $SED
"$delay_single_quote_subst"`'
compiler_RC='`$ECHO "$compiler_RC" | $SED
"$delay_single_quote_subst"`'
GCC_CXX='`$ECHO "$GCC_CXX" | $SED "$delay_single_quote_subst"`'
GCC_RC='`$ECHO "$GCC_RC" | $SED "$delay_single_quote_subst"`'
lt_prog_compiler_no_builtin_flag_CXX='`$ECHO
"$lt_prog_compiler_no_builtin_flag_CXX" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_no_builtin_flag_RC='`$ECHO
"$lt_prog_compiler_no_builtin_flag_RC" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_pic_CXX='`$ECHO "$lt_prog_compiler_pic_CXX" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_pic_RC='`$ECHO "$lt_prog_compiler_pic_RC" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_wl_CXX='`$ECHO "$lt_prog_compiler_wl_CXX" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_wl_RC='`$ECHO "$lt_prog_compiler_wl_RC" | $SED
"$delay_single_quote_subst"`'
lt_prog_compiler_static_CXX='`$ECHO "$lt_prog_compiler_static_CXX" |
$SED "$delay_single_quote_subst"`'
lt_prog_compiler_static_RC='`$ECHO "$lt_prog_compiler_static_RC" |
$SED "$delay_single_quote_subst"`'
lt_cv_prog_compiler_c_o_CXX='`$ECHO "$lt_cv_prog_compiler_c_o_CXX" |
$SED "$delay_single_quote_subst"`'
lt_cv_prog_compiler_c_o_RC='`$ECHO "$lt_cv_prog_compiler_c_o_RC" |
$SED "$delay_single_quote_subst"`'
archive_cmds_need_lc_CXX='`$ECHO "$archive_cmds_need_lc_CXX" | $SED
"$delay_single_quote_subst"`'
archive_cmds_need_lc_RC='`$ECHO "$archive_cmds_need_lc_RC" | $SED
"$delay_single_quote_subst"`'
enable_shared_with_static_runtimes_CXX='`$ECHO
"$enable_shared_with_static_runtimes_CXX" | $SED
"$delay_single_quote_subst"`'
enable_shared_with_static_runtimes_RC='`$ECHO
"$enable_shared_with_static_runtimes_RC" | $SED
"$delay_single_quote_subst"`'
export_dynamic_flag_spec_CXX='`$ECHO "$export_dynamic_flag_spec_CXX" |
$SED "$delay_single_quote_subst"`'
export_dynamic_flag_spec_RC='`$ECHO "$export_dynamic_flag_spec_RC" |
$SED "$delay_single_quote_subst"`'
```



```
whole_archive_flag_spec_CXX='`$ECHO "$whole_archive_flag_spec_CXX" |
$SED "$delay_single_quote_subst"`'
whole_archive_flag_spec_RC='`$ECHO "$whole_archive_flag_spec_RC" |
$SED "$delay_single_quote_subst"`'
compiler_needs_object_CXX='`$ECHO "$compiler_needs_object_CXX" | $SED
"$delay_single_quote_subst"`'
compiler_needs_object_RC='`$ECHO "$compiler_needs_object_RC" | $SED
"$delay_single_quote_subst"`'
old_archive_from_new_cmds_CXX='`$ECHO "$old_archive_from_new_cmds_CXX"
| $SED "$delay_single_quote_subst"`'
old_archive_from_new_cmds_RC='`$ECHO "$old_archive_from_new_cmds_RC" |
$SED "$delay_single_quote_subst"`'
old_archive_from_expsyms_cmds_CXX='`$ECHO
"$old_archive_from_expsyms_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
old_archive_from_expsyms_cmds_RC='`$ECHO
"$old_archive_from_expsyms_cmds_RC" | $SED
"$delay_single_quote_subst"`'
archive_cmds_CXX='`$ECHO "$archive_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
archive_cmds_RC='`$ECHO "$archive_cmds_RC" | $SED
"$delay_single_quote_subst"`'
archive_expsym_cmds_CXX='`$ECHO "$archive_expsym_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
archive_expsym_cmds_RC='`$ECHO "$archive_expsym_cmds_RC" | $SED
"$delay_single_quote_subst"`'
module_cmds_CXX='`$ECHO "$module_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
module_cmds_RC='`$ECHO "$module_cmds_RC" | $SED
"$delay_single_quote_subst"`'
module_expsym_cmds_CXX='`$ECHO "$module_expsym_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
module_expsym_cmds_RC='`$ECHO "$module_expsym_cmds_RC" | $SED
"$delay_single_quote_subst"`'
with_gnu_ld_CXX='`$ECHO "$with_gnu_ld_CXX" | $SED
"$delay_single_quote_subst"`'
with_gnu_ld_RC='`$ECHO "$with_gnu_ld_RC" | $SED
"$delay_single_quote_subst"`'
allow_undefined_flag_CXX='`$ECHO "$allow_undefined_flag_CXX" | $SED
"$delay_single_quote_subst"`'
allow_undefined_flag_RC='`$ECHO "$allow_undefined_flag_RC" | $SED
"$delay_single_quote_subst"`'
no_undefined_flag_CXX='`$ECHO "$no_undefined_flag_CXX" | $SED
"$delay_single_quote_subst"`'
no_undefined_flag_RC='`$ECHO "$no_undefined_flag_RC" | $SED
"$delay_single_quote_subst"`'
hardcode_libdir_flag_spec_CXX='`$ECHO "$hardcode_libdir_flag_spec_CXX"
| $SED "$delay_single_quote_subst"`'
hardcode_libdir_flag_spec_RC='`$ECHO "$hardcode_libdir_flag_spec_RC" |
$SED "$delay_single_quote_subst"`'
hardcode_libdir_separator_CXX='`$ECHO "$hardcode_libdir_separator_CXX"
| $SED "$delay_single_quote_subst"`'
```

```
hardcode_libdir_separator_RC='`$ECHO "$hardcode_libdir_separator_RC" |
$SED "$delay_single_quote_subst"`'
hardcode_direct_CXX='`$ECHO "$hardcode_direct_CXX" | $SED
"$delay_single_quote_subst"`'
hardcode_direct_RC='`$ECHO "$hardcode_direct_RC" | $SED
"$delay_single_quote_subst"`'
hardcode_direct_absolute_CXX='`$ECHO "$hardcode_direct_absolute_CXX" |
$SED "$delay_single_quote_subst"`'
hardcode_direct_absolute_RC='`$ECHO "$hardcode_direct_absolute_RC" |
$SED "$delay_single_quote_subst"`'
hardcode_minus_L_CXX='`$ECHO "$hardcode_minus_L_CXX" | $SED
"$delay_single_quote_subst"`'
hardcode_minus_L_RC='`$ECHO "$hardcode_minus_L_RC" | $SED
"$delay_single_quote_subst"`'
hardcode_shlibpath_var_CXX='`$ECHO "$hardcode_shlibpath_var_CXX" |
$SED "$delay_single_quote_subst"`'
hardcode_shlibpath_var_RC='`$ECHO "$hardcode_shlibpath_var_RC" | $SED
"$delay_single_quote_subst"`'
hardcode_automatic_CXX='`$ECHO "$hardcode_automatic_CXX" | $SED
"$delay_single_quote_subst"`'
hardcode_automatic_RC='`$ECHO "$hardcode_automatic_RC" | $SED
"$delay_single_quote_subst"`'
inherit_rpath_CXX='`$ECHO "$inherit_rpath_CXX" | $SED
"$delay_single_quote_subst"`'
inherit_rpath_RC='`$ECHO "$inherit_rpath_RC" | $SED
"$delay_single_quote_subst"`'
link_all_deplibs_CXX='`$ECHO "$link_all_deplibs_CXX" | $SED
"$delay_single_quote_subst"`'
link_all_deplibs_RC='`$ECHO "$link_all_deplibs_RC" | $SED
"$delay_single_quote_subst"`'
always_export_symbols_CXX='`$ECHO "$always_export_symbols_CXX" | $SED
"$delay_single_quote_subst"`'
always_export_symbols_RC='`$ECHO "$always_export_symbols_RC" | $SED
"$delay_single_quote_subst"`'
export_symbols_cmds_CXX='`$ECHO "$export_symbols_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
export_symbols_cmds_RC='`$ECHO "$export_symbols_cmds_RC" | $SED
"$delay_single_quote_subst"`'
exclude_expsyms_CXX='`$ECHO "$exclude_expsyms_CXX" | $SED
"$delay_single_quote_subst"`'
exclude_expsyms_RC='`$ECHO "$exclude_expsyms_RC" | $SED
"$delay_single_quote_subst"`'
include_expsyms_CXX='`$ECHO "$include_expsyms_CXX" | $SED
"$delay_single_quote_subst"`'
include_expsyms_RC='`$ECHO "$include_expsyms_RC" | $SED
"$delay_single_quote_subst"`'
prelink_cmds_CXX='`$ECHO "$prelink_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
prelink_cmds_RC='`$ECHO "$prelink_cmds_RC" | $SED
"$delay_single_quote_subst"`'
postlink_cmds_CXX='`$ECHO "$postlink_cmds_CXX" | $SED
"$delay_single_quote_subst"`'
```

```

postlink_cmds_RC='`$ECHO "$postlink_cmds_RC" | $SED
"$delay_single_quote_subst"`'
file_list_spec_CXX='`$ECHO "$file_list_spec_CXX" | $SED
"$delay_single_quote_subst"`'
file_list_spec_RC='`$ECHO "$file_list_spec_RC" | $SED
"$delay_single_quote_subst"`'
hardcode_action_CXX='`$ECHO "$hardcode_action_CXX" | $SED
"$delay_single_quote_subst"`'
hardcode_action_RC='`$ECHO "$hardcode_action_RC" | $SED
"$delay_single_quote_subst"`'
compiler_lib_search_dirs_CXX='`$ECHO "$compiler_lib_search_dirs_CXX" |
$SED "$delay_single_quote_subst"`'
compiler_lib_search_dirs_RC='`$ECHO "$compiler_lib_search_dirs_RC" |
$SED "$delay_single_quote_subst"`'
predep_objects_CXX='`$ECHO "$predep_objects_CXX" | $SED
"$delay_single_quote_subst"`'
predep_objects_RC='`$ECHO "$predep_objects_RC" | $SED
"$delay_single_quote_subst"`'
postdep_objects_CXX='`$ECHO "$postdep_objects_CXX" | $SED
"$delay_single_quote_subst"`'
postdep_objects_RC='`$ECHO "$postdep_objects_RC" | $SED
"$delay_single_quote_subst"`'
predeps_CXX='`$ECHO "$predeps_CXX" | $SED
"$delay_single_quote_subst"`'
predeps_RC='`$ECHO "$predeps_RC" | $SED "$delay_single_quote_subst"`'
postdeps_CXX='`$ECHO "$postdeps_CXX" | $SED
"$delay_single_quote_subst"`'
postdeps_RC='`$ECHO "$postdeps_RC" | $SED
"$delay_single_quote_subst"`'
compiler_lib_search_path_CXX='`$ECHO "$compiler_lib_search_path_CXX" |
$SED "$delay_single_quote_subst"`'
compiler_lib_search_path_RC='`$ECHO "$compiler_lib_search_path_RC" |
$SED "$delay_single_quote_subst"`'

```

```

LTCC='$LTCC'
LTCFLAGS='$LTCFLAGS'
compiler='$compiler_DEFAULT'

```

```

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
    eval 'cat <<_LTECHO_EOF
\${1}
_LTECHO_EOF'
}

```

```

# Quote eveled strings.
for var in SHELL \
ECHO \
PATH_SEPARATOR \
SED \
GREP \

```

EGREP \
FGREP \
LD \
NM \
LN_S \
lt_SP2NL \
lt_NL2SP \
reload_flag \
OBJDUMP \
deplibs_check_method \
file_magic_cmd \
file_magic_glob \
want_nocaseglob \
DLLTOOL \
sharedlib_from_linklib_cmd \
AR \
AR_FLAGS \
archiver_list_spec \
STRIP \
RANLIB \
CC \
CFLAGS \
compiler \
lt_cv_sys_global_symbol_pipe \
lt_cv_sys_global_symbol_to_cdecl \
lt_cv_sys_global_symbol_to_c_name_address \
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix \
nm_file_list_spec \
lt_prog_compiler_no_builtin_flag \
lt_prog_compiler_pic \
lt_prog_compiler_wl \
lt_prog_compiler_static \
lt_cv_prog_compiler_c_o \
need_locks \
MANIFEST_TOOL \
DSYMUTIL \
NMEDIT \
LIPO \
OTOOL \
OTOOL64 \
shrext_cmds \
export_dynamic_flag_spec \
whole_archive_flag_spec \
compiler_needs_object \
with_gnu_ld \
allow_undefined_flag \
no_undefined_flag \
hardcode_libdir_flag_spec \
hardcode_libdir_separator \
exclude_expsyms \
include_expsyms \
file_list_spec \

```
variables_saved_for_relink \  
libname_spec \  
library_names_spec \  
soname_spec \  
install_override_mode \  
finish_eval \  
old_striplib \  
striplib \  
compiler_lib_search_dirs \  
predep_objects \  
postdep_objects \  
predeps \  
postdeps \  
compiler_lib_search_path \  
LD_CXX \  
LD_RC \  
reload_flag_CXX \  
reload_flag_RC \  
compiler_CXX \  
compiler_RC \  
lt_prog_compiler_no_builtin_flag_CXX \  
lt_prog_compiler_no_builtin_flag_RC \  
lt_prog_compiler_pic_CXX \  
lt_prog_compiler_pic_RC \  
lt_prog_compiler_wl_CXX \  
lt_prog_compiler_wl_RC \  
lt_prog_compiler_static_CXX \  
lt_prog_compiler_static_RC \  
lt_cv_prog_compiler_c_o_CXX \  
lt_cv_prog_compiler_c_o_RC \  
export_dynamic_flag_spec_CXX \  
export_dynamic_flag_spec_RC \  
whole_archive_flag_spec_CXX \  
whole_archive_flag_spec_RC \  
compiler_needs_object_CXX \  
compiler_needs_object_RC \  
with_gnu_ld_CXX \  
with_gnu_ld_RC \  
allow_undefined_flag_CXX \  
allow_undefined_flag_RC \  
no_undefined_flag_CXX \  
no_undefined_flag_RC \  
hardcode_libdir_flag_spec_CXX \  
hardcode_libdir_flag_spec_RC \  
hardcode_libdir_separator_CXX \  
hardcode_libdir_separator_RC \  
exclude_expsyms_CXX \  
exclude_expsyms_RC \  
include_expsyms_CXX \  
include_expsyms_RC \  
file_list_spec_CXX \  
file_list_spec_RC
```

```

compiler_lib_search_dirs_CXX \
compiler_lib_search_dirs_RC \
predep_objects_CXX \
predep_objects_RC \
postdep_objects_CXX \
postdep_objects_RC \
predeps_CXX \
predeps_RC \
postdeps_CXX \
postdeps_RC \
compiler_lib_search_path_CXX \
compiler_lib_search_path_RC; do
    case `eval \\\\\\\$ECHO \\\\\\\"\\\\\\$\\$var"\\\\\\"` in
        *[\\\\\\\`\\\\\\"\\\\\\$]*)
            eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\`\\\\\\$ECHO \\\\\"\\\\\\$\\$var\\\\\\" | \\\\\\\$SED
\\\\\\"\\\\\\$sed_quote_subst\\\\\\"\\\\\\`\\\\\\\\\\\\\\\\\\""
                ;;
        *)
            eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\$\\$var\\\\\\\\\\\\\\\\\\""
                ;;
    esac
done

```

Double-quote double-eval'd strings.

```

for var in reload_cmds \
old_postinstall_cmds \
old_postuninstall_cmds \
old_archive_cmds \
extract_expsyms_cmds \
old_archive_from_new_cmds \
old_archive_from_expsyms_cmds \
archive_cmds \
archive_expsym_cmds \
module_cmds \
module_expsym_cmds \
export_symbols_cmds \
prelink_cmds \
postlink_cmds \
postinstall_cmds \
postuninstall_cmds \
finish_cmds \
sys_lib_search_path_spec \
sys_lib_dlsearch_path_spec \
reload_cmds_CXX \
reload_cmds_RC \
old_archive_cmds_CXX \
old_archive_cmds_RC \
old_archive_from_new_cmds_CXX \
old_archive_from_new_cmds_RC \
old_archive_from_expsyms_cmds_CXX \
old_archive_from_expsyms_cmds_RC \
archive_cmds_CXX \

```

```

archive_cmds_RC \
archive_expsym_cmds_CXX \
archive_expsym_cmds_RC \
module_cmds_CXX \
module_cmds_RC \
module_expsym_cmds_CXX \
module_expsym_cmds_RC \
export_symbols_cmds_CXX \
export_symbols_cmds_RC \
prelink_cmds_CXX \
prelink_cmds_RC \
postlink_cmds_CXX \
postlink_cmds_RC; do
    case `eval \\\\\\\$ECHO \\\\\\\"\\\\\\\\\\$\\$var"\\\\\\"` in
        *[\\\\\\\\\\\`\\\\\\"\\\\\\$]*)
            eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\\\\\$ECHO \\\\\\\"\\\\\\\\\\$\\$var\\\\\\" | \\\\\\\$SED -e
\\\\\\\\\\$double_quote_subst\\\\\\" -e \\\\\\\"\\\\\\\\\\$sed_quote_subst\\\\\\" -e
\\\\\\\\\\$delay_variable_subst\\\\\\"\\\\\\\\\\$\\$var"
                ;;
        *)
            eval "lt_\\$var=\\\\\\\\\\\\\\\\\\"\\\\\\\\\\$\\$var\\\\\\\\\\\\\\\\\\""
                ;;
    esac
done

```

```

ac_aux_dir='$ac_aux_dir'
xsi_shell='$xsi_shell'
lt_shell_append='$lt_shell_append'

```

```

# See if we are running on zsh, and set the options which allow our
# commands through without removal of \ escapes INIT.
if test -n "\${ZSH_VERSION+set}"; then
    setopt NO_GLOB_SUBST
fi

```

```

PACKAGE='$PACKAGE'
VERSION='$VERSION'
TIMESTAMP='$TIMESTAMP'
RM='$RM'
ofile='$ofile'

```

_ACEOF

```
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
```

```

# Handling of arguments.
for ac_config_target in $ac_config_targets
do
  case $ac_config_target in
    "config.h") CONFIG_HEADERS="$CONFIG_HEADERS config.h" ;;
    "depfiles") CONFIG_COMMANDS="$CONFIG_COMMANDS depfiles" ;;
    "libtool") CONFIG_COMMANDS="$CONFIG_COMMANDS libtool" ;;
    "Doxyfile") CONFIG_FILES="$CONFIG_FILES Doxyfile" ;;
    "dbus/versioninfo.rc") CONFIG_FILES="$CONFIG_FILES
dbus/versioninfo.rc" ;;
    "dbus/dbus-arch-deps.h") CONFIG_FILES="$CONFIG_FILES dbus/dbus-
arch-deps.h" ;;
    "bus/system.conf") CONFIG_FILES="$CONFIG_FILES bus/system.conf" ;;
    "bus/session.conf") CONFIG_FILES="$CONFIG_FILES bus/session.conf"
;;
    "bus/messagebus") CONFIG_FILES="$CONFIG_FILES bus/messagebus" ;;
    "bus/messagebus-config") CONFIG_FILES="$CONFIG_FILES
bus/messagebus-config" ;;
    "bus/org.freedesktop.dbus-session.plist")
CONFIG_FILES="$CONFIG_FILES bus/org.freedesktop.dbus-session.plist" ;;
    "bus/rc.messagebus") CONFIG_FILES="$CONFIG_FILES
bus/rc.messagebus" ;;
    "bus/dbus.service") CONFIG_FILES="$CONFIG_FILES bus/dbus.service"
;;
    "bus/dbus.socket") CONFIG_FILES="$CONFIG_FILES bus/dbus.socket" ;;
    "Makefile") CONFIG_FILES="$CONFIG_FILES Makefile" ;;
    "dbus/Makefile") CONFIG_FILES="$CONFIG_FILES dbus/Makefile" ;;
    "bus/Makefile") CONFIG_FILES="$CONFIG_FILES bus/Makefile" ;;
    "tools/Makefile") CONFIG_FILES="$CONFIG_FILES tools/Makefile" ;;
    "test/Makefile") CONFIG_FILES="$CONFIG_FILES test/Makefile" ;;
    "test/name-test/Makefile") CONFIG_FILES="$CONFIG_FILES test/name-
test/Makefile" ;;
    "doc/Makefile") CONFIG_FILES="$CONFIG_FILES doc/Makefile" ;;
    "doc/dbus-daemon.1") CONFIG_FILES="$CONFIG_FILES doc/dbus-
daemon.1" ;;
    "dbus-1.pc") CONFIG_FILES="$CONFIG_FILES dbus-1.pc" ;;
    "dbus-1-uninstalled.pc") CONFIG_FILES="$CONFIG_FILES dbus-1-
uninstalled.pc" ;;
    "test/data/valid-config-files/debug-allow-all.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files/debug-allow-
all.conf" ;;
    "test/data/valid-config-files/debug-allow-all-sha1.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files/debug-allow-
all-sha1.conf" ;;
    "test/data/valid-config-files-system/debug-allow-all-pass.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files-system/debug-
allow-all-pass.conf" ;;
    "test/data/valid-config-files-system/debug-allow-all-fail.conf")
CONFIG_FILES="$CONFIG_FILES test/data/valid-config-files-system/debug-
allow-all-fail.conf" ;;

```



```
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuite.PrivServer.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteEchoService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteForkingEchoService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteSegfaultService.service" ;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service"
;;
"test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-
files/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteEchoService.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteSegfaultService.service" ;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceSuccess.service"
;;
"test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service")
CONFIG_FILES="$CONFIG_FILES test/data/valid-service-files-
system/org.freedesktop.DBus.TestSuiteShellEchoServiceFail.service" ;;
"test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service")
CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoExec.service" ;;
"test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service")
CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoUser.service" ;;
"test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service")
```

```

CONFIG_FILES="$CONFIG_FILES test/data/invalid-service-files-
system/org.freedesktop.DBus.TestSuiteNoService.service" ;;

*) as_fn_error $? "invalid argument: \`$ac_config_target'" "$LINENO"
5;;
    esac
done

# If the user did not use the arguments to specify the items to
# instantiate,
# then the envvar interface is used.  Set only those that are not.
# We use the long form for the default assignment because of an
# extremely
# bizarre bug on SunOS 4.1.3.
if $ac_need_defaults; then
    test "${CONFIG_FILES+set}" = set || CONFIG_FILES=$config_files
    test "${CONFIG_HEADERS+set}" = set || CONFIG_HEADERS=$config_headers
    test "${CONFIG_COMMANDS+set}" = set ||
CONFIG_COMMANDS=$config_commands
fi

# Have a temporary directory for convenience.  Make it in the build
# tree
# simply because there is no reason against having it here, and in
# addition,
# creating and moving files from /tmp can sometimes cause problems.
# Hook for its removal unless debugging.
# Note that there is a small window in which the directory will not be
# cleaned:
# after its creation but before its name has been assigned to `$tmp'.
$debug ||
{
    tmp= ac_tmp=
    trap 'exit_status=$?
: "${ac_tmp:= $tmp}"
{ test ! -d "$ac_tmp" || rm -fr "$ac_tmp"; } && exit $exit_status
' 0
    trap 'as_fn_exit 1' 1 2 13 15
}
# Create a (secure) tmp directory for tmp files.

{
    tmp=`(umask 077 && mktemp -d "./confXXXXXX") 2>/dev/null` &&
    test -d "$tmp"
} ||
{
    tmp=./conf$$-$RANDOM
    (umask 077 && mkdir "$tmp")
} || as_fn_error $? "cannot create a temporary directory in ."
"$LINENO" 5
ac_tmp=$tmp

```

```

# Set up the scripts for CONFIG_FILES section.
# No need to generate them if there are no CONFIG_FILES.
# This happens for instance with `./config.status config.h'.
if test -n "$CONFIG_FILES"; then

ac_cr=`echo X | tr X '\015'`
# On cygwin, bash can eat \r inside `` if the user requested igncr.
# But we know of no other shell where ac_cr would be empty at this
# point, so we can use a bashism as a fallback.
if test "x$ac_cr" = x; then
    eval ac_cr=\$\'\r\'
fi
ac_cs_awk_cr=`$AWK 'BEGIN { print "a\rb" }' </dev/null 2>/dev/null`
if test "$ac_cs_awk_cr" = "a${ac_cr}b"; then
    ac_cs_awk_cr='\r'
else
    ac_cs_awk_cr=$ac_cr
fi

echo 'BEGIN {' >"$ac_tmp/subs1.awk" &&
_ACEOF

{
    echo "cat >conf$$$subs.awk <<_ACEOF" &&
    echo "$ac_subst_vars" | sed 's/./&!\$&$ac_delim/' &&
    echo "_ACEOF"
} >conf$$$subs.sh ||
as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5
ac_delim_num=`echo "$ac_subst_vars" | grep -c '^'`
ac_delim='%!_# '
for ac_last_try in false false false false false ;; do
    ./conf$$$subs.sh ||
        as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5

    ac_delim_n=`sed -n "s/.*$ac_delim\$/X/p" conf$$$subs.awk | grep -c X`
    if test $ac_delim_n = $ac_delim_num; then
        break
    elif $ac_last_try; then
        as_fn_error $? "could not make $CONFIG_STATUS" "$LINENO" 5
    else
        ac_delim="$ac_delim!$ac_delim _$ac_delim!! "
    fi
done
rm -f conf$$$subs.sh

cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
cat >>"\$ac_tmp/subs1.awk" <<\_ACAWK &&
_ACEOF
sed -n '

```

```

h
s/^/S["/; s/!.*/"=/  

p  

g  

s/^[^!]*!//  

:repl  

t repl  

s/"$ac_delim"$//  

t delim  

:nl  

h  

s/\(.\\{148\\}\)..*/\1/  

t more1  

s/["\\]/\&/g; s/^"/; s/$/\n"\\/  

p  

n  

b repl  

:more1  

s/["\\]/\&/g; s/^"/; s/$/"\\/  

p  

g  

s/\\.\\{148\\}//  

t nl  

:delim  

h  

s/\(.\\{148\\}\)..*/\1/  

t more2  

s/["\\]/\&/g; s/^"/; s/$"/  

p  

b  

:more2  

s/["\\]/\&/g; s/^"/; s/$/"\\/  

p  

g  

s/\\.\\{148\\}//  

t delim  

' <conf$$subs.awk | sed '  

/^[""]/{  

  N  

  s/\n//  

}  

' >>$CONFIG_STATUS || ac_write_fail=1  

rm -f conf$$subs.awk  

cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1  

_ACAWK  

cat >>"$ac_tmp/subs1.awk" <<_ACAWK &&  

  for (key in S) S_is_set[key] = 1  

  FS = " "  

}  

{  

  line = $ 0

```

```

nfields = split(line, field, "@")
substed = 0
len = length(field[1])
for (i = 2; i < nfields; i++) {
    key = field[i]
    keylen = length(key)
    if (S_is_set[key]) {
        value = S[key]
        line = substr(line, 1, len) "" value "" substr(line, len +
keylen + 3)
        len += length(value) + length(field[++i])
        substed = 1
    } else
        len += 1 + keylen
    }

print line
}

_ACAWK
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
if sed "s/$ac_cr//" < /dev/null > /dev/null 2>&1; then
    sed "s/$ac_cr\\$//; s/$ac_cr/$ac_cs_awk_cr/g"
else
    cat
fi < "$ac_tmp/subs1.awk" > "$ac_tmp/subs.awk" \
|| as_fn_error $? "could not setup config files machinery" "$LINENO"
5
_ACEOF

# VPATH may cause trouble with some makes, so we remove sole
$(srcdir),
# ${srcdir} and @srcdir@ entries from VPATH if srcdir is ".", strip
leading and
# trailing colons and then remove the whole line if VPATH becomes
empty
# (actually we leave an empty line to preserve line numbers).
if test "x$srcdir" = x.; then
    ac_vpsub='/^[\ ]*VPATH[\ ]*=[\ ]*{
h
s///
s/^\://
s/[\ ]*$\://
s/:\$(srcdir):\://g
s/:\${srcdir}:\://g
s/:\@srcdir@\://g
s/^\:*//
s/:\:*$//
x
s/\(=[\ ]*\)\.*\/\1/
G

```

```

s/\n//
s/^[^=]*=[ ]*$//
}'
fi

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
fi # test -n "$CONFIG_FILES"

# Set up the scripts for CONFIG_HEADERS section.
# No need to generate them if there are no CONFIG_HEADERS.
# This happens for instance with './config.status Makefile'.
if test -n "$CONFIG_HEADERS"; then
cat >"$ac_tmp/defines.awk" <<\_ACAWK ||
BEGIN {
\_ACEOF

# Transform confdefs.h into an awk script `defines.awk', embedded as
# here-document in config.status, that substitutes the proper values
into
# config.h.in to produce config.h.

# Create a delimiter string that does not exist in confdefs.h, to ease
# handling of long lines.
ac_delim='%!_!# '
for ac_last_try in false false ;; do
ac_tt=`sed -n "/$ac_delim/p" confdefs.h`
if test -z "$ac_tt"; then
break
elif $ac_last_try; then
as_fn_error $? "could not make $CONFIG_HEADERS" "$LINENO" 5
else
ac_delim="$ac_delim!$ac_delim _$ac_delim!! "
fi
done

# For the awk script, D is an array of macro values keyed by name,
# likewise P contains macro parameters if any. Preserve backslash
# newline sequences.

ac_word_re=[_$_as_cr_Letters][_$_as_cr_alnum]*
sed -n \
s/.\{148\}/&"$ac_delim"/g
t rset
:rset
s/^[ ]*#[ ]*define[ ]+[ ]*/ /
t def
d
:def
s/\$//
t bsnl
s/["\\]/\\&/g
s/^\ ('"$ac_word_re"'\\)\ ([[^\()]*\\)\ [ ]*\ (.*) /P["\1"]="\2"

```

```

D["\1"]=" \3"/p
s/^ \(("$ac_word_re"\\)[      ]*\(.*\)/D["\1"]=" \2"/p
d
:bsnl
s/["\\]/\\&/g
s/^ \(("$ac_word_re"\\)\(\([^\()]*\)\)[      ]*\(.*\)/P["\1"]="\2"\
D["\1"]=" \3\\\\"\\n"\\//p
t cont
s/^ \(("$ac_word_re"\\)[      ]*\(.*\)/D["\1"]=" \2\\\\"\\n"\\//p
t cont
d
:cont
n
s/.\{148\}/&'"$ac_delim"'/g
t clear
:clear
s/\\$///
t bsnlc
s/["\\]/\\&/g; s/^"/; s/$"/p
d
:bsnlc
s/["\\]/\\&/g; s/^"/; s/$/\\\\"\\n"\\//p
b cont
' <confdefs.h | sed '
s/'"$ac_delim"'/\\"\\
"/g' >>$CONFIG_STATUS || ac_write_fail=1

cat >>$CONFIG_STATUS <<_ACEOF || ac_write_fail=1
  for (key in D) D_is_set[key] = 1
  FS = " "
}
/^[ \t ]*#[ \t ]*(define|undef)[ \t ]+${ac_word_re}([ \t ]|\$)/ {
  line = \$0
  split(line, arg, " ")
  if (arg[1] == "#") {
    defundef = arg[2]
    mac1 = arg[3]
  } else {
    defundef = substr(arg[1], 2)
    mac1 = arg[2]
  }
  split(mac1, mac2, "(") #)
  macro = mac2[1]
  prefix = substr(line, 1, index(line, defundef) - 1)
  if (D_is_set[macro]) {
    # Preserve the white space surrounding the "#".
    print prefix "define", macro P[macro] D[macro]
    next
  } else {
    # Replace #undef with comments. This is necessary, for example,
    # in the case of _POSIX_SOURCE, which is predefined and required
    # on some systems where configure will not decide to define it.

```

```

    if (defundef == "undef") {
        print "/*", prefix defundef, macro, "*/"
        next
    }
}
}
{ print }
_ACAWK
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
    as_fn_error $? "could not setup config headers machinery" "$LINENO"
5
fi # test -n "$CONFIG_HEADERS"

eval set X " :F $CONFIG_FILES :H $CONFIG_HEADERS :C
$CONFIG_COMMANDS"
shift
for ac_tag
do
    case $ac_tag in
        :[FHLC]) ac_mode=$ac_tag; continue;;
    esac
    case $ac_mode$ac_tag in
        :[FHL]*:*) ;;
        :L* | :C*:*) as_fn_error $? "invalid tag \`$ac_tag'" "$LINENO" 5;;
        :[FH]-) ac_tag=-:-;;
        :[FH]*) ac_tag=$ac_tag:$ac_tag.in;;
    esac
    ac_save_IFS=$IFS
    IFS=:
    set x $ac_tag
    IFS=$ac_save_IFS
    shift
    ac_file=$1
    shift

    case $ac_mode in
        :L) ac_source=$1;;
        :[FH])
            ac_file_inputs=
            for ac_f
            do
                case $ac_f in
                    -) ac_f="$ac_tmp/stdin";;
                    *) # Look for the file first in the build tree, then in the
source tree
                        # (if the path is not absolute). The absolute path cannot be
DOS-style,
                        # because $ac_f cannot contain `:'.
                        test -f "$ac_f" ||
                            case $ac_f in

```



```

        [\\/$]*) false;;
        *) test -f "$srcdir/$ac_f" && ac_f="$srcdir/$ac_f";;
    esac ||
        as_fn_error 1 "cannot find input file: `\$ac_f'" "$LINENO" 5;;
esac
case $ac_f in
    *\\*) ac_f=`$as_echo "$ac_f" | sed
"s/'/'\\\\\\\\\\\\\\\\'/g"`;; esac
    as_fn_append ac_file_inputs " '$ac_f'"
done

# Let's still pretend it is `configure' which instantiates (i.e.,
don't
# use $as_me), people would be surprised to read:
# /* config.h. Generated by config.status. */
configure_input='Generated from `
$as_echo "$*" | sed 's|^[^:]*|/|;s|:[^:]*|/,|g'
`' by configure.'
if test x"$ac_file" != x-; then
    configure_input="$ac_file. $configure_input"
    { $as_echo "$as_me:${as_lineno-$LINENO}: creating $ac_file" >&5
$as_echo "$as_me: creating $ac_file" >&6;}
    fi
# Neutralize special characters interpreted by sed in replacement
strings.
case $configure_input in
    *\\&* | *\\|* | *\\* )
        ac_sed_conf_input=`$as_echo "$configure_input" |
sed 's/[\\\\\\&|]/\\\\\\\\&/g'`;; #
    *) ac_sed_conf_input=$configure_input;;
esac

case $ac_tag in
*:-:* | *:-) cat >"$ac_tmp/stdin" \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5 ;;
esac
;;
esac

ac_dir=`$as_dirname -- "$ac_file" ||
$as_expr X"$ac_file" : 'X\([^\/]\)\/*\([^\/]\)\/*$' \|\ \
X"$ac_file" : 'X\(/\/\)\[^\/]' \|\ \
X"$ac_file" : 'X\(/\/\)$' \|\ \
X"$ac_file" : 'X\(/\/\)' \|\ . 2>/dev/null ||
$as_echo X"$ac_file" |
sed '/^X\([^\/]\)\/*\([^\/]\)\/*$/{
s//\1/
q
}
/^X\(/\/\)\[^\/].*${
s//\1/
q
}
}'

```

```

        /^X\(\\\/\\\/)\$/ {
            s//\1/
            q
        }
        /^X\(\\\/)\.*$/ {
            s//\1/
            q
        }
        s/.*\/./; q'`
as_dir="$ac_dir"; as_fn_mkdir_p
ac_builddir=.

case "$ac_dir" in
.) ac_dir_suffix= ac_top_builddir_sub=. ac_top_build_prefix= ;;
*)
    ac_dir_suffix=`$as_echo "$ac_dir" | sed 's|^\.([\\/]|||)'`
    # A ".." for each directory in $ac_dir_suffix.
    ac_top_builddir_sub=`$as_echo "$ac_dir_suffix" | sed
's|/[^\\/]*|/..|g;s|/|||'`
    case $ac_top_builddir_sub in
    "") ac_top_builddir_sub=. ac_top_build_prefix= ;;
    *) ac_top_build_prefix=$ac_top_builddir_sub/ ;;
    esac ;;
esac
esac
ac_abs_top_builddir=$ac_pwd
ac_abs_builddir=$ac_pwd$ac_dir_suffix
# for backward compatibility:
ac_top_builddir=$ac_top_build_prefix

case $srcdir in
.) # We are building in place.
    ac_srcdir=.
    ac_top_srcdir=$ac_top_builddir_sub
    ac_abs_top_srcdir=$ac_pwd ;;
[\/]* | ?:[\/]* ) # Absolute name.
    ac_srcdir=$srcdir$ac_dir_suffix;
    ac_top_srcdir=$srcdir
    ac_abs_top_srcdir=$srcdir ;;
*) # Relative name.
    ac_srcdir=$ac_top_build_prefix$srcdir$ac_dir_suffix
    ac_top_srcdir=$ac_top_build_prefix$srcdir
    ac_abs_top_srcdir=$ac_pwd/$srcdir ;;
esac
ac_abs_srcdir=$ac_abs_top_srcdir$ac_dir_suffix

case $ac_mode in
:F)
#
# CONFIG_FILE
#

```

```

case $INSTALL in
[\\/$]* | ?:[\\/]*) ac_INSTALL=$INSTALL ;;
*) ac_INSTALL=$ac_top_build_prefix$INSTALL ;;
esac
ac_MKDIR_P=$MKDIR_P
case $MKDIR_P in
[\\/$]* | ?:[\\/]*) ;;
*/) ac_MKDIR_P=$ac_top_build_prefix$MKDIR_P ;;
esac
_ACEOF

cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
# If the template does not know about datarootdir, expand it.
# FIXME: This hack should be removed a few years after 2.60.
ac_datarootdir_hack=; ac_datarootdir_seen=
ac_sed_dataroot='
/datarootdir/ {
  p
  q
}
/@datadir@/p
/@docdir@/p
/@infodir@/p
/@localedir@/p
/@mandir@/p'
case `eval "sed -n \"\$ac_sed_dataroot\" \$ac_file_inputs"` in
*datarootdir*) ac_datarootdir_seen=yes;;
*@datadir@*|*@docdir@*|*@infodir@*|*@localedir@*|*@mandir@*)
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file_inputs
seems to ignore the --datarootdir setting" >&5
$as_echo "$as_me: WARNING: $ac_file_inputs seems to ignore the --
datarootdir setting" >&2;}
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
  ac_datarootdir_hack='
s&@datadir@&@datadir&g
s&@docdir@&@docdir&g
s&@infodir@&@infodir&g
s&@localedir@&@localedir&g
s&@mandir@&@mandir&g
s&\\\\\\$@{datarootdir}&@datarootdir&g' ;;
esac
_ACEOF

# Neutralize VPATH when `srcdir' = `.'.
# Shell code in configure.ac might set extrasub.
# FIXME: do we really want to maintain this feature?
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1
ac_sed_extra="$ac_vpsub
$extrasub
_ACEOF
cat >>$CONFIG_STATUS <<\_ACEOF || ac_write_fail=1

```

```

:t
/@[a-zA-Z_][a-zA-Z_0-9]*@/!b
s|@configure_input@|${ac_sed_conf_input}|;t t
s&@top_builddir@&${ac_top_builddir_sub}&;t t
s&@top_build_prefix@&${ac_top_build_prefix}&;t t
s&@srcdir@&${ac_srcdir}&;t t
s&@abs_srcdir@&${ac_abs_srcdir}&;t t
s&@top_srcdir@&${ac_top_srcdir}&;t t
s&@abs_top_srcdir@&${ac_abs_top_srcdir}&;t t
s&@builddir@&${ac_builddir}&;t t
s&@abs_builddir@&${ac_abs_builddir}&;t t
s&@abs_top_builddir@&${ac_abs_top_builddir}&;t t
s&@INSTALL@&${ac_INSTALL}&;t t
s&@MKDIR_P@&${ac_MKDIR_P}&;t t
${ac_datarootdir}_hack
"
eval sed \`${ac_sed_extra}\` "${ac_file_inputs}" | $AWK -f
"${ac_tmp}/subs.awk" \
>${ac_tmp}/out || as_fn_error $? "could not create $ac_file" "$LINENO"
5

test -z "${ac_datarootdir}_hack${ac_datarootdir}_seen" &&
{ ac_out=`sed -n '/\${datarootdir}/p' "${ac_tmp}/out"`; test -n
"${ac_out}"; } &&
{ ac_out=`sed -n '/^[ ]*datarootdir[ ]*:*=/p' \
"${ac_tmp}/out"`; test -z "${ac_out}"; } &&
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: $ac_file contains
a reference to the variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&5
$as_echo "$as_me: WARNING: $ac_file contains a reference to the
variable `datarootdir'
which seems to be undefined. Please make sure it is defined" >&2;}

rm -f "${ac_tmp}/stdin"
case $ac_file in
-) cat "${ac_tmp}/out" && rm -f "${ac_tmp}/out";;
*) rm -f "$ac_file" && mv "${ac_tmp}/out" "$ac_file";;
esac \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
;;
:H)
#
# CONFIG_HEADER
#
if test x"$ac_file" != x-; then
{
$as_echo "/* $configure_input */" \
&& eval '$AWK -f "${ac_tmp}/defines.awk"' "$ac_file_inputs"
} >"${ac_tmp}/config.h" \
|| as_fn_error $? "could not create $ac_file" "$LINENO" 5
if diff "$ac_file" "${ac_tmp}/config.h" >/dev/null 2>&1; then

```

```

        { $as_echo "$as_me:${as_lineno-$LINENO}: $ac_file is unchanged"
>&5
$as_echo "$as_me: $ac_file is unchanged" >&6;}
    else
        rm -f "$ac_file"
        mv "$ac_tmp/config.h" "$ac_file" \
        || as_fn_error $? "could not create $ac_file" "$LINENO" 5
    fi
else
    $as_echo "/* $configure_input */" \
    && eval '$AWK -f "$ac_tmp/defines.awk"' "$ac_file_inputs" \
    || as_fn_error $? "could not create -" "$LINENO" 5
fi
# Compute "$ac_file"'s index in $config_headers.
_am_arg="$ac_file"
_am_stamp_count=1
for _am_header in $config_headers ;; do
    case $_am_header in
        $_am_arg | $_am_arg:* )
            break ;;
        * )
            _am_stamp_count=`expr $_am_stamp_count + 1` ;;
    esac
done
echo "timestamp for $_am_arg" >`$as_dirname -- "$_am_arg" ||
$as_expr X"$_am_arg" : 'X\(.*[^/]\)\([^/]*\)[^/]*$' \|| \
X"$_am_arg" : 'X\(//\)[^/]' \|| \
X"$_am_arg" : 'X\(//\)$' \|| \
X"$_am_arg" : 'X\(//\)' \|| . 2>/dev/null ||
$as_echo X"$_am_arg" |
    sed '/^X\(.*[^/]\)\([^/]*\)[^/]*$/{
        s//\1/
        q
    }
/^X\(//\)[^/].*${
    s//\1/
    q
}
/^X\(//\)$/{
    s//\1/
    q
}
/^X\(//\).*${
    s//\1/
    q
}
s/.*/./; q'`/stamp-h$_am_stamp_count
;;

:C) { $as_echo "$as_me:${as_lineno-$LINENO}: executing $ac_file
commands" >&5
$as_echo "$as_me: executing $ac_file commands" >&6;}

```

```

;;
esac

case $ac_file$ac_mode in
  "depfiles":C) test x"$AMDEP_TRUE" != x"" || {
    # Autoconf 2.62 quotes --file arguments for eval, but not when files
    # are listed without --file. Let's play safe and only enable the
eval
    # if we detect the quoting.
    case $CONFIG_FILES in
      *\'*) eval set x "$CONFIG_FILES" ;;
      *) set x $CONFIG_FILES ;;
    esac
    shift
    for mf
    do
      # Strip MF so we end up with the name of the file.
      mf=`echo "$mf" | sed -e 's/:.*$//'\`
      # Check whether this is an Automake generated Makefile or not.
      # We used to match only the files named 'Makefile.in', but
      # some people rename them; so instead we look at the file content.
      # Grep'ing the first line is not enough: some people post-process
      # each Makefile.in and add a new line on top of each file to say
so.
      # Grep'ing the whole file is not good either: AIX grep has a line
      # limit of 2048, but all sed's we know have understand at least
4000.
      if sed -n 's,^#.*generated by automake.*,X,p' "$mf" | grep X
>/dev/null 2>&1; then
        dirpart=`$as_dirname -- "$mf" ||
$as_expr X"$mf" : 'X\(.*[^/]\)\([^/][^/]*/*$' \| \
  X"$mf" : 'X\(//\)[^/]' \| \
  X"$mf" : 'X\(//\)$' \| \
  X"$mf" : 'X\(/\)' \| . 2>/dev/null ||
$as_echo X"$mf" |
      sed '/^X\(.*[^/]\)\([^/][^/]*/*$/{
        s//\1/
        q
      }
/^X\(\\/\)\([^/]\)*/{
        s//\1/
        q
      }
/^X\(\\/\)\$/{
        s//\1/
        q
      }
/^X\(\\/\)*/{
        s//\1/
        q
      }
}'

```

```

        s/.*./.; q'`
else
    continue
fi
# Extract the definition of DEPDIR, am__include, and am__quote
# from the Makefile without running 'make'.
DEPDIR=`sed -n 's/^DEPDIR = //p' < "$mf"`
test -z "$DEPDIR" && continue
am__include=`sed -n 's/^am__include = //p' < "$mf"`
test -z "am__include" && continue
am__quote=`sed -n 's/^am__quote = //p' < "$mf"`
# Find all dependency output files, they are included files with
# $(DEPDIR) in their names. We invoke sed twice because it is the
# simplest approach to changing $(DEPDIR) to its actual value in
the
# expansion.
for file in `sed -n "
    s/^$am__include $am__quote\(.*(DEPDIR).*\) $am__quote" '$/\1/p'
<"$mf" | \
    sed -e 's/\$(DEPDIR)/"$DEPDIR"/g`; do
# Make sure the directory exists.
test -f "$dirpart/$file" && continue
fdir=`$as_dirname -- "$file" ||
$as_expr X"$file" : 'X\([^/]\)\/*\([^/]\)\/*$' \| \
    X"$file" : 'X\(/\)\[^/]' \| \
    X"$file" : 'X\(/\)\$' \| \
    X"$file" : 'X\(/)\' \| . 2>/dev/null ||
$as_echo X"$file" |
    sed '/^X\([^/]\)\*\/*\([^/]\)\*\/*$/{
        s//\1/
        q
    }
/^X\(\(\)\)\[^/].*/{
    s//\1/
    q
}
/^X\(\(\)\)\$/{
    s//\1/
    q
}
/^X\(\(\)\).*/{
    s//\1/
    q
}
'
        s/.*./.; q'`
as_dir=$dirpart/$fdir; as_fn_mkdir_p
# echo "creating $dirpart/$file"
echo '# dummy' > "$dirpart/$file"
done
done
}
;;

```

```

"libtool":C)

# See if we are running on zsh, and set the options which allow
our
# commands through without removal of \ escapes.
if test -n "${ZSH_VERSION+set}" ; then
    setopt NO_GLOB_SUBST
fi

cfgfile="${ofile}T"
trap "$RM \"${cfgfile}\"; exit 1" 1 2 15
$RM "${cfgfile}"

cat <<_LT_EOF >> "${cfgfile}"
#! $SHELL

# `SECHO "$ofile" | sed 's%^.*/%%'` - Provide generalized library-
building support services.
# Generated automatically by $as_me ($PACKAGE$TIMESTAMP) $VERSION
# Libtool was configured on host `(hostname || uname -n) 2>/dev/null |
sed lq`:
# NOTE: Changes made to this file will be lost: look at ltmain.sh.
#
# Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2003, 2004,
2005,
#           2006, 2007, 2008, 2009, 2010, 2011 Free Software
#           Foundation, Inc.
# Written by Gordon Matzigkeit, 1996
#
# This file is part of GNU Libtool.
#
# GNU Libtool is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License as
# published by the Free Software Foundation; either version 2 of
# the License, or (at your option) any later version.
#
# As a special exception to the GNU General Public License,
# if you distribute this file as part of a program or library that
# is built using GNU Libtool, you may include this file under the
# same distribution terms that you use for the rest of that program.
#
# GNU Libtool is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with GNU Libtool; see the file COPYING. If not, a copy
# can be downloaded from http://www.gnu.org/licenses/gpl.html, or
# obtained by writing to the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

```



```
# The names of the tagged configurations supported by this script.
available_tags="CXX RC "

# ### BEGIN LIBTOOL CONFIG

# Which release of libtool.m4 was used?
macro_version=$macro_version
macro_revision=$macro_revision

# Whether or not to build shared libraries.
build_libtool_libs=$enable_shared

# Whether or not to build static libraries.
build_old_libs=$enable_static

# What type of objects to build.
pic_mode=$pic_mode

# Whether or not to optimize for fast installation.
fast_install=$enable_fast_install

# Shell to use when invoking shell scripts.
SHELL=$lt_SHELL

# An echo program that protects backslashes.
ECHO=$lt_ECHO

# The PATH separator for the build system.
PATH_SEPARATOR=$lt_PATH_SEPARATOR

# The host system.
host_alias=$host_alias
host=$host
host_os=$host_os

# The build system.
build_alias=$build_alias
build=$build
build_os=$build_os

# A sed program that does not truncate output.
SED=$lt_SED

# Sed that helps us avoid accidentally triggering echo(1) options like
-n.
Xsed="\$SED -e 1s/^X/"

# A grep program that handles long lines.
GREP=$lt_GREP

# An ERE matcher.
```

```
EGREP=$lt_EGREP

# A literal string matcher.
FGREP=$lt_FGREP

# A BSD- or MS-compatible name lister.
NM=$lt_NM

# Whether we need soft or hard links.
LN_S=$lt_LN_S

# What is the maximum length of a command?
max_cmd_len=$max_cmd_len

# Object file suffix (normally "o").
objext=$ac_objext

# Executable file suffix (normally "").
exeext=$exeext

# whether the shell understands "unset".
lt_unset=$lt_unset

# turn spaces into newlines.
SP2NL=$lt_lt_SP2NL

# turn newlines into spaces.
NL2SP=$lt_lt_NL2SP

# convert \${build} file names to \${host} format.
to_host_file_cmd=$lt_cv_to_host_file_cmd

# convert \${build} files to toolchain format.
to_tool_file_cmd=$lt_cv_to_tool_file_cmd

# An object symbol dumper.
OBJDUMP=$lt_OBJDUMP

# Method to check whether dependent libraries are shared objects.
deplibs_check_method=$lt_deplibs_check_method

# Command to use when deplibs_check_method = "file_magic".
file_magic_cmd=$lt_file_magic_cmd

# How to find potential files when deplibs_check_method =
"file_magic".
file_magic_glob=$lt_file_magic_glob

# Find potential files using nocaseglob when deplibs_check_method =
"file_magic".
want_nocaseglob=$lt_want_nocaseglob
```

```
# DLL creation program.
DLLTOOL=$lt_DLLTOOL

# Command to associate shared and link libraries.
sharedlib_from_linklib_cmd=$lt_sharedlib_from_linklib_cmd

# The archiver.
AR=$lt_AR

# Flags to create an archive.
AR_FLAGS=$lt_AR_FLAGS

# How to feed a file listing to the archiver.
archiver_list_spec=$lt_archiver_list_spec

# A symbol stripping program.
STRIP=$lt_STRIP

# Commands used to install an old-style archive.
RANLIB=$lt_RANLIB
old_postinstall_cmds=$lt_old_postinstall_cmds
old_postuninstall_cmds=$lt_old_postuninstall_cmds

# Whether to use a lock for old archive extraction.
lock_old_archive_extraction=$lock_old_archive_extraction

# A C compiler.
LTCC=$lt_CC

# LTCC compiler flags.
LTCFLAGS=$lt_CFLAGS

# Take the output of nm and produce a listing of raw symbols and C
names.
global_symbol_pipe=$lt_lt_cv_sys_global_symbol_pipe

# Transform the output of nm in a proper C declaration.
global_symbol_to_cdecl=$lt_lt_cv_sys_global_symbol_to_cdecl

# Transform the output of nm in a C name address pair.
global_symbol_to_c_name_address=$lt_lt_cv_sys_global_symbol_to_c_name_
address

# Transform the output of nm in a C name address pair when lib prefix
is needed.
global_symbol_to_c_name_address_lib_prefix=$lt_lt_cv_sys_global_symbol
_to_c_name_address_lib_prefix

# Specify filename containing input files for \ $NM.
nm_file_list_spec=$lt_nm_file_list_spec
```

```
# The root where to search for dependent libraries, and in which our
libraries should be installed.
lt_sysroot=$lt_sysroot

# The name of the directory that contains temporary libtool files.
objdir=$objdir

# Used to examine libraries when file_magic_cmd begins with "file".
MAGIC_CMD=$MAGIC_CMD

# Must we lock files when doing compilation?
need_locks=$lt_need_locks

# Manifest tool.
MANIFEST_TOOL=$lt_MANIFEST_TOOL

# Tool to manipulate archived DWARF debug symbol files on Mac OS X.
DSYMUTIL=$lt_DSYMUTIL

# Tool to change global to local symbols on Mac OS X.
NMEDIT=$lt_NMEDIT

# Tool to manipulate fat objects and archives on Mac OS X.
LIPO=$lt_LIPO

# ldd/readelf like tool for Mach-O binaries on Mac OS X.
OTOOL=$lt_OTOOL

# ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4.
OTOOL64=$lt_OTOOL64

# Old archive suffix (normally "a").
libext=$libext

# Shared library suffix (normally ".so").
shrext_cmds=$lt_shrext_cmds

# The commands to extract the exported symbol list from a shared
archive.
extract_expsyms_cmds=$lt_extract_expsyms_cmds

# Variables whose values should be saved in libtool wrapper scripts
and
# restored at link time.
variables_saved_for_relink=$lt_variables_saved_for_relink

# Do we need the "lib" prefix for modules?
need_lib_prefix=$need_lib_prefix

# Do we need a version for libraries?
need_version=$need_version
```

```
# Library versioning type.
version_type=$version_type

# Shared library runtime path variable.
runpath_var=$runpath_var

# Shared library path variable.
shlibpath_var=$shlibpath_var

# Is shlibpath searched before the hard-coded library search path?
shlibpath_overrides_runpath=$shlibpath_overrides_runpath

# Format of library name prefix.
libname_spec=$lt_libname_spec

# List of archive names.  First name is the real one, the rest are
links.
# The last name is the one that the linker finds with -lNAME
library_names_spec=$lt_library_names_spec

# The coded name of the library, if different from the real name.
soname_spec=$lt_soname_spec

# Permission mode override for installation of shared libraries.
install_override_mode=$lt_install_override_mode

# Command to use after installation of a shared archive.
postinstall_cmds=$lt_postinstall_cmds

# Command to use after uninstallation of a shared archive.
postuninstall_cmds=$lt_postuninstall_cmds

# Commands used to finish a libtool library installation in a
directory.
finish_cmds=$lt_finish_cmds

# As "finish_cmds", except a single script fragment to be eval'd but
# not shown.
finish_eval=$lt_finish_eval

# Whether we should hardcode library paths into libraries.
hardcode_into_libs=$hardcode_into_libs

# Compile-time system search path for libraries.
sys_lib_search_path_spec=$lt_sys_lib_search_path_spec

# Run-time system search path for libraries.
sys_lib_dlsearch_path_spec=$lt_sys_lib_dlsearch_path_spec

# Whether dlopen is supported.
dlopen_support=$enable_dlopen
```

```
# Whether dlopen of programs is supported.
dlopen_self=$enable_dlopen_self

# Whether dlopen of statically linked programs is supported.
dlopen_self_static=$enable_dlopen_self_static

# Commands to strip libraries.
old_striplib=$lt_old_striplib
striplib=$lt_striplib

# The linker used to build libraries.
LD=$lt_LD

# How to create reloadable object files.
reload_flag=$lt_reload_flag
reload_cmds=$lt_reload_cmds

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds

# A language specific compiler.
CC=$lt_compiler

# Is the compiler the GNU compiler?
with_gcc=$GCC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec
```

```
# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds
archive_expsym_cmds=$lt_archive_expsym_cmds

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds
module_expsym_cmds=$lt_module_expsym_cmds

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute
```

```
# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs
```



```

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects
postdep_objects=$lt_postdep_objects
predeps=$lt_predeps
postdeps=$lt_postdeps

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path

# ### END LIBTOOL CONFIG

_LT_EOF

    case $host_os in
    aix3*)
        cat <<\_LT_EOF >> "$cfgfile"
        # AIX sometimes has problems with the GCC collect2 program.  For some
        # reason, if we set the COLLECT_NAMES environment variable, the
        # problems
        # vanish in a puff of smoke.
        if test "X${COLLECT_NAMES+set}" != Xset; then
            COLLECT_NAMES=
            export COLLECT_NAMES
        fi
    _LT_EOF
        ;;
    esac

ltmain="$ac_aux_dir/ltmain.sh"

# We use sed instead of cat because bash on DJGPP gets confused if
# it finds mixed CR/LF and LF-only lines.  Since sed operates in
# text mode, it properly converts lines to CR/LF.  This bash problem
# is reportedly fixed, but why not run on old versions too?
sed 'sq' "$ltmain" >> "$cfgfile" \
    || (rm -f "$cfgfile"; exit 1)

if test x"$xsi_shell" = xyes; then
    sed -e '/^func_dirname ()$/,/^{}/ # func_dirname /c\
func_dirname ()\
{\
\   case ${1} in\
\     */*) func_dirname_result="${1%/*}${2}" ;;\
\     * ) func_dirname_result="${3}" ;;\
\   esac\
} # Extended-shell func_dirname implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \

```

```

    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_basename ()$/,/^{ } # func_basename /c\
func_basename ()\
{\
\   func_basename_result="${1##*/}"\
} # Extended-shell func_basename implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_dirname_and_basename ()$/,/^{ } #
func_dirname_and_basename /c\
func_dirname_and_basename ()\
{\
\   case ${1} in\
\     */*) func_dirname_result="${1%/*}${2}" ;;\
\     * ) func_dirname_result="${3}" ;;\
\   esac\
\   func_basename_result="${1##*/}"\
} # Extended-shell func_dirname_and_basename implementation'
"$cfgfile" > $cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_stripname ()$/,/^{ } # func_stripname /c\
func_stripname ()\
{\
\   # pdksh 5.2.14 does not do ${X%$Y} correctly if both X and Y are\
\   # positional parameters, so assign one to ordinary parameter
first.\
\   func_stripname_result=${3}\
\   func_stripname_result=${func_stripname_result#"${1}"}\
\   func_stripname_result=${func_stripname_result%"${2}"}\
} # Extended-shell func_stripname implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_split_long_opt ()$/,/^{ } # func_split_long_opt /c\

```

```

func_split_long_opt ()\
{\
\   func_split_long_opt_name=${1%%=*}\
\   func_split_long_opt_arg=${1#*=}\
} # Extended-shell func_split_long_opt implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_split_short_opt ()$/,/^{ # func_split_short_opt /c\
func_split_short_opt ()\
{\
\   func_split_short_opt_arg=${1#??}\
\   func_split_short_opt_name=${1%"$func_split_short_opt_arg"}\
} # Extended-shell func_split_short_opt implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_lo2o ()$/,/^{ # func_lo2o /c\
func_lo2o ()\
{\
\   case ${1} in\
\     *.lo) func_lo2o_result=${1%.lo}.${objext} ;;\
\     *)    func_lo2o_result=${1} ;;\
\   esac\
} # Extended-shell func_lo2o implementation' "$cfgfile" > $cfgfile.tmp
\
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

  sed -e '/^func_xform ()$/,/^{ # func_xform /c\
func_xform ()\
{\
\   func_xform_result=${1%.*}.lo\
} # Extended-shell func_xform implementation' "$cfgfile" >
$cfgfile.tmp \
  && mv -f "$cfgfile.tmp" "$cfgfile" \
  || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

    sed -e '/^func_arith ()$/,/^{ # func_arith /c\
func_arith ()\
{\
    func_arith_result=$(( $* ))\
} # Extended-shell func_arith implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_len ()$/,/^{ # func_len /c\
func_len ()\
{\
    func_len_result=${#1}\
} # Extended-shell func_len implementation' "$cfgfile" > $cfgfile.tmp
\
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

fi

if test x"$lt_shell_append" = xyes; then
    sed -e '/^func_append ()$/,/^{ # func_append /c\
func_append ()\
{\
    eval "${1}+=\\${2}"\
} # Extended-shell func_append implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

    sed -e '/^func_append_quoted ()$/,/^{ # func_append_quoted /c\
func_append_quoted ()\
{\
    func_quote_for_eval "${2}"\
    eval "${1}+=\\\ \\$func_quote_for_eval_result"\
} # Extended-shell func_append_quoted implementation' "$cfgfile" >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:

```

```

# Save a `func_append' function call where possible by direct use of
'+='
sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1+= "%g' $cfgfile >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:
else
# Save a `func_append' function call even when '+=' is not available
sed -e 's%func_append \([a-zA-Z_]\{1,\}\) "%\1=" $\1%g' $cfgfile >
$cfgfile.tmp \
    && mv -f "$cfgfile.tmp" "$cfgfile" \
    || (rm -f "$cfgfile" && cp "$cfgfile.tmp" "$cfgfile" && rm -f
"$cfgfile.tmp")
test 0 -eq $? || _lt_function_replace_fail=:
fi

if test x"$_lt_function_replace_fail" = x:"; then
{ $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: Unable to
substitute extended shell functions in $ofile" >&5
$as_echo "$as_me: WARNING: Unable to substitute extended shell
functions in $ofile" >&2;}
fi

mv -f "$cfgfile" "$ofile" ||
(rm -f "$ofile" && cp "$cfgfile" "$ofile" && rm -f "$cfgfile")
chmod +x "$ofile"

cat <<_LT_EOF >> "$ofile"

# ### BEGIN LIBTOOL TAG CONFIG: CXX

# The linker used to build libraries.
LD=${lt_LD_CXX}

# How to create reloadable object files.
reload_flag=${lt_reload_flag_CXX}
reload_cmds=${lt_reload_cmds_CXX}

# Commands used to build an old-style archive.
old_archive_cmds=${lt_old_archive_cmds_CXX}

# A language specific compiler.
CC=${lt_compiler_CXX}

# Is the compiler the GNU compiler?
with_gcc=${GCC_CXX}

# Compiler flag to turn off builtin functions.

```

```
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag_CXX

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic_CXX

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl_CXX

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static_CXX

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o_CXX

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc_CXX

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes_CXX

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec_CXX

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec_CXX

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object_CXX

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds_CXX

# Create a temporary old-style archive to link instead of a shared
archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds_CXX

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds_CXX
archive_expsym_cmds=$lt_archive_expsym_cmds_CXX

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds_CXX
module_expsym_cmds=$lt_module_expsym_cmds_CXX

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld_CXX

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag_CXX
```

```
# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag_CXX

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec_CXX

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator_CXX

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct_CXX

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute_CXX

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L_CXX

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var_CXX

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic_CXX

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath_CXX

# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs_CXX

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols_CXX

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds_CXX
```

```

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms_CXX

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms_CXX

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds_CXX

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds_CXX

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec_CXX

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action_CXX

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs_CXX

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects_CXX
postdep_objects=$lt_postdep_objects_CXX
predeps=$lt_predeps_CXX
postdeps=$lt_postdeps_CXX

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path_CXX

# ### END LIBTOOL TAG CONFIG: CXX
_LT_EOF

    cat <<_LT_EOF >> "$ofile"

# ### BEGIN LIBTOOL TAG CONFIG: RC

# The linker used to build libraries.
LD=$lt_LD_RC

# How to create reloadable object files.
reload_flag=$lt_reload_flag_RC
reload_cmds=$lt_reload_cmds_RC

# Commands used to build an old-style archive.
old_archive_cmds=$lt_old_archive_cmds_RC

```



```
# A language specific compiler.
CC=$lt_compiler_RC

# Is the compiler the GNU compiler?
with_gcc=$GCC_RC

# Compiler flag to turn off builtin functions.
no_builtin_flag=$lt_lt_prog_compiler_no_builtin_flag_RC

# Additional compiler flags for building library objects.
pic_flag=$lt_lt_prog_compiler_pic_RC

# How to pass a linker flag through the compiler.
wl=$lt_lt_prog_compiler_wl_RC

# Compiler flag to prevent dynamic linking.
link_static_flag=$lt_lt_prog_compiler_static_RC

# Does compiler simultaneously support -c and -o options?
compiler_c_o=$lt_lt_cv_prog_compiler_c_o_RC

# Whether or not to add -lc for building shared libraries.
build_libtool_need_lc=$archive_cmds_need_lc_RC

# Whether or not to disallow shared libs when runtime libs are static.
allow_libtool_libs_with_static_runtimes=$enable_shared_with_static_runtimes_RC

# Compiler flag to allow reflexive dlopens.
export_dynamic_flag_spec=$lt_export_dynamic_flag_spec_RC

# Compiler flag to generate shared objects directly from archives.
whole_archive_flag_spec=$lt_whole_archive_flag_spec_RC

# Whether the compiler copes with passing no objects directly.
compiler_needs_object=$lt_compiler_needs_object_RC

# Create an old-style archive from a shared archive.
old_archive_from_new_cmds=$lt_old_archive_from_new_cmds_RC

# Create a temporary old-style archive to link instead of a shared archive.
old_archive_from_expsyms_cmds=$lt_old_archive_from_expsyms_cmds_RC

# Commands used to build a shared archive.
archive_cmds=$lt_archive_cmds_RC
archive_expsym_cmds=$lt_archive_expsym_cmds_RC

# Commands used to build a loadable module if different from building
# a shared archive.
module_cmds=$lt_module_cmds_RC
```

```
module_expsym_cmds=$lt_module_expsym_cmds_RC

# Whether we are building with GNU ld or not.
with_gnu_ld=$lt_with_gnu_ld_RC

# Flag that allows shared libraries with undefined symbols to be
built.
allow_undefined_flag=$lt_allow_undefined_flag_RC

# Flag that enforces no undefined symbols.
no_undefined_flag=$lt_no_undefined_flag_RC

# Flag to hardcode \${libdir} into a binary during linking.
# This must work even if \${libdir} does not exist
hardcode_libdir_flag_spec=$lt_hardcode_libdir_flag_spec_RC

# Whether we need a single "-rpath" flag with a separated argument.
hardcode_libdir_separator=$lt_hardcode_libdir_separator_RC

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary.
hardcode_direct=$hardcode_direct_RC

# Set to "yes" if using DIR/libNAME\${shared_ext} during linking
hardcodes
# DIR into the resulting binary and the resulting library dependency
is
# "absolute", i.e impossible to change by setting \${shlibpath_var} if
the
# library is relocated.
hardcode_direct_absolute=$hardcode_direct_absolute_RC

# Set to "yes" if using the -LDIR flag during linking hardcodes DIR
# into the resulting binary.
hardcode_minus_L=$hardcode_minus_L_RC

# Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
# into the resulting binary.
hardcode_shlibpath_var=$hardcode_shlibpath_var_RC

# Set to "yes" if building a shared library automatically hardcodes
DIR
# into the library and all subsequent libraries and executables linked
# against it.
hardcode_automatic=$hardcode_automatic_RC

# Set to yes if linker adds runtime paths of dependent libraries
# to runtime path list.
inherit_rpath=$inherit_rpath_RC
```

```
# Whether libtool must link a program against all its dependency
libraries.
link_all_deplibs=$link_all_deplibs_RC

# Set to "yes" if exported symbols are required.
always_export_symbols=$always_export_symbols_RC

# The commands to list exported symbols.
export_symbols_cmds=$lt_export_symbols_cmds_RC

# Symbols that should not be listed in the preloaded symbols.
exclude_expsyms=$lt_exclude_expsyms_RC

# Symbols that must always be exported.
include_expsyms=$lt_include_expsyms_RC

# Commands necessary for linking programs (against libraries) with
templates.
prelink_cmds=$lt_prelink_cmds_RC

# Commands necessary for finishing linking programs.
postlink_cmds=$lt_postlink_cmds_RC

# Specify filename containing input files.
file_list_spec=$lt_file_list_spec_RC

# How to hardcode a shared library path into an executable.
hardcode_action=$hardcode_action_RC

# The directories searched by this compiler when creating a shared
library.
compiler_lib_search_dirs=$lt_compiler_lib_search_dirs_RC

# Dependencies to place before and after the objects being linked to
# create a shared library.
predep_objects=$lt_predep_objects_RC
postdep_objects=$lt_postdep_objects_RC
predeps=$lt_predeps_RC
postdeps=$lt_postdeps_RC

# The library search path used internally by the compiler when linking
# a shared library.
compiler_lib_search_path=$lt_compiler_lib_search_path_RC

# ### END LIBTOOL TAG CONFIG: RC
_LT_EOF

;;

esac
done # for ac_tag
```

```

as_fn_exit 0
_ACEOF
ac_clean_files=$ac_clean_files_save

test $ac_write_fail = 0 ||
  as_fn_error $? "write failure creating $CONFIG_STATUS" "$LINENO" 5

# configure is writing to config.log, and then calls config.status.
# config.status does its own redirection, appending to config.log.
# Unfortunately, on DOS this fails, as config.log is still kept open
# by configure, so config.status won't be able to write to it; its
# output is simply discarded.  So we exec the FD to /dev/null,
# effectively closing config.log, so it can be properly (re)opened and
# appended to by config.status.  When coming back to configure, we
# need to make the FD available again.
if test "$no_create" != yes; then
  ac_cs_success=:
  ac_config_status_args=
  test "$silent" = yes &&
    ac_config_status_args="$ac_config_status_args --quiet"
  exec 5>/dev/null
  $SHELL $CONFIG_STATUS $ac_config_status_args || ac_cs_success=false
  exec 5>>config.log
  # Use ||, not &&, to avoid exiting from the if with $? = 1, which
  # would make configure fail if this is the last instruction.
  $ac_cs_success || as_fn_exit 1
fi
if test -n "$ac_unrecognized_opts" && test "$enable_option_checking"
!= no; then
  { $as_echo "$as_me:${as_lineno-$LINENO}: WARNING: unrecognized
options: $ac_unrecognized_opts" >&5
$as_echo "$as_me: WARNING: unrecognized options:
$ac_unrecognized_opts" >&2;}
fi

echo "
          D-Bus $VERSION
          =====

prefix:                ${EXPANDED_PREFIX}
exec_prefix:           ${exec_prefix}
  libdir:               ${EXPANDED_LIBDIR}
  libexecdir:           ${EXPANDED_LIBEXECDIR}
  bindir:               ${EXPANDED_BINDIR}
  sysconfdir:           ${EXPANDED_SYSCONFDIR}
  localstatedir:        ${EXPANDED_LOCALSTATEDIR}
datadir:               ${EXPANDED_DATADIR}
source code location:  ${srcdir}
compiler:              ${CC}

```

```

cflags:                ${CFLAGS}
cppflags:              ${CPPFLAGS}
cxxflags:              ${CXXFLAGS}
64-bit int:           ${DBUS_INT64_TYPE}
32-bit int:           ${DBUS_INT32_TYPE}
16-bit int:           ${DBUS_INT16_TYPE}
  Doxygen:            ${DOXYGEN:-not found}
  xmlto:              ${XMLTO:-not found}
  man2html:           ${MAN2HTML:-not found}"

echo "
Rebuilding generated files: ${USE_MAINTAINER_MODE}
gcc coverage profiling:    ${enable_compiler_coverage}
Building embedded tests:  ${enable_embedded_tests}
Building modular tests:   ${enable_modular_tests}
  - with GLib:            ${with_glib}
Building verbose mode:    ${enable_verbose_mode}
Building assertions:      ${enable_asserts}
Building checks:          ${enable_checks}
Building bus stats API:   ${enable_stats}
Building SELinux support: ${have_selinux}
Building inotify support: ${have_inotify}
Building dnotify support: ${have_dnotify}
Building kqueue support:  ${have_kqueue}
Building systemd support: ${have_systemd}
Building X11 code:        ${enable_x11}
Building Doxygen docs:    ${enable_doxygen_docs}
Building XML docs:        ${enable_xml_docs}
Building cache support:   ${enable_userdb_cache}
Building launchd support: ${have_launchd}
Using XML parser:         ${with_xml}
Init scripts style:      ${with_init_scripts}
Abstract socket names:    ${ac_cv_have_abstract_sockets}
System bus socket:        ${DBUS_SYSTEM_SOCKET}
System bus address:       ${DBUS_SYSTEM_BUS_DEFAULT_ADDRESS}
System bus PID file:      ${DBUS_SYSTEM_PID_FILE}
Session bus address:      ${DBUS_SESSION_BUS_DEFAULT_ADDRESS}
Console auth dir:         ${DBUS_CONSOLE_AUTH_DIR}
Console owner file:       ${have_console_owner_file}
Console owner file path:  ${DBUS_CONSOLE_OWNER_FILE}
System bus user:          ${DBUS_USER}
Session bus services dir: ${EXPANDED_DATADIR}/dbus-1/services
  'make check' socket dir: ${TEST_SOCKET_DIR}
"
if test x$have_launchd = xyes; then
  echo "      launchd agent dir:      ${LAUNCHD_AGENT_DIR}"
fi
echo

if test x$enable_embedded_tests = xyes; then
  echo "NOTE: building with unit tests increases the size of the
installed library and renders it insecure."

```

```

fi
if test x$enable_embedded_tests = xyes -a x$enable_asserts = xno; then
    echo "NOTE: building with embedded tests but without
assertions means tests may not properly report failures (this
configuration is only useful when doing something like profiling the
tests)"
fi
if test x$enable_compiler_coverage = xyes; then
    echo "NOTE: building with coverage profiling is definitely for
developers only."
fi
if test x$enable_verbose_mode = xyes; then
    echo "NOTE: building with verbose mode increases library size,
may slightly increase security risk, and decreases performance."
fi
if test x$enable_asserts = xyes; then
    echo "NOTE: building with assertions increases library size
and decreases performance."
fi
if test x$enable_checks = xno; then
    echo "NOTE: building without checks for arguments passed to
public API makes it harder to debug apps using D-Bus, but will
slightly decrease D-Bus library size and _very_ slightly improve
performance."
fi
if test x$dbus_use_libxml = xtrue; then
    echo
    echo "WARNING: You have chosen to use libxml as your xml parser
however this code path is not maintained by the D-Bus developers and
if it breaks you get to keep the pieces.  If you have selected this
option in err please reconfigure with expat (e.g. --with-xml=expat)."

```

This removes parts of the standard D-Bus API and ABI (the 't' and 'x'

typecodes, the dbus_int64_t and dbus_uint64_t types, etc.) and should only be

used if your compiler lacks support for 64-bit integers. Please report a bug

with details of your platform and compiler.

This option is likely to be removed in future, unless the D-Bus developers

receive reports that it is still needed.

```
" >&2;}
```

```
fi
```

```
File = ConfigureChecks.cmake
```

```
include(CheckIncludeFile)
include(CheckSymbolExists)
include(CheckStructMember)
include(CheckTypeSize)
```

```
check_include_file(dirent.h      HAVE_DIRENT_H) # dbus-sysdeps-util.c
check_include_file(io.h          HAVE_IO_H)   # internal
check_include_file(grp.h         HAVE_GRP_H)   # dbus-sysdeps-util-
win.c
check_include_file(sys/poll.h    HAVE_POLL)   # dbus-sysdeps.c,
dbus-sysdeps-win.c
check_include_file(sys/time.h   HAVE_SYS_TIME_H) # dbus-sysdeps-win.c
check_include_file(sys/wait.h   HAVE_SYS_WAIT_H) # dbus-sysdeps-win.c
check_include_file(time.h       HAVE_TIME_H)   # dbus-sysdeps-win.c
check_include_file(ws2tcpip.h   HAVE_WS2TCPIP_H) # dbus-sysdeps-win.c
check_include_file(wspiapi.h    HAVE_WSPIAPI_H) # dbus-sysdeps-win.c
check_include_file(unistd.h     HAVE_UNISTD_H) # dbus-sysdeps-util-
win.c
check_include_file(stdio.h       HAVE_STDIO_H)  # dbus-sysdeps.h
check_include_file(sys/syslimits.h HAVE_SYS_SYSLIMITS_H) # dbus-
sysdeps-unix.c
check_include_file(errno.h      HAVE_ERRNO_H)  # dbus-sysdeps.c
check_include_file(signal.h     HAVE_SIGNAL_H)
check_include_file(locale.h     HAVE_LOCALE_H)
check_include_file(inttypes.h   HAVE_INTTYPES_H) # dbus-pipe.h
check_include_file(stdint.h     HAVE_STDINT_H)  # dbus-pipe.h

check_symbol_exists(backtrace  "execinfo.h"    HAVE_BACKTRACE)
# dbus-sysdeps.c, dbus-sysdeps-win.c
check_symbol_exists(getgrouplist "grp.h"      HAVE_GETGROUPLIST)
# dbus-sysdeps.c
check_symbol_exists(getpeerucrd "ucrd.h"     HAVE_GETPEERUCRED)
# dbus-sysdeps.c, dbus-sysdeps-win.c
```

```

check_symbol_exists(nanosleep      "time.h"          HAVE_NANOSLEEP)
# dbus-sysdeps.c
check_symbol_exists(getpwnam_r     "errno.h pwd.h"
HAVE_POSIX_GETPWNAM_R) # dbus-sysdeps-util-unix.c
check_symbol_exists(setenv         "stdlib.h"       HAVE_SETENV)
# dbus-sysdeps.c
check_symbol_exists(unsetenv       "stdlib.h"       HAVE_UNSETENV)
# dbus-sysdeps.c
check_symbol_exists(clearenv       "stdlib.h"       HAVE_CLEARENV)
# dbus-sysdeps.c
check_symbol_exists(writev         "sys/uio.h"      HAVE_WRITEV)
# dbus-sysdeps.c, dbus-sysdeps-win.c
check_symbol_exists(setrlimit      "sys/resource.h" HAVE_SETRLIMIT)
# dbus-sysdeps.c, dbus-sysdeps-win.c, test/test-segfault.c
check_symbol_exists(socketpair    "sys/socket.h"  HAVE_SOCKETPAIR)
# dbus-sysdeps.c
check_symbol_exists(socklen_t     "sys/socket.h"  HAVE_SOCKETLEN_T)
# dbus-sysdeps-unix.c
check_symbol_exists(setlocale     "locale.h"      HAVE_SETLOCALE)
# dbus-test-main.c
check_symbol_exists(localeconv    "locale.h"      HAVE_LOCALECONV)
# dbus-sysdeps.c
check_symbol_exists(strtoll       "stdlib.h"      HAVE_STRTOULL)
# dbus-send.c
check_symbol_exists(strtoll       "stdlib.h"      HAVE_STRTOULL)
# dbus-send.c

check_struct_member(msgcred cmcred_pid "sys/types.h sys/socket.h"
HAVE_MSGCRED) # dbus-sysdeps.c

# missing:
# HAVE_ABSTRACT_SOCKETS
# DBUS_HAVE_GCC33_GCOV

check_type_size("short"          SIZEOF_SHORT)
check_type_size("int"           SIZEOF_INT)
check_type_size("long"          SIZEOF_LONG)
check_type_size("long long"     SIZEOF_LONG_LONG)
check_type_size("__int64"       SIZEOF__INT64)

# DBUS_INT64_TYPE
if(SIZEOF_INT EQUAL 8)
    set (DBUS_HAVE_INT64 1)
    set (DBUS_INT64_TYPE "int")
    set (DBUS_INT64_CONSTANT "(val)")
    set (DBUS_UINT64_CONSTANT "(val#U)")
elseif(SIZEOF_LONG EQUAL 8)
    set (DBUS_HAVE_INT64 1)
    set (DBUS_INT64_TYPE "long")
    set (DBUS_INT64_CONSTANT "(val##L)")
    set (DBUS_UINT64_CONSTANT "(val##UL)")
elseif(SIZEOF_LONG_LONG EQUAL 8)

```



```

    set (DBUS_HAVE_INT64 1)
    set (DBUS_INT64_TYPE "long long")
    set (DBUS_INT64_CONSTANT "(val##LL)")
    set (DBUS_UINT64_CONSTANT "(val##ULL)")
elseif(SIZEOF__INT64 EQUAL 8)
    set (DBUS_HAVE_INT64 1)
    set (DBUS_INT64_TYPE "__int64")
    set (DBUS_INT64_CONSTANT "(val##i64)")
    set (DBUS_UINT64_CONSTANT "(val##ui64)")
endif(SIZEOF_INT EQUAL 8)

# DBUS_INT32_TYPE
if(SIZEOF_INT EQUAL 4)
    set (DBUS_INT32_TYPE "int")
elseif(SIZEOF_LONG EQUAL 4)
    set (DBUS_INT32_TYPE "long")
elseif(SIZEOF_LONG_LONG EQUAL 4)
    set (DBUS_INT32_TYPE "long long")
endif(SIZEOF_INT EQUAL 4)

# DBUS_INT16_TYPE
if(SIZEOF_INT EQUAL 2)
    set (DBUS_INT16_TYPE "int")
elseif(SIZEOF_SHORT EQUAL 2)
    set (DBUS_INT16_TYPE "short")
endif(SIZEOF_INT EQUAL 2)

find_program(DOXYGEN doxygen)
find_program(XMLTO xmlto)

if(MSVC)
    SET(DBUS_VA_COPY_FUNC "_DBUS_VA_COPY_ASSIGN";)
else(MSVC)
write_file("${CMAKE_BINARY_DIR}${CMAKE_FILES_DIRECTORY}/CMakeTmp/cmake
_try_compile.c" "#include <stdarg.h>
#include <stdlib.h>
    static void f (int i, ...) {
        va_list args1, args2;
        va_start (args1, i);
        va_copy (args2, args1);
        if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
            exit (1);
        va_end (args1); va_end (args2);
    }
int main() {
    f (0, 42);
    return 0;
}
")
try_compile(DBUS_HAVE_VA_COPY
    ${CMAKE_BINARY_DIR}

```

```

${CMAKE_BINARY_DIR}${CMAKE_FILES_DIRECTORY}/CMakeTmp/cmake_try_compile
.c)

if(DBUS_HAVE_VA_COPY)
    SET(DBUS_VA_COPY_FUNC va_copy CACHE STRING "va_copy function")
else(DBUS_HAVE_VA_COPY)

write_file("${CMAKE_BINARY_DIR}${CMAKE_FILES_DIRECTORY}/CMakeTmp/cmake
_try_compile.c" "#include <stdarg.h>
#include <stdlib.h>
static void f (int i, ...) {
va_list args1, args2;
va_start (args1, i);
__va_copy (args2, args1);
if (va_arg (args2, int) != 42 || va_arg (args1, int) != 42)
    exit (1);
va_end (args1); va_end (args2);
}
int main() {
    f (0, 42);
    return 0;
}
")
try_compile(DBUS_HAVE_VA_COPY
    ${CMAKE_BINARY_DIR}

${CMAKE_BINARY_DIR}${CMAKE_FILES_DIRECTORY}/CMakeTmp/cmake_try_compile
.c)
if(DBUS_HAVE_VA_COPY)
    SET(DBUS_VA_COPY_FUNC __va_copy CACHE STRING "va_copy function")
else(DBUS_HAVE_VA_COPY)
    SET(DBUS_VA_COPY_AS_ARRAY "1" CACHE STRING "'va_lists' cannot be
copies as values")
endif(DBUS_HAVE_VA_COPY)
endif(DBUS_HAVE_VA_COPY)
endif(MSVC) # _not_ MSVC
#### Abstract sockets

if (DBUS_ENABLE_ABSTRACT_SOCKETS)

    try_compile(HAVE_ABSTRACT_SOCKETS
        ${CMAKE_BINARY_DIR}
        ${CMAKE_SOURCE_DIR}/modules/CheckForAbstractSockets.c)

endif(DBUS_ENABLE_ABSTRACT_SOCKETS)

if(HAVE_ABSTRACT_SOCKETS)
    set(DBUS_PATH_OR_ABSTRACT_VALUE abstract)
else(HAVE_ABSTRACT_SOCKETS)
    set(DBUS_PATH_OR_ABSTRACT_VALUE path)
endif(HAVE_ABSTRACT_SOCKETS)

```

```

File = connection.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* connection.c Client connections
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#include "connection.h"
#include "dispatch.h"
#include "policy.h"
#include "services.h"
#include "utils.h"
#include "signals.h"
#include "expirelist.h"
#include "selinux.h"
#include <dbus/dbus-list.h>
#include <dbus/dbus-hash.h>
#include <dbus/dbus-timeout.h>

/* Trim executed commands to this length; we want to keep logs
readable */
#define MAX_LOG_COMMAND_LEN 50

static void bus_connection_remove_transactions (DBusConnection
*connection);

```

```

typedef struct
{
    BusExpireItem expire_item;

    DBusConnection *will_get_reply;
    DBusConnection *will_send_reply;

    dbus_uint32_t reply_serial;
} BusPendingReply;

struct BusConnections
{
    int refcount;
    DBusList *completed; /**< List of all completed connections */
    int n_completed; /**< Length of completed list */
    DBusList *incomplete; /**< List of all not-yet-active connections */
    int n_incomplete; /**< Length of incomplete list */
    BusContext *context;
    DBusHashTable *completed_by_user; /**< Number of completed
connections for each UID */
    DBusTimeout *expire_timeout; /**< Timeout for expiring incomplete
connections. */
    int stamp; /**< Incrementing number */
    BusExpireList *pending_replies; /**< List of pending replies */

#ifdef DBUS_ENABLE_STATS
    int total_match_rules;
    int peak_match_rules;
    int peak_match_rules_per_conn;

    int total_bus_names;
    int peak_bus_names;
    int peak_bus_names_per_conn;
#endif
};

static dbus_int32_t connection_data_slot = -1;

typedef struct
{
    BusConnections *connections;
    DBusList *link_in_connection_list;
    DBusConnection *connection;
    DBusList *services_owned;
    int n_services_owned;
    DBusList *match_rules;
    int n_match_rules;
    char *name;
    DBusList *transaction_messages; /**< Stuff we need to send as part
of a transaction */
    DBusMessage *oom_message;
}

```

```

DBusPreallocatedSend *oom_preallocated;
BusClientPolicy *policy;

char *cached_logininfo_string;
BusSELinuxID *selinux_id;

long connection_tv_sec; /**< Time when we connected (seconds
component) */
long connection_tv_usec; /**< Time when we connected (microsec
component) */
int stamp; /**< connections->stamp last time we were
traversed */

#ifdef DBUS_ENABLE_STATS
int peak_match_rules;
int peak_bus_names;
#endif
} BusConnectionData;

static dbus_bool_t bus_pending_reply_expired (BusExpireList *list,
                                              DBusList *link,
                                              void *data);

static void bus_connection_drop_pending_replies (BusConnections
*connections,
                                              DBusConnection
*connection);

static dbus_bool_t expire_incomplete_timeout (void *data);

#define BUS_CONNECTION_DATA(connection) (dbus_connection_get_data
((connection), connection_data_slot))

static DBusLoop*
connection_get_loop (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    return bus_context_get_loop (d->connections->context);
}

static int
get_connections_for_uid (BusConnections *connections,
                        dbus_uid_t uid)
{
    void *val;
    int current_count;

    /* val is NULL is 0 when it isn't in the hash yet */

```

```

    val = _dbus_hash_table_lookup_uintptr (connections-
>completed_by_user,
                                         uid);

    current_count = _DBUS_POINTER_TO_INT (val);

    return current_count;
}

static dbus_bool_t
adjust_connections_for_uid (BusConnections *connections,
                            dbus_uid_t     uid,
                            int            adjustment)
{
    int current_count;

    current_count = get_connections_for_uid (connections, uid);

    _dbus_verbose ("Adjusting connection count for UID " DBUS_UID_FORMAT
                  ": was %d adjustment %d making %d\n",
                  uid, current_count, adjustment, current_count +
adjustment);

    _dbus_assert (current_count >= 0);

    current_count += adjustment;

    _dbus_assert (current_count >= 0);

    if (current_count == 0)
    {
        _dbus_hash_table_remove_uintptr (connections->completed_by_user,
uid);
        return TRUE;
    }
    else
    {
        dbus_bool_t retval;

        retval = _dbus_hash_table_insert_uintptr (connections-
>completed_by_user,
                                                  uid,
        _DBUS_INT_TO_POINTER (current_count));

        /* only positive adjustment can fail as otherwise
         * a hash entry should already exist
         */
        _dbus_assert (adjustment > 0 ||
                    (adjustment <= 0 && retval));

        return retval;
    }
}

```

```

    }
}

void
bus_connection_disconnected (DBusConnection *connection)
{
    BusConnectionData *d;
    BusService *service;
    BusMatchmaker *matchmaker;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    _dbus_verbose ("%s disconnected, dropping all service ownership and
releasing\n",
                   d->name ? d->name : "(inactive)");

    /* Delete our match rules */
    if (d->n_match_rules > 0)
    {
        matchmaker = bus_context_get_matchmaker (d->connections-
>context);
        bus_matchmaker_disconnected (matchmaker, connection);
    }

    /* Drop any service ownership. Unfortunately, this requires
    * memory allocation and there doesn't seem to be a good way to
    * handle it other than sleeping; we can't "fail" the operation of
    * disconnecting a client, and preallocating a broadcast "service is
    * now gone" message for every client-service pair seems kind of
    * involved.
    */
    while ((service = _dbus_list_get_last (&d->services_owned)))
    {
        BusTransaction *transaction;
        DBusError error;

        retry:

            dbus_error_init (&error);

            while ((transaction = bus_transaction_new (d->connections-
>context)) == NULL)
                _dbus_wait_for_memory ();

            if (!bus_service_remove_owner (service, connection,
                                           transaction, &error))
            {
                _DBUS_ASSERT_ERROR_IS_SET (&error);

                if (dbus_error_has_name (&error, DBUS_ERROR_NO_MEMORY))
                {

```

```

        dbus_error_free (&error);
        bus_transaction_cancel_and_free (transaction);
        _dbus_wait_for_memory ();
        goto retry;
    }
    else
    {
        _dbus_verbose ("Failed to remove service owner: %s
%s\n",
                        error.name, error.message);
        _dbus_assert_not_reached ("Removing service owner failed
for non-memory-related reason");
    }
}

    bus_transaction_execute_and_free (transaction);
}

bus_dispatch_remove_connection (connection);

/* no more watching */
if (!dbus_connection_set_watch_functions (connection,
                                         NULL, NULL, NULL,
                                         connection,
                                         NULL))
    _dbus_assert_not_reached ("setting watch functions to NULL
failed");

if (!dbus_connection_set_timeout_functions (connection,
                                           NULL, NULL, NULL,
                                           connection,
                                           NULL))
    _dbus_assert_not_reached ("setting timeout functions to NULL
failed");

dbus_connection_set_unix_user_function (connection,
                                       NULL, NULL, NULL);
dbus_connection_set_windows_user_function (connection,
                                           NULL, NULL, NULL);

dbus_connection_set_dispatch_status_function (connection,
                                             NULL, NULL, NULL);

bus_connection_remove_transactions (connection);

if (d->link_in_connection_list != NULL)
{
    if (d->name != NULL)
    {
        unsigned long uid;

```



```

        _dbus_list_remove_link (&d->connections->completed, d-
>link_in_connection_list);
        d->link_in_connection_list = NULL;
        d->connections->n_completed -= 1;

        if (dbus_connection_get_unix_user (connection, &uid))
        {
            if (!adjust_connections_for_uid (d->connections,
                                              uid, -1))
                _dbus_assert_not_reached ("adjusting downward should
never fail");
        }
        else
        {
            _dbus_list_remove_link (&d->connections->incomplete, d-
>link_in_connection_list);
            d->link_in_connection_list = NULL;
            d->connections->n_incomplete -= 1;
        }

        _dbus_assert (d->connections->n_incomplete >= 0);
        _dbus_assert (d->connections->n_completed >= 0);
    }

    bus_connection_drop_pending_replies (d->connections, connection);

    /* frees "d" as side effect */
    dbus_connection_set_data (connection,
                             connection_data_slot,
                             NULL, NULL);

    dbus_connection_unref (connection);
}

static dbus_bool_t
add_connection_watch (DBusWatch      *watch,
                    void             *data)
{
    DBusConnection *connection = data;

    return _dbus_loop_add_watch (connection_get_loop (connection),
watch);
}

static void
remove_connection_watch (DBusWatch      *watch,
                    void             *data)
{
    DBusConnection *connection = data;

    _dbus_loop_remove_watch (connection_get_loop (connection), watch);
}

```

```

}

static void
toggle_connection_watch (DBusWatch      *watch,
                        void             *data)
{
    DBusConnection *connection = data;

    _dbus_loop_toggle_watch (connection_get_loop (connection), watch);
}

static dbus_bool_t
add_connection_timeout (DBusTimeout      *timeout,
                      void               *data)
{
    DBusConnection *connection = data;

    return _dbus_loop_add_timeout (connection_get_loop (connection),
    timeout);
}

static void
remove_connection_timeout (DBusTimeout      *timeout,
                          void              *data)
{
    DBusConnection *connection = data;

    _dbus_loop_remove_timeout (connection_get_loop (connection),
    timeout);
}

static void
dispatch_status_function (DBusConnection *connection,
                        DBusDispatchStatus new_status,
                        void               *data)
{
    DBusLoop *loop = data;

    if (new_status != DBUS_DISPATCH_COMPLETE)
    {
        while (!_dbus_loop_queue_dispatch (loop, connection))
            _dbus_wait_for_memory ();
    }
}

static dbus_bool_t
allow_unix_user_function (DBusConnection *connection,
                        unsigned long    uid,
                        void              *data)
{
    BusConnectionData *d;

```

```

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return bus_context_allow_unix_user (d->connections->context, uid);
}

static void
free_connection_data (void *data)
{
    BusConnectionData *d = data;

    /* services_owned should be NULL since we should be disconnected */
    _dbus_assert (d->services_owned == NULL);
    _dbus_assert (d->n_services_owned == 0);
    /* similarly */
    _dbus_assert (d->transaction_messages == NULL);

    if (d->oom_preallocated)
        dbus_connection_free_preallocated_send (d->connection, d-
>oom_preallocated);

    if (d->oom_message)
        dbus_message_unref (d->oom_message);

    if (d->policy)
        bus_client_policy_unref (d->policy);

    if (d->selinux_id)
        bus_selinux_id_unref (d->selinux_id);

    dbus_free (d->cached_logininfo_string);

    dbus_free (d->name);

    dbus_free (d);
}

BusConnections*
bus_connections_new (BusContext *context)
{
    BusConnections *connections;

    if (!dbus_connection_allocate_data_slot (&connection_data_slot))
        goto failed_0;

    connections = dbus_new0 (BusConnections, 1);
    if (connections == NULL)
        goto failed_1;

    connections->completed_by_user = _dbus_hash_table_new
(DBUS_HASH_UINTPTR,

```

```

NULL, NULL);
if (connections->completed_by_user == NULL)
    goto failed_2;

connections->expire_timeout = _dbus_timeout_new (100, /* irrelevant
*/
expire_incomplete_timeout,
connections, NULL);
if (connections->expire_timeout == NULL)
    goto failed_3;

_dbus_timeout_set_enabled (connections->expire_timeout, FALSE);

connections->pending_replies = bus_expire_list_new
(bus_context_get_loop (context),
bus_context_get_reply_timeout (context),
bus_pending_reply_expired,
connections);
if (connections->pending_replies == NULL)
    goto failed_4;

if (!_dbus_loop_add_timeout (bus_context_get_loop (context),
connections->expire_timeout))
    goto failed_5;

connections->refcount = 1;
connections->context = context;

return connections;

failed_5:
bus_expire_list_free (connections->pending_replies);
failed_4:
_dbus_timeout_unref (connections->expire_timeout);
failed_3:
_dbus_hash_table_unref (connections->completed_by_user);
failed_2:
dbus_free (connections);
failed_1:
dbus_connection_free_data_slot (&connection_data_slot);
failed_0:
return NULL;
}

BusConnections *
bus_connections_ref (BusConnections *connections)
{
    _dbus_assert (connections->refcount > 0);
    connections->refcount += 1;

```

```

    return connections;
}

void
bus_connections_unref (BusConnections *connections)
{
    _dbus_assert (connections->refcount > 0);
    connections->refcount -= 1;
    if (connections->refcount == 0)
    {
        /* drop all incomplete */
        while (connections->incomplete != NULL)
        {
            DBusConnection *connection;

            connection = connections->incomplete->data;

            dbus_connection_ref (connection);
            dbus_connection_close (connection);
            bus_connection_disconnected (connection);
            dbus_connection_unref (connection);
        }

        _dbus_assert (connections->n_incomplete == 0);

        /* drop all real connections */
        while (connections->completed != NULL)
        {
            DBusConnection *connection;

            connection = connections->completed->data;

            dbus_connection_ref (connection);
            dbus_connection_close (connection);
            bus_connection_disconnected (connection);
            dbus_connection_unref (connection);
        }

        _dbus_assert (connections->n_completed == 0);

        bus_expire_list_free (connections->pending_replies);

        _dbus_loop_remove_timeout (bus_context_get_loop (connections->context),
                                   connections->expire_timeout);

        _dbus_timeout_unref (connections->expire_timeout);
        _dbus_hash_table_unref (connections->completed_by_user);

        dbus_free (connections);
    }
}

```

```

        dbus_connection_free_data_slot (&connection_data_slot);
    }
}

/* Used for logging */
static dbus_bool_t
cache_peer_logininfo_string (BusConnectionData *d,
                             DBusConnection *connection)
{
    DBusString logininfo_buf;
    unsigned long uid;
    unsigned long pid;
    char *windows_sid;
    dbus_bool_t prev_added;

    if (!_dbus_string_init (&logininfo_buf))
        return FALSE;

    prev_added = FALSE;
    if (dbus_connection_get_unix_user (connection, &uid))
    {
        if (!_dbus_string_append_printf (&logininfo_buf, "uid=%ld", uid))
            goto oom;
        else
            prev_added = TRUE;
    }

    if (dbus_connection_get_unix_process_id (connection, &pid))
    {
        if (prev_added)
        {
            if (!_dbus_string_append_byte (&logininfo_buf, ' '))
                goto oom;
        }
        if (!_dbus_string_append_printf (&logininfo_buf, "pid=%ld
comm=\"", pid))
            goto oom;
        /* Ignore errors here; we may not have permissions to read the
        * proc file. */
        _dbus_command_for_pid (pid, &logininfo_buf, MAX_LOG_COMMAND_LEN,
NULL);
        if (!_dbus_string_append_byte (&logininfo_buf, '"'))
            goto oom;
    }

    if (dbus_connection_get_windows_user (connection, &windows_sid))
    {
        dbus_bool_t did_append;
        did_append = _dbus_string_append_printf (&logininfo_buf,
"sid=\"%s\" ",
windows_sid);
    }
}

```

```

        dbus_free (windows_sid);
        if (!did_append)
            goto oom;
    }

    if (!_dbus_string_steal_data (&loginfo_buf, &(d-
>cached_logininfo_string)))
        goto oom;

    _dbus_string_free (&loginfo_buf);

    return TRUE;
oom:
    _dbus_string_free (&loginfo_buf);
    return FALSE;
}

dbus_bool_t
bus_connections_setup_connection (BusConnections *connections,
                                  DBusConnection *connection)
{
    BusConnectionData *d;
    dbus_bool_t retval;
    DBusError error;

    d = dbus_new0 (BusConnectionData, 1);

    if (d == NULL)
        return FALSE;

    d->connections = connections;
    d->connection = connection;

    _dbus_get_monotonic_time (&d->connection_tv_sec,
                              &d->connection_tv_usec);

    _dbus_assert (connection_data_slot >= 0);

    if (!dbus_connection_set_data (connection,
                                   connection_data_slot,
                                   d, free_connection_data))
    {
        dbus_free (d);
        return FALSE;
    }

    dbus_connection_set_route_peer_messages (connection, TRUE);

    retval = FALSE;
}

```

```

dbus_error_init (&error);
d->selinux_id = bus_selinux_init_connection_id (connection,
                                                &error);

if (dbus_error_is_set (&error))
{
    /* This is a bit bogus because we pretend all errors
     * are OOM; this is done because we know that in bus.c
     * an OOM error disconnects the connection, which is
     * the same thing we want on any other error.
     */
    dbus_error_free (&error);
    goto out;
}

if (!dbus_connection_set_watch_functions (connection,
                                         add_connection_watch,
                                         remove_connection_watch,
                                         toggle_connection_watch,
                                         connection,
                                         NULL))

    goto out;

if (!dbus_connection_set_timeout_functions (connection,
                                           add_connection_timeout,
remove_connection_timeout,
                                           NULL,
                                           connection, NULL))

    goto out;

/* For now we don't need to set a Windows user function because
 * there are no policies in the config file controlling what
 * Windows users can connect. The default 'same user that owns the
 * bus can connect' behavior of DbusConnection is fine on Windows.
 */
dbus_connection_set_unix_user_function (connection,
                                       allow_unix_user_function,
                                       NULL, NULL);

dbus_connection_set_dispatch_status_function (connection,
dispatch_status_function,
                                       bus_context_get_loop
(connection->context),
                                       NULL);

d->link_in_connection_list = _dbus_list_alloc_link (connection);
if (d->link_in_connection_list == NULL)
    goto out;

/* Setup the connection with the dispatcher */
if (!bus_dispatch_add_connection (connection))

```



```

goto out;

if (dbus_connection_get_dispatch_status (connection) !=
DBUS_DISPATCH_COMPLETE)
{
    if (!_dbus_loop_queue_dispatch (bus_context_get_loop
(connections->context), connection))
        {
            bus_dispatch_remove_connection (connection);
            goto out;
        }
}

_dbus_list_append_link (&connections->incomplete, d-
>link_in_connection_list);
connections->n_incomplete += 1;

dbus_connection_ref (connection);

/* Note that we might disconnect ourselves here, but it only takes
* effect on return to the main loop. We call this to free up
* expired connections if possible, and to queue the timeout for our
* own expiration.
*/
bus_connections_expire_incomplete (connections);

/* And we might also disconnect ourselves here, but again it
* only takes effect on return to main loop.
*/
if (connections->n_incomplete >
    bus_context_get_max_incomplete_connections (connections-
>context))
{
    _dbus_verbose ("Number of incomplete connections exceeds max,
dropping oldest one\n");

    _dbus_assert (connections->incomplete != NULL);
    /* Disconnect the oldest unauthenticated connection.  FIXME
    * would it be more secure to drop a *random* connection?  This
    * algorithm seems to mean that if someone can create new
    * connections quickly enough, they can keep anyone else from
    * completing authentication.  But random may or may not really
    * help with that, a more elaborate solution might be required.
    */
    dbus_connection_close (connections->incomplete->data);
}

retval = TRUE;

out:
if (!retval)
{

```

```

    if (d->selinux_id)
        bus_selinux_id_unref (d->selinux_id);
    d->selinux_id = NULL;

    if (!dbus_connection_set_watch_functions (connection,
                                             NULL, NULL, NULL,
                                             connection,
                                             NULL))
        _dbus_assert_not_reached ("setting watch functions to NULL
failed");

    if (!dbus_connection_set_timeout_functions (connection,
                                              NULL, NULL, NULL,
                                              connection,
                                              NULL))
        _dbus_assert_not_reached ("setting timeout functions to NULL
failed");

    dbus_connection_set_unix_user_function (connection,
                                           NULL, NULL, NULL);

    dbus_connection_set_windows_user_function (connection,
                                              NULL, NULL, NULL);

    dbus_connection_set_dispatch_status_function (connection,
                                                NULL, NULL, NULL);

    if (d->link_in_connection_list != NULL)
    {
        _dbus_assert (d->link_in_connection_list->next == NULL);
        _dbus_assert (d->link_in_connection_list->prev == NULL);
        _dbus_list_free_link (d->link_in_connection_list);
        d->link_in_connection_list = NULL;
    }

    if (!dbus_connection_set_data (connection,
                                  connection_data_slot,
                                  NULL, NULL))
        _dbus_assert_not_reached ("failed to set connection data to
null");

    /* "d" has now been freed */
}

return retval;
}

void
bus_connections_expire_incomplete (BusConnections *connections)
{
    int next_interval;

```

```

next_interval = -1;

if (connections->incomplete != NULL)
{
    long tv_sec, tv_usec;
    DBusList *link;
    int auth_timeout;

    _dbus_get_monotonic_time (&tv_sec, &tv_usec);
    auth_timeout = bus_context_get_auth_timeout (connections-
>context);

    link = _dbus_list_get_first_link (&connections->incomplete);
    while (link != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (&connections-
>incomplete, link);
        DBusConnection *connection;
        BusConnectionData *d;
        double elapsed;

        connection = link->data;

        d = BUS_CONNECTION_DATA (connection);

        _dbus_assert (d != NULL);

        elapsed = ELAPSED_MILLISECONDS SINCE (d->connection_tv_sec,
                                                d->connection_tv_usec,
                                                tv_sec, tv_usec);

        if (elapsed >= (double) auth_timeout)
        {
            _dbus_verbose ("Timing out authentication for connection
%p\n", connection);
            dbus_connection_close (connection);
        }
        else
        {
            /* We can end the loop, since the connections are in
oldest-first order */
            next_interval = ((double)auth_timeout) - elapsed;
            _dbus_verbose ("Connection %p authentication expires in
%d milliseconds\n",
                            connection, next_interval);

            break;
        }

        link = next;
    }
}

```

```

    bus_expire_timeout_set_interval (connections->expire_timeout,
                                    next_interval);
}

static dbus_bool_t
expire_incomplete_timeout (void *data)
{
    BusConnections *connections = data;

    _dbus_verbose ("Running\n");

    /* note that this may remove the timeout */
    bus_connections_expire_incomplete (connections);

    return TRUE;
}

dbus_bool_t
bus_connection_get_unix_groups (DBusConnection *connection,
                                unsigned long **groups,
                                int *n_groups,
                                DBusError *error)
{
    unsigned long uid;

    *groups = NULL;
    *n_groups = 0;

    if (dbus_connection_get_unix_user (connection, &uid))
    {
        if (!_dbus_unix_groups_from_uid (uid, groups, n_groups))
        {
            _dbus_verbose ("Did not get any groups for UID %lu\n",
                            uid);
            return FALSE;
        }
        else
        {
            _dbus_verbose ("Got %d groups for UID %lu\n",
                            *n_groups, uid);
            return TRUE;
        }
    }
    else
        return TRUE; /* successfully got 0 groups */
}

dbus_bool_t
bus_connection_is_in_unix_group (DBusConnection *connection,
                                unsigned long gid)
{

```

```

int i;
unsigned long *group_ids;
int n_group_ids;

if (!bus_connection_get_unix_groups (connection, &group_ids,
&n_group_ids,
                                NULL))
    return FALSE;

i = 0;
while (i < n_group_ids)
{
    if (group_ids[i] == gid)
    {
        dbus_free (group_ids);
        return TRUE;
    }
    ++i;
}

dbus_free (group_ids);
return FALSE;
}

const char *
bus_connection_get_logininfo (DBusConnection      *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    if (!bus_connection_is_active (connection))
        return "inactive";
    return d->cached_logininfo_string;
}

BusClientPolicy*
bus_connection_get_policy (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);
    _dbus_assert (d->policy != NULL);

    return d->policy;
}

static dbus_bool_t
foreach_active (BusConnections      *connections,
                BusConnectionForeachFunction function,

```

```

        void                                *data)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&connections->completed);
    while (link != NULL)
    {
        DBusConnection *connection = link->data;
        DBusList *next = _dbus_list_get_next_link (&connections-
>completed, link);

        if (!(* function) (connection, data))
            return FALSE;

        link = next;
    }

    return TRUE;
}

static dbus_bool_t
foreach_inactive (BusConnections            *connections,
                 BusConnectionForeachFunction function,
                 void                      *data)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&connections->incomplete);
    while (link != NULL)
    {
        DBusConnection *connection = link->data;
        DBusList *next = _dbus_list_get_next_link (&connections-
>incomplete, link);

        if (!(* function) (connection, data))
            return FALSE;

        link = next;
    }

    return TRUE;
}

/**
 * Calls function on each active connection; if the function returns
 * #FALSE, stops iterating. Active connections are authenticated
 * and have sent a Hello message.
 *
 * @param connections the connections object
 * @param function the function
 * @param data data to pass to it as a second arg
 */

```

```

void
bus_connections_foreach_active (BusConnections
*connections,
                                BusConnectionForeachFunction
function,
                                void
                                *data)
{
    foreach_active (connections, function, data);
}

/**
 * Calls function on each connection; if the function returns
 * #FALSE, stops iterating.
 *
 * @param connections the connections object
 * @param function the function
 * @param data data to pass to it as a second arg
 */
void
bus_connections_foreach (BusConnections
                        BusConnectionForeachFunction
                        void
                        *connections,
                        function,
                        *data)
{
    if (!foreach_active (connections, function, data))
        return;

    foreach_inactive (connections, function, data);
}

BusContext*
bus_connections_get_context (BusConnections *connections)
{
    return connections->context;
}

/*
 * This is used to avoid covering the same connection twice when
 * traversing connections. Note that it assumes we will
 * bus_connection_mark_stamp() each connection at least once per
 * INT_MAX increments of the global stamp, or wraparound would break
 * things.
 */
void
bus_connections_increment_stamp (BusConnections *connections)
{
    connections->stamp += 1;
}

/* Mark connection with current stamp, return TRUE if it
 * didn't already have that stamp
 */
dbus_bool_t

```

```

bus_connection_mark_stamp (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    if (d->stamp == d->connections->stamp)
        return FALSE;
    else
    {
        d->stamp = d->connections->stamp;
        return TRUE;
    }
}

BusContext*
bus_connection_get_context (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return d->connections->context;
}

BusConnections*
bus_connection_get_connections (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return d->connections;
}

BusRegistry*
bus_connection_get_registry (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return bus_context_get_registry (d->connections->context);
}

```



```

BusActivation*
bus_connection_get_activation (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return bus_context_get_activation (d->connections->context);
}

BusMatchmaker*
bus_connection_get_matchmaker (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return bus_context_get_matchmaker (d->connections->context);
}

BusSELinuxID*
bus_connection_get_selinux_id (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    return d->selinux_id;
}

/**
 * Checks whether the connection is registered with the message bus.
 *
 * @param connection the connection
 * @returns #TRUE if we're an active message bus participant
 */
dbus_bool_t
bus_connection_is_active (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    return d != NULL && d->name != NULL;
}

```

```

dbus_bool_t
bus_connection_preallocate_oom_error (DBusConnection *connection)
{
    DBusMessage *message;
    DBusPreallocatedSend *preallocated;
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);

    if (d->oom_preallocated != NULL)
        return TRUE;

    preallocated = dbus_connection_preallocate_send (connection);
    if (preallocated == NULL)
        return FALSE;

    message = dbus_message_new (DBUS_MESSAGE_TYPE_ERROR);

    if (message == NULL)
    {
        dbus_connection_free_preallocated_send (connection,
preallocated);
        return FALSE;
    }

    /* d->name may be NULL, but that is OK */
    if (!dbus_message_set_error_name (message, DBUS_ERROR_NO_MEMORY) ||
        !dbus_message_set_destination (message, d->name) ||
        !dbus_message_set_sender (message,
                                DBUS_SERVICE_DBUS))
    {
        dbus_connection_free_preallocated_send (connection,
preallocated);
        dbus_message_unref (message);
        return FALSE;
    }

    /* set reply serial to placeholder value just so space is already
    allocated
    * for it.
    */
    if (!dbus_message_set_reply_serial (message, 14))
    {
        dbus_connection_free_preallocated_send (connection,
preallocated);
        dbus_message_unref (message);
        return FALSE;
    }
}

```

```

    d->oom_message = message;
    d->oom_preallocated = preallocated;

    return TRUE;
}

void
bus_connection_send_oom_error (DBusConnection *connection,
                              DBusMessage    *in_reply_to)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);

    _dbus_assert (d != NULL);
    _dbus_assert (d->oom_message != NULL);

    /* should always succeed since we set it to a placeholder earlier */
    if (!dbus_message_set_reply_serial (d->oom_message,
                                        dbus_message_get_serial
(in_reply_to)))
        _dbus_assert_not_reached ("Failed to set reply serial for
preallocated oom message");

    _dbus_assert (dbus_message_get_sender (d->oom_message) != NULL);

    dbus_connection_send_preallocated (connection, d->oom_preallocated,
                                       d->oom_message, NULL);

    dbus_message_unref (d->oom_message);
    d->oom_message = NULL;
    d->oom_preallocated = NULL;
}

#ifdef DBUS_ENABLE_STATS
static void
update_peak (int *peak,
             int n)
{
    if (*peak < n)
        *peak = n;
}
#endif

void
bus_connection_add_match_rule_link (DBusConnection *connection,
                                    DBusList       *link)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

```

```

    _dbus_list_append_link (&d->match_rules, link);

    d->n_match_rules += 1;

#ifdef DBUS_ENABLE_STATS
    update_peak (&d->peak_match_rules, d->n_match_rules);
    update_peak (&d->connections->peak_match_rules_per_conn, d->n_match_rules);

    d->connections->total_match_rules += 1;
    update_peak (&d->connections->peak_match_rules,
                d->connections->total_match_rules);
#endif
}

dbus_bool_t
bus_connection_add_match_rule (DBusConnection *connection,
                               BusMatchRule  *rule)
{
    DBusList *link;

    link = _dbus_list_alloc_link (rule);

    if (link == NULL)
        return FALSE;

    bus_connection_add_match_rule_link (connection, link);

    return TRUE;
}

void
bus_connection_remove_match_rule (DBusConnection *connection,
                                  BusMatchRule  *rule)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    _dbus_list_remove_last (&d->match_rules, rule);

    d->n_match_rules -= 1;
    _dbus_assert (d->n_match_rules >= 0);

#ifdef DBUS_ENABLE_STATS
    d->connections->total_match_rules -= 1;
#endif
}

int

```

```

bus_connection_get_n_match_rules (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    return d->n_match_rules;
}

void
bus_connection_add_owned_service_link (DBusConnection *connection,
                                       DBusList      *link)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    _dbus_list_append_link (&d->services_owned, link);

    d->n_services_owned += 1;

#ifdef DBUS_ENABLE_STATS
    update_peak (&d->peak_bus_names, d->n_services_owned);
    update_peak (&d->connections->peak_bus_names_per_conn,
                 d->n_services_owned);

    d->connections->total_bus_names += 1;
    update_peak (&d->connections->peak_bus_names,
                 d->connections->total_bus_names);
#endif
}

dbus_bool_t
bus_connection_add_owned_service (DBusConnection *connection,
                                  BusService     *service)
{
    DBusList *link;

    link = _dbus_list_alloc_link (service);

    if (link == NULL)
        return FALSE;

    bus_connection_add_owned_service_link (connection, link);

    return TRUE;
}

void
bus_connection_remove_owned_service (DBusConnection *connection,

```

```

BusService      *service)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    _dbus_list_remove_last (&d->services_owned, service);

    d->n_services_owned -= 1;
    _dbus_assert (d->n_services_owned >= 0);

#ifdef DBUS_ENABLE_STATS
    d->connections->total_bus_names -= 1;
#endif
}

int
bus_connection_get_n_services_owned (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    return d->n_services_owned;
}

dbus_bool_t
bus_connection_complete (DBusConnection      *connection,
                        const DBusString *name,
                        DBusError          *error)
{
    BusConnectionData *d;
    unsigned long uid;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);
    _dbus_assert (d->name == NULL);
    _dbus_assert (d->policy == NULL);

    _dbus_assert (!bus_connection_is_active (connection));

    if (!_dbus_string_copy_data (name, &d->name))
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    _dbus_assert (d->name != NULL);

    _dbus_verbose ("Name %s assigned to %p\n", d->name, connection);
}

```

```

    d->policy = bus_context_create_client_policy (d->connections-
>context,
                                                    connection,
                                                    error);

/* we may have a NULL policy on OOM or error getting list of
 * groups for a user. In the latter case we don't handle it so
 * well currently, as it will just keep failing over and over.
 */

if (d->policy == NULL)
{
    _dbus_verbose ("Failed to create security policy for connection
%p\n",
                  connection);
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_free (d->name);
    d->name = NULL;
    return FALSE;
}

if (dbus_connection_get_unix_user (connection, &uid))
{
    if (!adjust_connections_for_uid (d->connections,
                                    uid, 1))
        goto fail;
}

/* Create and cache a string which holds information about the
 * peer process; used for logging purposes.
 */
if (!cache_peer_logininfo_string (d, connection))
    goto fail;

/* Now the connection is active, move it between lists */
_dbus_list_unlink (&d->connections->incomplete,
                  d->link_in_connection_list);
d->connections->n_incomplete -= 1;
_dbus_list_append_link (&d->connections->completed,
                       d->link_in_connection_list);
d->connections->n_completed += 1;

_dbus_assert (d->connections->n_incomplete >= 0);
_dbus_assert (d->connections->n_completed > 0);

/* See if we can remove the timeout */
bus_connections_expire_incomplete (d->connections);

_dbus_assert (bus_connection_is_active (connection));

return TRUE;

```

```

fail:
    BUS_SET_OOM (error);
    dbus_free (d->name);
    d->name = NULL;
    if (d->policy)
        bus_client_policy_unref (d->policy);
    d->policy = NULL;
    return FALSE;
}

const char *
bus_connection_get_name (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    return d->name;
}

/**
 * Check whether completing the passed-in connection would
 * exceed limits, and if so set error and return #FALSE
 */
dbus_bool_t
bus_connections_check_limits (BusConnections *connections,
                             DBusConnection *requesting_completion,
                             DBusError *error)
{
    unsigned long uid;

    if (connections->n_completed >=
        bus_context_get_max_completed_connections (connections->context))
    {
        dbus_set_error (error, DBUS_ERROR_LIMITS_EXCEEDED,
                       "The maximum number of active connections has
been reached");
        return FALSE;
    }

    if (dbus_connection_get_unix_user (requesting_completion, &uid))
    {
        if (get_connections_for_uid (connections, uid) >=
            bus_context_get_max_connections_per_user (connections->context))
        {
            dbus_set_error (error, DBUS_ERROR_LIMITS_EXCEEDED,
                           "The maximum number of active connections
for UID %lu has been reached",
                           uid);

```



```

        return FALSE;
    }
}

return TRUE;
}

static void
bus_pending_reply_free (BusPendingReply *pending)
{
    _dbus_verbose ("Freeing pending reply %p, replier %p receiver %p
serial %u\n",
                  pending,
                  pending->will_send_reply,
                  pending->will_get_reply,
                  pending->reply_serial);

    dbus_free (pending);
}

static dbus_bool_t
bus_pending_reply_send_no_reply (BusConnections *connections,
                                BusTransaction *transaction,
                                BusPendingReply *pending)
{
    DBusMessage *message;
    DBusMessageIter iter;
    dbus_bool_t retval;
    const char *errmsg;

    retval = FALSE;

    message = dbus_message_new (DBUS_MESSAGE_TYPE_ERROR);
    if (message == NULL)
        return FALSE;

    dbus_message_set_no_reply (message, TRUE);

    if (!dbus_message_set_reply_serial (message,
                                       pending->reply_serial))
        goto out;

    if (!dbus_message_set_error_name (message,
                                     DBUS_ERROR_NO_REPLY))
        goto out;

    errmsg = "Message did not receive a reply (timeout by message bus)";
    dbus_message_iter_init_append (message, &iter);
    if (!dbus_message_iter_append_basic (&iter, DBUS_TYPE_STRING,
&errmsg))
        goto out;
}

```

```

    if (!bus_transaction_send_from_driver (transaction, pending-
>will_get_reply,
                                         message))
        goto out;

    retval = TRUE;

out:
    dbus_message_unref (message);
    return retval;
}

static dbus_bool_t
bus_pending_reply_expired (BusExpireList *list,
                           DBusList      *link,
                           void          *data)
{
    BusPendingReply *pending = link->data;
    BusConnections *connections = data;
    BusTransaction *transaction;

    /* No reply is forthcoming. So nuke it if we can. If not,
     * leave it in the list to try expiring again later when we
     * get more memory.
     */

    _dbus_verbose ("Expiring pending reply %p, replier %p receiver %p
serial %u\n",
                  pending,
                  pending->will_send_reply,
                  pending->will_get_reply,
                  pending->reply_serial);

    transaction = bus_transaction_new (connections->context);
    if (transaction == NULL)
        return FALSE;

    if (!bus_pending_reply_send_no_reply (connections,
                                         transaction,
                                         pending))
    {
        bus_transaction_cancel_and_free (transaction);
        return FALSE;
    }

    bus_expire_list_remove_link (connections->pending_replies, link);

    bus_pending_reply_free (pending);
    bus_transaction_execute_and_free (transaction);

    return TRUE;
}

```

```

static void
bus_connection_drop_pending_replies (BusConnections *connections,
                                     DBusConnection *connection)
{
    /* The DBusConnection is almost 100% finalized here, so you can't
     * do anything with it except check for pointer equality
     */
    DBusList *link;

    _dbus_verbose ("Dropping pending replies that involve connection
%p\n",
                  connection);

    link = bus_expire_list_get_first_link (connections-
>pending_replies);
    while (link != NULL)
    {
        DBusList *next;
        BusPendingReply *pending;

        next = bus_expire_list_get_next_link (connections-
>pending_replies,
                                             link);

        pending = link->data;

        if (pending->will_get_reply == connection)
        {
            /* We don't need to track this pending reply anymore */

            _dbus_verbose ("Dropping pending reply %p, replier %p
receiver %p serial %u\n",
                          pending,
                          pending->will_send_reply,
                          pending->will_get_reply,
                          pending->reply_serial);

            bus_expire_list_remove_link (connections->pending_replies,
                                        link);
            bus_pending_reply_free (pending);
        }
        else if (pending->will_send_reply == connection)
        {
            /* The reply isn't going to be sent, so set things
             * up so it will be expired right away
             */
            _dbus_verbose ("Will expire pending reply %p, replier %p
receiver %p serial %u\n",
                          pending,
                          pending->will_send_reply,
                          pending->will_get_reply,
                          pending->reply_serial);
        }
    }
}

```

```

        pending->will_send_reply = NULL;
        pending->expire_item.added_tv_sec = 0;
        pending->expire_item.added_tv_usec = 0;

        bus_expire_list_recheck_immediately (connections-
>pending_replies);
    }

    link = next;
}

typedef struct
{
    BusPendingReply *pending;
    BusConnections *connections;
} CancelPendingReplyData;

static void
cancel_pending_reply (void *data)
{
    CancelPendingReplyData *d = data;

    _dbus_verbose ("d = %p\n", d);

    if (!bus_expire_list_remove (d->connections->pending_replies,
                                &d->pending->expire_item))
        _dbus_assert_not_reached ("pending reply did not exist to be
cancelled");

    bus_pending_reply_free (d->pending); /* since it's been cancelled */
}

static void
cancel_pending_reply_data_free (void *data)
{
    CancelPendingReplyData *d = data;

    _dbus_verbose ("d = %p\n", d);

    /* d->pending should be either freed or still
     * in the list of pending replies (owned by someone
     * else)
     */

    dbus_free (d);
}

/*
 * Record that a reply is allowed; return TRUE on success.

```

```

*/
dbus_bool_t
bus_connections_expect_reply (BusConnections *connections,
                               BusTransaction *transaction,
                               DBusConnection *will_get_reply,
                               DBusConnection *will_send_reply,
                               DBusMessage *reply_to_this,
                               DBusError *error)
{
    BusPendingReply *pending;
    dbus_uint32_t reply_serial;
    DBusList *link;
    CancelPendingReplyData *cprd;
    int count;

    _dbus_assert (will_get_reply != NULL);
    _dbus_assert (will_send_reply != NULL);
    _dbus_assert (reply_to_this != NULL);

    if (dbus_message_get_no_reply (reply_to_this))
        return TRUE; /* we won't allow a reply, since client doesn't care
for one. */

    reply_serial = dbus_message_get_serial (reply_to_this);

    link = bus_expire_list_get_first_link (connections-
>pending_replies);
    count = 0;
    while (link != NULL)
    {
        pending = link->data;

        if (pending->reply_serial == reply_serial &&
            pending->will_get_reply == will_get_reply &&
            pending->will_send_reply == will_send_reply)
        {
            dbus_set_error (error, DBUS_ERROR_ACCESS_DENIED,
                "Message has the same reply serial as a
currently-outstanding existing method call");
            return FALSE;
        }

        link = bus_expire_list_get_next_link (connections-
>pending_replies,
                                                link);
        if (pending->will_get_reply == will_get_reply)
            ++count;
    }

    if (count >=
        bus_context_get_max_replies_per_connection (connections-
>context))

```

```

    {
        dbus_set_error (error, DBUS_ERROR_LIMITS_EXCEEDED,
            "The maximum number of pending replies per connection
has been reached");
        return FALSE;
    }

    pending = dbus_new0 (BusPendingReply, 1);
    if (pending == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

#ifdef DBUS_ENABLE_VERBOSE_MODE
    /* so we can see a not-yet-added pending reply */
    pending->expire_item.added_tv_sec = 1;
    pending->expire_item.added_tv_usec = 1;
#endif

    pending->will_get_reply = will_get_reply;
    pending->will_send_reply = will_send_reply;
    pending->reply_serial = reply_serial;

    cprd = dbus_new0 (CancelPendingReplyData, 1);
    if (cprd == NULL)
    {
        BUS_SET_OOM (error);
        bus_pending_reply_free (pending);
        return FALSE;
    }

    if (!bus_expire_list_add (connections->pending_replies,
        &pending->expire_item))
    {
        BUS_SET_OOM (error);
        dbus_free (cprd);
        bus_pending_reply_free (pending);
        return FALSE;
    }

    if (!bus_transaction_add_cancel_hook (transaction,
        cancel_pending_reply,
        cprd,
cancel_pending_reply_data_free))
    {
        BUS_SET_OOM (error);
        bus_expire_list_remove (connections->pending_replies, &pending-
>expire_item);
        dbus_free (cprd);
        bus_pending_reply_free (pending);
    }

```

```

        return FALSE;
    }

    cprd->pending = pending;
    cprd->connections = connections;

    _dbus_get_monotonic_time (&pending->expire_item.added_tv_sec,
                              &pending->expire_item.added_tv_usec);

    _dbus_verbose ("Added pending reply %p, replier %p receiver %p
serial %u\n",
                  pending,
                  pending->will_send_reply,
                  pending->will_get_reply,
                  pending->reply_serial);

    return TRUE;
}

typedef struct
{
    DBusList          *link;
    BusConnections    *connections;
} CheckPendingReplyData;

static void
cancel_check_pending_reply (void *data)
{
    CheckPendingReplyData *d = data;

    _dbus_verbose ("d = %p\n",d);

    bus_expire_list_add_link (d->connections->pending_replies,
                              d->link);

    d->link = NULL;
}

static void
check_pending_reply_data_free (void *data)
{
    CheckPendingReplyData *d = data;

    _dbus_verbose ("d = %p\n",d);

    if (d->link != NULL)
    {
        BusPendingReply *pending = d->link->data;

        _dbus_assert (!bus_expire_list_contains_item (d->connections-
>pending_replies,
                                                    &pending-
>expire_item));
    }
}

```

```

        bus_pending_reply_free (pending);
        _dbus_list_free_link (d->link);
    }

    dbus_free (d);
}

/*
 * Check whether a reply is allowed, remove BusPendingReply
 * if so, return TRUE if so.
 */
dbus_bool_t
bus_connections_check_reply (BusConnections *connections,
                             BusTransaction *transaction,
                             DBusConnection *sending_reply,
                             DBusConnection *receiving_reply,
                             DBusMessage *reply,
                             DBusError *error)
{
    CheckPendingReplyData *cprd;
    DBusList *link;
    dbus_uint32_t reply_serial;

    _dbus_assert (sending_reply != NULL);
    _dbus_assert (receiving_reply != NULL);

    reply_serial = dbus_message_get_reply_serial (reply);

    link = bus_expire_list_get_first_link (connections-
>pending_replies);
    while (link != NULL)
    {
        BusPendingReply *pending = link->data;

        if (pending->reply_serial == reply_serial &&
            pending->will_get_reply == receiving_reply &&
            pending->will_send_reply == sending_reply)
        {
            _dbus_verbose ("Found pending reply with serial %u\n",
reply_serial);
            break;
        }

        link = bus_expire_list_get_next_link (connections-
>pending_replies,
                                             link);
    }

    if (link == NULL)
    {
        _dbus_verbose ("No pending reply expected\n");
    }
}

```



```

        return FALSE;
    }

    cprd = dbus_new0 (CheckPendingReplyData, 1);
    if (cprd == NULL)
    {
        BUS_SET_OOM (error);
        return FALSE;
    }

    if (!bus_transaction_add_cancel_hook (transaction,
                                         cancel_check_pending_reply,
                                         cprd,
                                         check_pending_reply_data_free))
    {
        BUS_SET_OOM (error);
        dbus_free (cprd);
        return FALSE;
    }

    cprd->link = link;
    cprd->connections = connections;

    bus_expire_list_unlink (connections->pending_replies,
                           link);

    _dbus_assert (!bus_expire_list_contains_item (connections->
                                                  pending_replies, link->data));

    return TRUE;
}

/*
 * Transactions
 *
 * Note that this is fairly fragile; in particular, don't try to use
 * one transaction across any main loop iterations.
 */

typedef struct
{
    BusTransaction *transaction;
    DBusMessage     *message;
    DBusPreallocatedSend *preallocated;
} MessageToSend;

typedef struct
{
    BusTransactionCancelFunction cancel_function;
    DBusFreeFunction free_data_function;
}

```

```

    void *data;
} CancelHook;

struct BusTransaction
{
    DBusList *connections;
    BusContext *context;
    DBusList *cancel_hooks;
};

static void
message_to_send_free (DBusConnection *connection,
                     MessageToSend *to_send)
{
    if (to_send->message)
        dbus_message_unref (to_send->message);

    if (to_send->preallocated)
        dbus_connection_free_preallocated_send (connection, to_send->preallocated);

    dbus_free (to_send);
}

static void
cancel_hook_cancel (void *element,
                   void *data)
{
    CancelHook *ch = element;

    _dbus_verbose ("Running transaction cancel hook\n");

    if (ch->cancel_function)
        (* ch->cancel_function) (ch->data);
}

static void
cancel_hook_free (void *element,
                 void *data)
{
    CancelHook *ch = element;

    if (ch->free_data_function)
        (* ch->free_data_function) (ch->data);

    dbus_free (ch);
}

static void
free_cancel_hooks (BusTransaction *transaction)
{
    _dbus_list_foreach (&transaction->cancel_hooks,

```

```

        cancel_hook_free, NULL);

_dbus_list_clear (&transaction->cancel_hooks);
}

BusTransaction*
bus_transaction_new (BusContext *context)
{
    BusTransaction *transaction;

    transaction = dbus_new0 (BusTransaction, 1);
    if (transaction == NULL)
        return NULL;

    transaction->context = context;

    return transaction;
}

BusContext*
bus_transaction_get_context (BusTransaction *transaction)
{
    return transaction->context;
}

BusConnections*
bus_transaction_get_connections (BusTransaction *transaction)
{
    return bus_context_get_connections (transaction->context);
}

dbus_bool_t
bus_transaction_send_from_driver (BusTransaction *transaction,
                                  DBusConnection *connection,
                                  DBusMessage *message)
{
    /* We have to set the sender to the driver, and have
     * to check security policy since it was not done in
     * dispatch.c
     */
    _dbus_verbose ("Sending %s %s %s from driver\n",
                  dbus_message_get_interface (message) ?
                  dbus_message_get_interface (message) : "(no
interface)",
                  dbus_message_get_member (message) ?
                  dbus_message_get_member (message) : "(no member)",
                  dbus_message_get_error_name (message) ?
                  dbus_message_get_error_name (message) : "(no error
name)");

    if (!dbus_message_set_sender (message, DBUS_SERVICE_DBUS))
        return FALSE;
}

```

```

if (bus_connection_is_active (connection))
{
    if (!dbus_message_set_destination (message,
                                      bus_connection_get_name
(connection)))
        return FALSE;
}

/* bus driver never wants a reply */
dbus_message_set_no_reply (message, TRUE);

/* If security policy doesn't allow the message, we silently
 * eat it; the driver doesn't care about getting a reply.
 */
if (!bus_context_check_security_policy (bus_transaction_get_context
(transaction),
                                      transaction,
                                      NULL, connection,
connection, message, NULL))
    return TRUE;

return bus_transaction_send (transaction, connection, message);
}

dbus_bool_t
bus_transaction_send (BusTransaction *transaction,
                    DBusConnection *connection,
                    DBusMessage *message)
{
    MessageToSend *to_send;
    BusConnectionData *d;
    DBusList *link;

    _dbus_verbose (" trying to add %s interface=%s member=%s error=%s
to transaction%s\n",
                  dbus_message_get_type (message) ==
DBUS_MESSAGE_TYPE_ERROR ? "error" :
                  dbus_message_get_reply_serial (message) != 0 ?
"reply" :
                  "message",
                  dbus_message_get_interface (message) ?
dbus_message_get_interface (message) : "(unset)",
                  dbus_message_get_member (message) ?
dbus_message_get_member (message) : "(unset)",
                  dbus_message_get_error_name (message) ?
dbus_message_get_error_name (message) : "(unset)",
                  dbus_connection_get_is_connected (connection) ?
"" : " (disconnected)");

    _dbus_assert (dbus_message_get_sender (message) != NULL);

```

```

if (!dbus_connection_get_is_connected (connection))
    return TRUE; /* silently ignore disconnected connections */

d = BUS_CONNECTION_DATA (connection);
_dbus_assert (d != NULL);

to_send = dbus_new (MessageToSend, 1);
if (to_send == NULL)
{
    return FALSE;
}

to_send->preallocated = dbus_connection_preallocate_send
(connection);
if (to_send->preallocated == NULL)
{
    dbus_free (to_send);
    return FALSE;
}

dbus_message_ref (message);
to_send->message = message;
to_send->transaction = transaction;

_dbus_verbose ("about to prepend message\n");

if (!_dbus_list_prepend (&d->transaction_messages, to_send))
{
    message_to_send_free (connection, to_send);
    return FALSE;
}

_dbus_verbose ("prepended message\n");

/* See if we already had this connection in the list
 * for this transaction. If we have a pending message,
 * then we should already be in transaction->connections
 */
link = _dbus_list_get_first_link (&d->transaction_messages);
_dbus_assert (link->data == to_send);
link = _dbus_list_get_next_link (&d->transaction_messages, link);
while (link != NULL)
{
    MessageToSend *m = link->data;
    DBusList *next = _dbus_list_get_next_link (&d-
>transaction_messages, link);

    if (m->transaction == transaction)
        break;

    link = next;
}

```

```

if (link == NULL)
{
    if (!_dbus_list_prepend (&transaction->connections, connection))
    {
        _dbus_list_remove (&d->transaction_messages, to_send);
        message_to_send_free (connection, to_send);
        return FALSE;
    }
}

return TRUE;
}

static void
connection_cancel_transaction (DBusConnection *connection,
                               BusTransaction *transaction)
{
    DBusList *link;
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    link = _dbus_list_get_first_link (&d->transaction_messages);
    while (link != NULL)
    {
        MessageToSend *m = link->data;
        DBusList *next = _dbus_list_get_next_link (&d-
>transaction_messages, link);

        if (m->transaction == transaction)
        {
            _dbus_list_remove_link (&d->transaction_messages,
                                   link);

            message_to_send_free (connection, m);
        }

        link = next;
    }
}

void
bus_transaction_cancel_and_free (BusTransaction *transaction)
{
    DBusConnection *connection;

    _dbus_verbose ("TRANSACTION: cancelled\n");

    while ((connection = _dbus_list_pop_first (&transaction-
>connections)))

```

```

    connection_cancel_transaction (connection, transaction);

    _dbus_assert (transaction->connections == NULL);

    _dbus_list_foreach (&transaction->cancel_hooks,
                       cancel_hook_cancel, NULL);

    free_cancel_hooks (transaction);

    dbus_free (transaction);
}

static void
connection_execute_transaction (DBusConnection *connection,
                               BusTransaction *transaction)
{
    DBusList *link;
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    /* Send the queue in order (FIFO) */
    link = _dbus_list_get_last_link (&d->transaction_messages);
    while (link != NULL)
    {
        MessageToSend *m = link->data;
        DBusList *prev = _dbus_list_get_prev_link (&d-
>transaction_messages, link);

        if (m->transaction == transaction)
        {
            _dbus_list_remove_link (&d->transaction_messages,
                                   link);

            _dbus_assert (dbus_message_get_sender (m->message) != NULL);

            dbus_connection_send_preallocated (connection,
                                              m->preallocated,
                                              m->message,
                                              NULL);

            m->preallocated = NULL; /* so we don't double-free it */

            message_to_send_free (connection, m);
        }

        link = prev;
    }
}

void

```

```

bus_transaction_execute_and_free (BusTransaction *transaction)
{
    /* For each connection in transaction->connections
     * send the messages
     */
    DBusConnection *connection;

    _dbus_verbose ("TRANSACTION: executing\n");

    while ((connection = _dbus_list_pop_first (&transaction-
>connections))
        connection_execute_transaction (connection, transaction);

    _dbus_assert (transaction->connections == NULL);

    free_cancel_hooks (transaction);

    dbus_free (transaction);
}

static void
bus_connection_remove_transactions (DBusConnection *connection)
{
    MessageToSend *to_send;
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    _dbus_assert (d != NULL);

    while ((to_send = _dbus_list_get_first (&d->transaction_messages))
        {
            /* only has an effect for the first MessageToSend listing this
            transaction */
            _dbus_list_remove (&to_send->transaction->connections,
                connection);

            _dbus_list_remove (&d->transaction_messages, to_send);
            message_to_send_free (connection, to_send);
        }
    }

/**
 * Converts the DBusError to a message reply
 */
dbus_bool_t
bus_transaction_send_error_reply (BusTransaction *transaction,
                                DBusConnection *connection,
                                const DBusError *error,
                                DBusMessage *in_reply_to)
{
    DBusMessage *reply;

```



```

_dbus_assert (error != NULL);
_DBUS_ASSERT_ERROR_IS_SET (error);

_dbus_verbose ("Sending error reply %s \"%s\"\n",
               error->name, error->message);

reply = dbus_message_new_error (in_reply_to,
                               error->name,
                               error->message);

if (reply == NULL)
    return FALSE;

if (!bus_transaction_send_from_driver (transaction, connection,
reply))
{
    dbus_message_unref (reply);
    return FALSE;
}

dbus_message_unref (reply);

return TRUE;
}

dbus_bool_t
bus_transaction_add_cancel_hook (BusTransaction
*transaction,
                               BusTransactionCancelFunction
cancel_function,
                               void
                               *data,
                               DBusFreeFunction
free_data_function)
{
    CancelHook *ch;

    ch = dbus_new (CancelHook, 1);
    if (ch == NULL)
        return FALSE;

    _dbus_verbose ("    adding cancel hook function = %p data = %p\n",
                  cancel_function, data);

    ch->cancel_function = cancel_function;
    ch->data = data;
    ch->free_data_function = free_data_function;

    /* It's important that the hooks get run in reverse order that they
     * were added
     */
    if (!_dbus_list_prepend (&transaction->cancel_hooks, ch))
    {
        dbus_free (ch);
    }
}

```

```

        return FALSE;
    }

    return TRUE;
}

#ifdef DBUS_ENABLE_STATS
int
bus_connections_get_n_active (BusConnections *connections)
{
    return connections->n_completed;
}

int
bus_connections_get_n_incomplete (BusConnections *connections)
{
    return connections->n_incomplete;
}

int
bus_connections_get_total_match_rules (BusConnections *connections)
{
    return connections->total_match_rules;
}

int
bus_connections_get_peak_match_rules (BusConnections *connections)
{
    return connections->peak_match_rules;
}

int
bus_connections_get_peak_match_rules_per_conn (BusConnections
*connections)
{
    return connections->peak_match_rules_per_conn;
}

int
bus_connections_get_total_bus_names (BusConnections *connections)
{
    return connections->total_bus_names;
}

int
bus_connections_get_peak_bus_names (BusConnections *connections)
{
    return connections->peak_bus_names;
}

int

```

```

bus_connections_get_peak_bus_names_per_conn (BusConnections
*connections)
{
    return connections->peak_bus_names_per_conn;
}

int
bus_connection_get_peak_match_rules (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    return d->peak_match_rules;
}

int
bus_connection_get_peak_bus_names (DBusConnection *connection)
{
    BusConnectionData *d;

    d = BUS_CONNECTION_DATA (connection);
    return d->peak_bus_names;
}
#endif /* DBUS_ENABLE_STATS */

```

File = connection.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* connection.h Client connections
 *
 * Copyright (C) 2003, 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

```

```

*
*/

#ifndef BUS_CONNECTION_H
#define BUS_CONNECTION_H

#include <dbus/dbus.h>
#include <dbus/dbus-list.h>
#include "bus.h"

typedef dbus_bool_t (* BusConnectionForeachFunction) (DBusConnection
*connection,
                                                    void
*data);

BusConnections* bus_connections_new          (BusContext
*context);
BusConnections* bus_connections_ref         (BusConnections
*connections);
void            bus_connections_unref       (BusConnections
*connections);
dbus_bool_t    bus_connections_setup_connection (BusConnections
*connections,
                                                    DBusConnection
*connection);
void            bus_connections_foreach     (BusConnections
*connections,
                                                    BusConnectionForeachFunction
function,
                                                    void
*data);
void            bus_connections_foreach_active (BusConnections
*connections,
                                                    BusConnectionForeachFunction
function,
                                                    void
*data);
BusContext*    bus_connections_get_context  (BusConnections
*connections);
void            bus_connections_increment_stamp (BusConnections
*connections);
BusContext*    bus_connection_get_context  (DBusConnection
*connection);
BusConnections* bus_connection_get_connections (DBusConnection
*connection);
BusRegistry*   bus_connection_get_registry (DBusConnection
*connection);
BusActivation* bus_connection_get_activation (DBusConnection
*connection);
BusMatchmaker* bus_connection_get_matchmaker (DBusConnection
*connection);

```

```

const char *    bus_connection_get_loginfo      (DBusConnection
*connection);
BusSELinuxID*  bus_connection_get_selinux_id   (DBusConnection
*connection);
dbus_bool_t     bus_connections_check_limits    (BusConnections
*connections,
                                                DBusConnection
*requesting_completion,
                                                DBusError
*error);
void           bus_connections_expire_incomplete (BusConnections
*connections);

dbus_bool_t     bus_connections_expect_reply    (BusConnections
*connections,
                                                BusTransaction
*transaction,
                                                DBusConnection
*will_get_reply,
                                                DBusConnection
*will_send_reply,
                                                DBusMessage
*reply_to_this,
                                                DBusError
*error);
dbus_bool_t     bus_connections_check_reply     (BusConnections
*connections,
                                                BusTransaction
*transaction,
                                                DBusConnection
*sending_reply,
                                                DBusConnection
*receiving_reply,
                                                DBusMessage
*reply,
                                                DBusError
*error);

dbus_bool_t     bus_connection_mark_stamp      (DBusConnection
*connection);

dbus_bool_t     bus_connection_is_active       (DBusConnection *connection);
const char *    bus_connection_get_name       (DBusConnection *connection);

dbus_bool_t     bus_connection_preallocate_oom_error (DBusConnection
*connection);
void           bus_connection_send_oom_error   (DBusConnection
*connection,
                                                DBusMessage
*in_reply_to);

/* called by signals.c */

```



```

BusClientPolicy* bus_connection_get_policy (DBusConnection
*connection);

/* transaction API so we can send or not send a block of messages as a
whole */

typedef void (* BusTransactionCancelFunction) (void *data);

BusTransaction* bus_transaction_new (BusContext
*context);
BusContext* bus_transaction_get_context (BusTransaction
*transaction);
BusConnections* bus_transaction_get_connections (BusTransaction
*transaction);
dbus_bool_t bus_transaction_send (BusTransaction
*transaction,
DBusConnection
*connection,
DBusMessage
*message);
dbus_bool_t bus_transaction_send_from_driver (BusTransaction
*transaction,
DBusConnection
*connection,
DBusMessage
*message);
dbus_bool_t bus_transaction_send_error_reply (BusTransaction
*transaction,
DBusConnection
*connection,
const DBusError
*error,
DBusMessage
*in_reply_to);
void bus_transaction_cancel_and_free (BusTransaction
*transaction);
void bus_transaction_execute_and_free (BusTransaction
*transaction);
dbus_bool_t bus_transaction_add_cancel_hook (BusTransaction
*transaction,
BusTransactionCancelFunction cancel_function,
void
*data,
DBusFreeFunction
free_data_function);

/* called by stats.c, only present if DBUS_ENABLE_STATS */
int bus_connections_get_n_active (BusConnections
*connections);
int bus_connections_get_n_incomplete (BusConnections
*connections);

```

```

int bus_connections_get_total_match_rules      (BusConnections
*connections);
int bus_connections_get_peak_match_rules      (BusConnections
*connections);
int bus_connections_get_peak_match_rules_per_conn (BusConnections
*connections);
int bus_connections_get_total_bus_names      (BusConnections
*connections);
int bus_connections_get_peak_bus_names      (BusConnections
*connections);
int bus_connections_get_peak_bus_names_per_conn (BusConnections
*connections);

int bus_connection_get_peak_match_rules      (DBusConnection
*connection);
int bus_connection_get_peak_bus_names      (DBusConnection
*connection);

#endif /* BUS_CONNECTION_H */

```

File = COPYING

The D-Bus glib bindings are licensed to you under your choice of the Academic Free License version 2.1, or the GNU General Public License version 2. Both licenses are included here. Some of the standalone binaries are under the GPL only; in particular, but not limited to, tests/decode-gcov.c. Each source code file is marked with the proper copyright information.

The Academic Free License
v. 2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

- b) to prepare derivative works ("Derivative Works") based upon the Original Work;
- c) to distribute copies of the Original Work and Derivative Works to the public;
- d) to perform the Original Work publicly; and
- e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution

Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software

or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101

et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

--

END OF ACADEMIC FREE LICENSE. The following is intended to describe the essential

differences between the Academic Free License (AFL) version 1.0 and other open source licenses:

The Academic Free License is similar to the BSD, MIT, UoI/NCSA and Apache licenses in many respects but it is intended to solve a few problems with those licenses.

* The AFL is written so as to make it clear what software is being licensed (by the inclusion of a statement following the copyright notice in the software). This way, the license functions better than a template license. The BSD, MIT and UoI/NCSA licenses apply to unidentified software.

* The AFL contains a complete copyright grant to the software. The BSD and Apache licenses are vague and incomplete in that respect.

* The AFL contains a complete patent grant to the software. The BSD, MIT, UoI/NCSA and Apache licenses rely on an implied patent license and contain no explicit patent grant.

* The AFL makes it clear that no trademark rights are granted to the licensor's trademarks. The Apache license contains such a provision, but the BSD, MIT and UoI/NCSA licenses do not.

* The AFL includes the warranty by the licensor that it either owns the copyright or that it is distributing the software under a license. None of the other licenses contain that warranty. All other warranties are disclaimed, as is the case for the other licenses.

* The AFL is itself copyrighted (with the right granted to copy and distribute without modification). This ensures that the owner of the copyright to the license will control changes. The Apache license contains a copyright notice, but the BSD, MIT and UoI/NCSA licenses do not.

--

START OF GNU GENERAL PUBLIC LICENSE

--

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original

authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'.
```

```
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

File = COPYING.~1~

D-Bus is licensed to you under your choice of the Academic Free License version 2.1, or the GNU General Public License version 2 (or, at your option any later version).

Both licenses are included here. Some of the standalone binaries are under the GPL only; in particular, but not limited to, tools/dbus-cleanup-sockets.c and test/decode-gcov.c. Each source code file is marked with the proper copyright information - if you find a file that isn't marked please bring it to our attention.

The Academic Free License
v. 2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

- a) to reproduce the Original Work in copies;
- b) to prepare derivative works ("Derivative Works") based upon the Original Work;
- c) to distribute copies of the Original Work and Derivative Works to the public;
- d) to perform the Original Work publicly; and
- e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an

"Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This

termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. Æ,Â§ 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

END OF ACADEMIC FREE LICENSE. The following is intended to describe the essential differences between the Academic Free License (AFL) version 1.0 and other open source licenses:

The Academic Free License is similar to the BSD, MIT, UoI/NCSA and Apache licenses in many respects but it is intended to solve a few problems with those licenses.

* The AFL is written so as to make it clear what software is being licensed (by the inclusion of a statement following the copyright notice in the software). This way, the license functions better than a template license. The BSD, MIT and UoI/NCSA licenses apply to unidentified software.

* The AFL contains a complete copyright grant to the software. The BSD and Apache licenses are vague and incomplete in that respect.

* The AFL contains a complete patent grant to the software. The BSD, MIT, UoI/NCSA and Apache licenses rely on an implied patent license and contain no explicit patent grant.

* The AFL makes it clear that no trademark rights are granted to the licensor's trademarks. The Apache license contains such a provision, but the BSD, MIT and UoI/NCSA licenses do not.

* The AFL includes the warranty by the licensor that it either owns the copyright or that it is distributing the software under a license. None of the other licenses contain that warranty. All other warranties are disclaimed, as is the case for the other licenses.

* The AFL is itself copyrighted (with the right granted to copy and distribute without modification). This ensures that the owner of the copyright to the license will control changes. The Apache license contains a copyright notice, but the BSD, MIT and UoI/NCSA licenses do not.

--

START OF GNU GENERAL PUBLIC LICENSE

--

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component

itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot

impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'.
```

```
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program
`Gnomovision' (which makes passes at compilers) written by James
Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

```
File = CPackInstallConfig.cmake
```

```
if (DBUS_INSTALL_SYSTEM_LIBS)
  if (MINGW)
    if (DBUS_USE_EXPAT)
      # expat
      install_files(/bin FILES ${LIBEXPAT_LIBRARIES})
    else (DBUS_USE_EXPAT)
      # xml2
      install_files(/bin FILES ${LIBXML2_LIBRARIES})
      install_files(/bin FILES ${LIBICONV_LIBRARIES})
    endif (DBUS_USE_EXPAT)
  else (MINGW)
    INCLUDE(InstallRequiredSystemLibraries)
  endif (MINGW)
endif (DBUS_INSTALL_SYSTEM_LIBS)

SET(CPACK_PACKAGE_DESCRIPTION_SUMMARY "D-BUS For Windows")
SET(CPACK_PACKAGE_VENDOR "D-BUS Windows Team")
SET(CPACK_PACKAGE_DESCRIPTION_FILE "${CMAKE_SOURCE_DIR}/../README")
SET(CPACK_RESOURCE_FILE_LICENSE "${CMAKE_SOURCE_DIR}/../COPYING")
# duplicated from VERSION
SET(CPACK_PACKAGE_VERSION_MAJOR ${VERSION_MAJOR})
SET(CPACK_PACKAGE_VERSION_MINOR ${VERSION_MINOR})
SET(CPACK_PACKAGE_VERSION_PATCH ${VERSION_PATCH})
```



```

#SET(CPACK_PACKAGE_INSTALL_DIRECTORY "dbus
${CMake_VERSION_MAJOR}.${CMake_VERSION_MINOR}")
SET(CPACK_PACKAGE_INSTALL_DIRECTORY "dbus")
IF(WIN32 AND NOT UNIX)
    SET(CPACK_GENERATOR NSIS ZIP)      # can be NSIS, STGZ, TBZ2, TGZ, TZ
and ZIP
    SET(CPACK_NSIS_COMPRESSOR "/SOLID lzma")
    # There is a bug in NSI that does not handle full unix paths
properly. Make
    # sure there is at least one set of four (4) backslashes.
# SET(CPACK_PACKAGE_ICON
"${CMake_SOURCE_DIR}/Utilities/Release\\\\\\InstallIcon.bmp")
    SET(CPACK_NSIS_INSTALLED_ICON_NAME "bin\\\\\\dbus-launch.bat")
    SET(CPACK_NSIS_DISPLAY_NAME "D-Bus for Windows")
    SET(CPACK_NSIS_HELP_LINK
"http:\\\\\\sourceforge.net/projects/windbus")
    SET(CPACK_NSIS_URL_INFO_ABOUT
"http:\\\\\\sourceforge.net/projects/windbus")
    SET(CPACK_NSIS_CONTACT "me@my-personal-home-page.com")
    SET(CPACK_NSIS_MODIFY_PATH ON)
ELSE(WIN32 AND NOT UNIX)
    SET(CPACK_STRIP_FILES "bin/MyExecutable")
    SET(CPACK_SOURCE_STRIP_FILES "")
ENDIF(WIN32 AND NOT UNIX)
SET(CPACK_PACKAGE_EXECUTABLES "dbus-launch" "D-Bus Daemon")
INCLUDE(CPack)

```

File = cross-compile.sh

```

#!/bin/sh
#
# cross compile script for cmake
#
# initial written by Fridrich Strba
# refactored to debian/lenny by Ralf Habacker
#
# reported to work at least on debian/lenny
#

if test -f /usr/bin/i686-pc-mingw32-gcc; then
    cross_cc=i686-pc-mingw32
elif test -f /usr/bin/i586-mingw32msvc-gcc; then
    cross_cc=i586-mingw32msvc
else
    echo "could not determine mingw cross compiler"
    exit 1
fi

if test -d ~/$cross_cc; then
    cross_root=~/$cross_cc

```

```

elif test -d /usr/$cross_cc/sys-root/mingw; then
    cross_root=/usr/$cross_cc/sys-root/mingw
elif test -d /usr/$cross_cc/lib; then
    cross_root=/usr/$cross_cc
else
    echo "could not determine mingw cross compiler sdk"
    exit 1
fi

if ! TEMP=`mktemp --tmpdir -d dbus-cross-compile.XXXXXX`; then
    echo "mktemp failed, try with coreutils 6.10 or later?" >&2
    exit 1
fi

# make cmake happy
export TEMP

HOST_CC=gcc; export HOST_CC;

if test -d $cross_root/lib/pkgconfig; then

PKG_CONFIG_PATH="$cross_root/lib/pkgconfig:$cross_root/share/pkgconfig
"; export PKG_CONFIG_PATH;
fi

if test -d "$MINGW32_CLASSPATH" || test -f
"$cross_root/share/java/libgcj.jar"; then
    CLASSPATH="$CLASSPATH:${MINGW32_CLASSPATH:-
$cross_root/share/java/libgcj.jar:$cross_root/share/java/libgcj-
tools.jar}"; export CLASSPATH;
fi

_PREFIX="/usr/bin/$cross_cc-";
for i in `ls -1 ${_PREFIX}* | grep -v 'gcc-'`; do
    x=`echo $i|sed "s,${_PREFIX},,"|sed "s,\.awk*,,"|tr "a-z+-" "A-
ZX "`;
    declare -x $x="$i" ; export $x;
done;
unset _PREFIX;

CC="${MINGW32_CC:-$cross_cc-gcc}"; export CC;
CFLAGS="${MINGW32_CFLAGS:--O2 -g -pipe -Wall -fexceptions -fno-omit-
frame-pointer -fno-optimize-sibling-calls --param=ssp-buffer-size=4 -
mms-bitfields}"; export CFLAGS;
LDFLAGS="${MINGW32_LDFLAGS:--Wl,--exclude-libs=libintl.a -Wl,--
exclude-libs=libiconv.a}"; export LDFLAGS;

if [ -x "/usr/bin/$cross_cc-g++" ]; then
    CXX="${MINGW32_CXX:-$cross_cc-g++}"; export CXX;
    CXXFLAGS="${MINGW32_CXXFLAGS:--O2 -g -pipe -Wall -fexceptions -
fno-omit-frame-pointer -fno-optimize-sibling-calls --param=ssp-buffer-
size=4 -mms-bitfields}"; export CXXFLAGS;

```

```

else
    CXX=; export CXX;
    ac_cv_prog_CXX=no; export ac_cv_prog_CXX;
    CXXFLAGS=; export CXXFLAGS;
fi;
for i in `ls $cross_root/bin/*|grep -- "-config$" ` ; do
    x=`basename $i|tr "a-z+-" "A-ZX_"|sed "s,\.,, "`;
    declare -x $x="$i" ; export $x;
done;
unset x i ;

if ! test -f "$cross_root/lib/libexpat.dll.a"; then
    (cd $TEMP && wget
http://www.winkde.org/pub/kde/ports/win32/repository/win32libs/expat-
2.0.1-bin.zip)
    (cd $TEMP && wget
http://www.winkde.org/pub/kde/ports/win32/repository/win32libs/expat-
2.0.1-lib.zip)
    (cd $cross_root && unzip -x $TMP/expat-2.0.1-bin.zip)
    (cd $cross_root && unzip -x $TMP/expat-2.0.1-lib.zip)
fi

if test -f "$cross_root/lib/libexpat.dll.a"; then
    xml_library=-DDBUS_USE_EXPAT=On -
DLIBEXPAT_INCLUDE_DIR:PATH=$cross_root/include -
DLIBEXPAT_LIBRARIES:PATH=$cross_root/lib/libexpat.dll.a
else
    echo "could not find a cross compile xml libraray"
    exit 1
fi

cmake \
    -DCMAKE_SYSTEM_NAME="Windows" \
    -DCMAKE_VERBOSE_MAKEFILE=ON \
    -DCMAKE_INSTALL_PREFIX:PATH=$cross_root \
    -DCMAKE_INSTALL_LIBDIR:PATH=$cross_root/lib \
    -DINCLUDE_INSTALL_DIR:PATH=$cross_root/include \
    -DLIB_INSTALL_DIR:PATH=$cross_root/lib \
    -DSYSCONF_INSTALL_DIR:PATH=$cross_root/etc \
    -DSHARE_INSTALL_PREFIX:PATH=$cross_root/share \
    -DBUILD_SHARED_LIBS:BOOL=ON \
    -DCMAKE_C_COMPILER="/usr/bin/$cross_cc-gcc" \
    -DCMAKE_CXX_COMPILER="/usr/bin/$cross_cc-g++" \
    -DCMAKE_FIND_ROOT_PATH="$cross_root" \
    -DCMAKE_FIND_ROOT_PATH_MODE_LIBRARY=ONLY \
    -DCMAKE_FIND_ROOT_PATH_MODE_INCLUDE=ONLY \
    -DCMAKE_CXX_COMPILER="/usr/bin/$cross_cc-g++" \
    -DCMAKE_FIND_ROOT_PATH="$cross_root" \
    -DCMAKE_FIND_ROOT_PATH_MODE_LIBRARY=ONLY \
    -DCMAKE_FIND_ROOT_PATH_MODE_INCLUDE=ONLY \
    $xml_library \
    -DCMAKE_FIND_ROOT_PATH_MODE_PROGRAM=NEVER \

```

\$*

File = dbus-1-uninstalled.pc

```
abs_top_builddir=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8
abs_top_srcdir=/home/gangadhar/newyoctobuild/tisdk/build/arago-tmp-external-linaro-toolchain/work/cortexa8hf-vfp-neon-3.8-oe-linux-gnueabi/dbus/1.6.8-r6.0-arago1/dbus-1.6.8
prefix=
exec_prefix=
system_bus_default_address=unix:path=/var/run/dbus/system_bus_socket
sysconfdir=/etc
session_bus_services_dir=/usr/share/dbus-1/services
system_bus_services_dir=/usr/share/dbus-1/system-services
interfaces_dir=/usr/share/dbus-1/interfaces
daemondir=/usr/bin
```

Name: dbus

Description: Free desktop message bus (uninstalled copy)

Version: 1.6.8

Libs: \${abs_top_builddir}/dbus/libdbus-1.la

Libs.private: -lpthread -lrt

Cflags: -I\${abs_top_srcdir}

File = dbus-1-uninstalled.pc.in

```
abs_top_builddir=@abs_top_builddir@
abs_top_srcdir=@abs_top_srcdir@
prefix=
exec_prefix=
system_bus_default_address=@DBUS_SYSTEM_BUS_DEFAULT_ADDRESS@
sysconfdir=@EXPANDED_SYSCONFDIR@
session_bus_services_dir=@EXPANDED_DATADIR@/dbus-1/services
system_bus_services_dir=@EXPANDED_DATADIR@/dbus-1/system-services
interfaces_dir=@EXPANDED_DATADIR@/dbus-1/interfaces
daemondir=@DBUS_DAEMONDIR@
```

Name: dbus

Description: Free desktop message bus (uninstalled copy)

Version: @VERSION@

Libs: \${abs_top_builddir}/dbus/libdbus-1.la

Libs.private: @LIBDBUS_LIBS@

Cflags: -I\${abs_top_srcdir} @DBUS_STATIC_BUILD_CPPFLAGS@

File = dbus-1.pc

```
prefix=/usr
exec_prefix=/usr
libdir=/usr/lib
includedir=/usr/include
system_bus_default_address=unix:path=/var/run/dbus/system_bus_socket
sysconfdir=/etc
session_bus_services_dir=/usr/share/dbus-1/services
system_bus_services_dir=/usr/share/dbus-1/system-services
interfaces_dir=/usr/share/dbus-1/interfaces
daemondir=/usr/bin
```

Name: dbus

Description: Free desktop message bus

Version: 1.6.8

Libs: -L\${libdir} -ldbus-1

Libs.private: -lpthread -lrt

Cflags: -I\${includedir}/dbus-1.0 -I\${libdir}/dbus-1.0/include

File = dbus-1.pc.in

```
prefix=@prefix@
exec_prefix=@exec_prefix@
libdir=@libdir@
includedir=@includedir@
system_bus_default_address=@DBUS_SYSTEM_BUS_DEFAULT_ADDRESS@
sysconfdir=@EXPANDED_SYSCONFDIR@
session_bus_services_dir=@EXPANDED_DATADIR@/dbus-1/services
system_bus_services_dir=@EXPANDED_DATADIR@/dbus-1/system-services
interfaces_dir=@EXPANDED_DATADIR@/dbus-1/interfaces
daemondir=@DBUS_DAEMONDIR@
```

Name: dbus

Description: Free desktop message bus

Version: @VERSION@

Libs: -L\${libdir} -ldbus-1

Libs.private: @LIBDBUS_LIBS@

Cflags: -I\${includedir}/dbus-1.0 -I\${libdir}/dbus-1.0/include

@DBUS_STATIC_BUILD_CPPFLAGS@

File = dbus-address.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-address.c Server address parser.
```

*

* Copyright (C) 2003 CodeFactory AB

```
* Copyright (C) 2004,2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-address.h"
#include "dbus-internals.h"
#include "dbus-list.h"
#include "dbus-string.h"
#include "dbus-protocol.h"
```

```
/**
 * @defgroup DBusAddressInternals Address parsing
 * @ingroup DBusInternals
 * @brief Implementation of parsing addresses of D-Bus servers.
 *
 * @{
 */
```

```
/**
 * Internals of DBusAddressEntry
 */
```

```
struct DBusAddressEntry
{
    DBusString method; /**< The address type (unix, tcp, etc.) */

    DBusList *keys;    /**< List of keys */
    DBusList *values;  /**< List of values */
};
```

```
/**
 *
```

```

* Sets #DBUS_ERROR_BAD_ADDRESS.
* If address_problem_type and address_problem_field are not #NULL,
* sets an error message about how the field is no good. Otherwise,
sets
* address_problem_other as the error message.
*
* @param error the error to set
* @param address_problem_type the address type of the bad address or
#NULL
* @param address_problem_field the missing field of the bad address
or #NULL
* @param address_problem_other any other error message or #NULL
*/
void
_dbus_set_bad_address (DBusError *error,
                      const char *address_problem_type,
                      const char *address_problem_field,
                      const char *address_problem_other)
{
    if (address_problem_type != NULL)
        dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
                        "Server address of type %s was missing argument
%s",
                        address_problem_type, address_problem_field);
    else
        dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
                        "Could not parse server address: %s",
                        address_problem_other);
}

/**
 * #TRUE if the byte need not be escaped when found in a dbus address.
 * All other bytes are required to be escaped in a valid address.
 */
#define _DBUS_ADDRESS_OPTIONALLY_ESCAPED_BYTE(b) \
    ((b) >= 'a' && (b) <= 'z') || \
    ((b) >= 'A' && (b) <= 'Z') || \
    ((b) >= '0' && (b) <= '9') || \
    (b) == '-' || \
    (b) == '_' || \
    (b) == '/' || \
    (b) == '\\\ ' || \
    (b) == '*' || \
    (b) == '.'

/**
 * Appends an escaped version of one string to another string,
 * using the D-Bus address escaping mechanism
 *
 * @param escaped the string to append to
 * @param unescaped the string to escape
 * @returns #FALSE if no memory

```

```

*/
dbus_bool_t
_dbus_address_append_escaped (DBusString      *escaped,
                              const DBusString *unescape)
{
    const char *p;
    const char *end;
    dbus_bool_t ret;
    int orig_len;

    ret = FALSE;

    orig_len = _dbus_string_get_length (escaped);
    p = _dbus_string_get_const_data (unescape);
    end = p + _dbus_string_get_length (unescape);
    while (p != end)
    {
        if (_DBUS_ADDRESS_OPTIONALLY_ESCAPED_BYTE (*p))
        {
            {
                if (!_dbus_string_append_byte (escaped, *p))
                    goto out;
            }
        }
        else
        {
            {
                if (!_dbus_string_append_byte (escaped, '%'))
                    goto out;
                if (!_dbus_string_append_byte_as_hex (escaped, *p))
                    goto out;
            }
        }

        ++p;
    }

    ret = TRUE;

out:
    if (!ret)
        _dbus_string_set_length (escaped, orig_len);
    return ret;
}

/** @} */ /* End of internals */

static void
dbus_address_entry_free (DBusAddressEntry *entry)
{
    DBusList *link;

    _dbus_string_free (&entry->method);

    link = _dbus_list_get_first_link (&entry->keys);
    while (link != NULL)

```



```

    {
        _dbus_string_free (link->data);
        dbus_free (link->data);

        link = _dbus_list_get_next_link (&entry->keys, link);
    }
    _dbus_list_clear (&entry->keys);

    link = _dbus_list_get_first_link (&entry->values);
    while (link != NULL)
    {
        _dbus_string_free (link->data);
        dbus_free (link->data);

        link = _dbus_list_get_next_link (&entry->values, link);
    }
    _dbus_list_clear (&entry->values);

    dbus_free (entry);
}

/**
 * @defgroup DBusAddress Address parsing
 * @ingroup DBus
 * @brief Parsing addresses of D-Bus servers.
 *
 * @{
 */

/**
 * Frees a #NULL-terminated array of address entries.
 *
 * @param entries the array.
 */
void
dbus_address_entries_free (DBusAddressEntry **entries)
{
    int i;

    for (i = 0; entries[i] != NULL; i++)
        dbus_address_entry_free (entries[i]);
    dbus_free (entries);
}

static DBusAddressEntry *
create_entry (void)
{
    DBusAddressEntry *entry;

    entry = dbus_new0 (DBusAddressEntry, 1);

    if (entry == NULL)

```

```

    return NULL;

    if (!_dbus_string_init (&entry->method))
    {
        dbus_free (entry);
        return NULL;
    }

    return entry;
}

/**
 * Returns the method string of an address entry. For example, given
 * the address entry "tcp:host=example.com" it would return the string
 * "tcp"
 *
 * @param entry the entry.
 * @returns a string describing the method. This string
 * must not be freed.
 */
const char *
dbus_address_entry_get_method (DBusAddressEntry *entry)
{
    return _dbus_string_get_const_data (&entry->method);
}

/**
 * Returns a value from a key of an entry. For example,
 * given the address "tcp:host=example.com,port=8073" if you asked
 * for the key "host" you would get the value "example.com"
 *
 * The returned value is already unescaped.
 *
 * @param entry the entry.
 * @param key the key.
 * @returns the key value. This string must not be freed.
 */
const char *
dbus_address_entry_get_value (DBusAddressEntry *entry,
                             const char *key)
{
    DBusList *values, *keys;

    keys = _dbus_list_get_first_link (&entry->keys);
    values = _dbus_list_get_first_link (&entry->values);

    while (keys != NULL)
    {
        _dbus_assert (values != NULL);

        if (_dbus_string_equal_c_str (keys->data, key))
            return _dbus_string_get_const_data (values->data);
    }
}

```

```

        keys = _dbus_list_get_next_link (&entry->keys, keys);
        values = _dbus_list_get_next_link (&entry->values, values);
    }

    return NULL;
}

static dbus_bool_t
append_unescaped_value (DBusString      *unescaped,
                       const DBusString *escaped,
                       int               escaped_start,
                       int               escaped_len,
                       DBusError        *error)
{
    const char *p;
    const char *end;
    dbus_bool_t ret;

    ret = FALSE;

    p = _dbus_string_get_const_data (escaped) + escaped_start;
    end = p + escaped_len;
    while (p != end)
    {
        if (_DBUS_ADDRESS_OPTIONALLY_ESCAPED_BYTE (*p))
        {
            if (!_dbus_string_append_byte (unescaped, *p))
                goto out;
        }
        else if (*p == '%')
        {
            /* Efficiency is king */
            char buf[3];
            DBusString hex;
            int hex_end;

            ++p;

            if ((p + 2) > end)
            {
                dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
                                "In D-Bus address, percent character was
                                not followed by two hex digits");
                goto out;
            }

            buf[0] = *p;
            ++p;
            buf[1] = *p;
            buf[2] = '\0';

```

```

        _dbus_string_init_const (&hex, buf);

        if (!_dbus_string_hex_decode (&hex, 0, &hex_end,
                                     unescaped,
                                     _dbus_string_get_length
(unescaped)))
            goto out;

        if (hex_end != 2)
        {
            dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
                "In D-Bus address, percent character was
followed by characters other than hex digits");
            goto out;
        }
    }
else
    {
        /* Error, should have been escaped */
        dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
            "In D-Bus address, character '%c' should
have been escaped\n",
                *p);
        goto out;
    }

    ++p;
}

ret = TRUE;

out:
if (!ret && error && !dbus_error_is_set (error))
    _DBUS_SET_OOM (error);

_dbus_assert (ret || error == NULL || dbus_error_is_set (error));

return ret;
}

/**
 * Parses an address string of the form:
 *
 * method:key=value,key=value;method:key=value
 *
 * See the D-Bus specification for complete docs on the format.
 *
 * When connecting to an address, the first address entries
 * in the semicolon-separated list should be tried first.
 *
 * @param address the address.
 * @param entry return location to an array of entries.

```

```

* @param array_len return location for array length.
* @param error address where an error can be returned.
* @returns #TRUE on success, #FALSE otherwise.
*/
dbus_bool_t
dbus_parse_address (const char          *address,
                   DBusAddressEntry ***entry,
                   int                 *array_len,
                   DBusError           *error)
{
    DBusString str;
    int pos, end_pos, len, i;
    DBusList *entries, *link;
    DBusAddressEntry **entry_array;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_string_init_const (&str, address);

    entries = NULL;
    pos = 0;
    len = _dbus_string_get_length (&str);

    if (len == 0)
    {
        dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
                       "Empty address '%s'", address);
        goto error;
    }

    while (pos < len)
    {
        DBusAddressEntry *entry;

        int found_pos;

        entry = create_entry ();
        if (!entry)
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);

            goto error;
        }

        /* Append the entry */
        if (!_dbus_list_append (&entries, entry))
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            dbus_address_entry_free (entry);
            goto error;
        }
    }
}

```

```

/* Look for a semi-colon */
if (!_dbus_string_find (&str, pos, ";", &end_pos))
end_pos = len;

/* Look for the colon : */
if (!_dbus_string_find_to (&str, pos, end_pos, ":", &found_pos))
{
    dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS, "Address does
not contain a colon");
    goto error;
}

if (!_dbus_string_copy_len (&str, pos, found_pos - pos, &entry-
>method, 0))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto error;
}

pos = found_pos + 1;

while (pos < end_pos)
{
    int comma_pos, equals_pos;

    if (!_dbus_string_find_to (&str, pos, end_pos, ",",
&comma_pos))
        comma_pos = end_pos;

    if (!_dbus_string_find_to (&str, pos, comma_pos, "=",
&equals_pos) ||
        equals_pos == pos || equals_pos + 1 == comma_pos)
    {
        dbus_set_error (error, DBUS_ERROR_BAD_ADDRESS,
            "'=' character not found or has no value
following it");
        goto error;
    }
    else
    {
        DBusString *key;
        DBusString *value;

        key = dbus_new0 (DBusString, 1);

        if (!key)
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            goto error;
        }

        value = dbus_new0 (DBusString, 1);

```

```

if (!value)
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    dbus_free (key);
    goto error;
}

if (!_dbus_string_init (key))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    dbus_free (key);
    dbus_free (value);

    goto error;
}

if (!_dbus_string_init (value))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (key);

    dbus_free (key);
    dbus_free (value);
    goto error;
}

if (!_dbus_string_copy_len (&str, pos, equals_pos - pos,
key, 0))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (key);
    _dbus_string_free (value);

    dbus_free (key);
    dbus_free (value);
    goto error;
}

if (!append_unescaped_value (value, &str, equals_pos + 1,
error))
    comma_pos - equals_pos - 1,
{
    _dbus_assert (error == NULL || dbus_error_is_set
(error));

    _dbus_string_free (key);
    _dbus_string_free (value);

    dbus_free (key);
    dbus_free (value);
    goto error;
}

```

```

        if (!_dbus_list_append (&entry->keys, key))
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            _dbus_string_free (key);
            _dbus_string_free (value);

            dbus_free (key);
            dbus_free (value);
            goto error;
        }

        if (!_dbus_list_append (&entry->values, value))
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            _dbus_string_free (value);

            dbus_free (value);
            goto error;
        }
    }

    pos = comma_pos + 1;
}

pos = end_pos + 1;
}

*array_len = _dbus_list_get_length (&entries);
entry_array = dbus_new (DBusAddressEntry *, *array_len + 1);

if (!entry_array)
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);

    goto error;
}

entry_array [*array_len] = NULL;

link = _dbus_list_get_first_link (&entries);
i = 0;
while (link != NULL)
{
    entry_array[i] = link->data;
    i++;
    link = _dbus_list_get_next_link (&entries, link);
}

_dbus_list_clear (&entries);
*entry = entry_array;

```



```

    return TRUE;

error:

    link = _dbus_list_get_first_link (&entries);
    while (link != NULL)
    {
        dbus_address_entry_free (link->data);
        link = _dbus_list_get_next_link (&entries, link);
    }

    _dbus_list_clear (&entries);

    return FALSE;
}

/**
 * Escapes the given string as a value in a key=value pair
 * for a D-Bus address.
 *
 * @param value the unescaped value
 * @returns newly-allocated escaped value or #NULL if no memory
 */
char*
dbus_address_escape_value (const char *value)
{
    DBusString escaped;
    DBusString unescaped;
    char *ret;

    ret = NULL;

    _dbus_string_init_const (&unescaped, value);

    if (!_dbus_string_init (&escaped))
        return NULL;

    if (!_dbus_address_append_escaped (&escaped, &unescaped))
        goto out;

    if (!_dbus_string_steal_data (&escaped, &ret))
        goto out;

out:
    _dbus_string_free (&escaped);
    return ret;
}

/**
 * Unescapes the given string as a value in a key=value pair
 * for a D-Bus address. Note that dbus_address_entry_get_value()

```

```

* returns an already-unescaped value.
*
* @param value the escaped value
* @param error error to set if the unescaping fails
* @returns newly-allocated unescaped value or #NULL if no memory
*/
char*
dbus_address_unescape_value (const char *value,
                             DBusError *error)
{
    DBusString unescaped;
    DBusString escaped;
    char *ret;

    ret = NULL;

    _dbus_string_init_const (&escaped, value);

    if (!_dbus_string_init (&unescaped))
        return NULL;

    if (!_dbus_append_unescaped_value (&unescaped, &escaped,
                                       0, _dbus_string_get_length (&escaped),
                                       error))
        goto out;

    if (!_dbus_string_steal_data (&unescaped, &ret))
        goto out;

out:
    if (ret == NULL && error && !dbus_error_is_set (error))
        _DBUS_SET_OOM (error);

    _dbus_assert (ret != NULL || error == NULL || dbus_error_is_set
                  (error));

    _dbus_string_free (&unescaped);
    return ret;
}

/** @} */ /* End of public API */

#ifdef DBUS_BUILD_TESTS

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include "dbus-test.h"
#include <stdlib.h>

typedef struct
{
    const char *escaped;

```

```

    const char *unescaped;
} EscapeTest;

static const EscapeTest escape_tests[] = {
    { "abcde", "abcde" },
    { "", "" },
    { "%20%20", "  " },
    { "%24", "$" },
    { "%25", "%" },
    { "abc%24", "abc$" },
    { "%24abc", "$abc" },
    { "abc%24abc", "abc$abc" },
    { "/", "/" },
    { "-", "-" },
    { " ", " " },
    { "̄", "̄" },
    { "Ā", "Ā" },
    { "İ", "İ" },
    { "Z", "Z" },
    { "a", "a" },
    { "i", "i" },
    { "z", "z" }
};

static const char* invalid_escaped_values[] = {
    "%a",
    "%q",
    "%az",
    "%%",
    "%$$",
    "abc%a",
    "%axyz",
    "%",
    "$",
    " ",
};

dbus_bool_t
_dbus_address_test (void)
{
    DBusAddressEntry **entries;
    int len;
    DBusError error = DBUS_ERROR_INIT;
    int i;

    i = 0;
    while (i < _DBUS_N_ELEMENTS (escape_tests))
    {
        const EscapeTest *test = &escape_tests[i];
        char *escaped;
        char *unescaped;

        escaped = dbus_address_escape_value (test->unescaped);

```

```

if (escaped == NULL)
    _dbus_assert_not_reached ("oom");

if (strcmp (escaped, test->escaped) != 0)
    {
        _dbus_warn ("Escaped '%s' as '%s' should have been '%s'\n",
                    test->unescaped, escaped, test->escaped);
        exit (1);
    }
dbus_free (escaped);

unescaped = dbus_address_unescape_value (test->escaped, &error);
if (unescaped == NULL)
    {
        _dbus_warn ("Failed to unescape '%s': %s\n",
                    test->escaped, error.message);
        dbus_error_free (&error);
        exit (1);
    }

if (strcmp (unescaped, test->unescaped) != 0)
    {
        _dbus_warn ("Unescaped '%s' as '%s' should have been
'%s'\n",
                    test->escaped, unescaped, test->unescaped);
        exit (1);
    }
dbus_free (unescaped);

++i;
}

i = 0;
while (i < _DBUS_N_ELEMENTS (invalid_escaped_values))
    {
        char *unescaped;

        unescaped = dbus_address_unescape_value
(invalid_escaped_values[i],
                                                &error);

        if (unescaped != NULL)
            {
                _dbus_warn ("Should not have successfully unescaped '%s' to
'%s'\n",
                            invalid_escaped_values[i], unescaped);
                dbus_free (unescaped);
                exit (1);
            }

        _dbus_assert (dbus_error_is_set (&error));
        dbus_error_free (&error);
    }

```

```

    ++i;
}

if (!dbus_parse_address
("unix:path=/tmp/foo;debug:name=test,sliff=sloff;",
    &entries, &len, &error))
    _dbus_assert_not_reached ("could not parse address");
_dbus_assert (len == 2);
_dbus_assert (strcmp (dbus_address_entry_get_value (entries[0],
"path"), "/tmp/foo") == 0);
_dbus_assert (strcmp (dbus_address_entry_get_value (entries[1],
"name"), "test") == 0);
_dbus_assert (strcmp (dbus_address_entry_get_value (entries[1],
"sliff"), "sloff") == 0);

dbus_address_entries_free (entries);

/* Different possible errors */
if (dbus_parse_address ("", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo:bar", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo:bar,baz", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo:bar=foo,baz", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo:bar=foo;baz", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

if (dbus_parse_address ("foo:=foo", &entries, &len, &error))
    _dbus_assert_not_reached ("Parsed incorrect address.");
else
    dbus_error_free (&error);

```

```

    if (dbus_parse_address ("foo:foo=", &entries, &len, &error))
        _dbus_assert_not_reached ("Parsed incorrect address.");
    else
        dbus_error_free (&error);

    if (dbus_parse_address ("foo:foo,bar=baz", &entries, &len, &error))
        _dbus_assert_not_reached ("Parsed incorrect address.");
    else
        dbus_error_free (&error);

    return TRUE;
}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif

File = dbus-address.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-address.h  Server address parser.
 *
 * Copyright (C) 2003  CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

```

```

#ifndef DBUS_ADDRESS_H
#define DBUS_ADDRESS_H

#include <dbus/dbus-types.h>
#include <dbus/dbus-errors.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusAddress
 * @{
 */

/** Opaque type representing one of the semicolon-separated items in
an address */
typedef struct DBusAddressEntry DBusAddressEntry;

DBUS_EXPORT
dbus_bool_t dbus_parse_address          (const char
*address,
                                         DBusAddressEntry ***entry,
                                         int
                                         *array_len,
                                         DBusError
                                         *error);

DBUS_EXPORT
const char *dbus_address_entry_get_value (DBusAddressEntry *entry,
                                         const char
                                         *key);

DBUS_EXPORT
const char *dbus_address_entry_get_method (DBusAddressEntry *entry);
DBUS_EXPORT
void        dbus_address_entries_free    (DBusAddressEntry
**entries);

DBUS_EXPORT
char* dbus_address_escape_value    (const char *value);
DBUS_EXPORT
char* dbus_address_unescape_value (const char *value,
                                   DBusError *error);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_ADDRESS_H */

```

File = dbus-arch-deps.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-arch-deps.h Header with architecture/compiler specific
information, installed to libdir

```

```

*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.0
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_ARCH_DEPS_H
#define DBUS_ARCH_DEPS_H

#include <dbus/dbus-macros.h>

DBUS_BEGIN_DECLS

#if 1
#define DBUS_HAVE_INT64 1
_DBUS_GNUC_EXTENSION typedef long long dbus_int64_t;
_DBUS_GNUC_EXTENSION typedef unsigned long long dbus_uint64_t;

#define DBUS_INT64_CONSTANT(val) (_DBUS_GNUC_EXTENSION (val##LL))
#define DBUS_UINT64_CONSTANT(val) (_DBUS_GNUC_EXTENSION (val##ULL))

#else
#undef DBUS_HAVE_INT64
#undef DBUS_INT64_CONSTANT
#undef DBUS_UINT64_CONSTANT
#endif

typedef int dbus_int32_t;
typedef unsigned int dbus_uint32_t;

```



```

typedef short dbus_int16_t;
typedef unsigned short dbus_uint16_t;

/* This is not really arch-dependent, but it's not worth
 * creating an additional generated header just for this
 */
#define DBUS_MAJOR_VERSION 1
#define DBUS_MINOR_VERSION 6
#define DBUS_MICRO_VERSION 8

#define DBUS_VERSION_STRING "1.6.8"

#define DBUS_VERSION ((1 << 16) | (6 << 8) | (8))

DBUS_END_DECLS

#endif /* DBUS_ARCH_DEPS_H */

File = dbus-arch-deps.h.in

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-arch-deps.h Header with architecture/compiler specific
information, installed to libdir
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.0
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

```

```

#ifndef DBUS_ARCH_DEPS_H
#define DBUS_ARCH_DEPS_H

#include <dbus/dbus-macros.h>

DBUS_BEGIN_DECLS

#ifdef @DBUS_HAVE_INT64@
#define DBUS_HAVE_INT64 1
_DBUS_GNUC_EXTENSION typedef @DBUS_INT64_TYPE@ dbus_int64_t;
_DBUS_GNUC_EXTENSION typedef unsigned @DBUS_INT64_TYPE@ dbus_uint64_t;

#define DBUS_INT64_CONSTANT(val) (_DBUS_GNUC_EXTENSION
@DBUS_INT64_CONSTANT@)
#define DBUS_UINT64_CONSTANT(val) (_DBUS_GNUC_EXTENSION
@DBUS_UINT64_CONSTANT@)

#else
#undef DBUS_HAVE_INT64
#undef DBUS_INT64_CONSTANT
#undef DBUS_UINT64_CONSTANT
#endif

typedef @DBUS_INT32_TYPE@ dbus_int32_t;
typedef unsigned @DBUS_INT32_TYPE@ dbus_uint32_t;

typedef @DBUS_INT16_TYPE@ dbus_int16_t;
typedef unsigned @DBUS_INT16_TYPE@ dbus_uint16_t;

/* This is not really arch-dependent, but it's not worth
 * creating an additional generated header just for this
 */
#define DBUS_MAJOR_VERSION @DBUS_MAJOR_VERSION@
#define DBUS_MINOR_VERSION @DBUS_MINOR_VERSION@
#define DBUS_MICRO_VERSION @DBUS_MICRO_VERSION@

#define DBUS_VERSION_STRING "@DBUS_VERSION@"

#define DBUS_VERSION ((@DBUS_MAJOR_VERSION@ << 16) |
(@DBUS_MINOR_VERSION@ << 8) | (@DBUS_MICRO_VERSION@))

DBUS_END_DECLS

#endif /* DBUS_ARCH_DEPS_H */

File = dbus-auth-script.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */

```

```

/* dbus-auth-script.c Test DBusAuth using a special script file
(internal to D-Bus implementation)
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#include <config.h>

#ifdef DBUS_BUILD_TESTS

#include "dbus-auth-script.h"
#include "dbus-auth.h"
#include "dbus-string.h"
#include "dbus-hash.h"
#include "dbus-credentials.h"
#include "dbus-internals.h"

/**
 * @defgroup DBusAuthScript code for running unit test scripts for
DBusAuth
 * @ingroup DBusInternals
 * @brief DBusAuth unit test scripting
 *
 * The code in here is used for unit testing, it loads
 * up a script that tests DBusAuth.
 *
 * @{
 */

/* this is slightly different from the other append_quoted_string
 * in dbus-message-builder.c
 */
static dbus_bool_t

```

```

append_quoted_string (DBusString      *dest,
                     const DBusString *quoted)
{
    dbus_bool_t in_quotes = FALSE;
    dbus_bool_t in_backslash = FALSE;
    int i;

    i = 0;
    while (i < _dbus_string_get_length (quoted))
        {
            unsigned char b;

            b = _dbus_string_get_byte (quoted, i);

            if (in_backslash)
                {
                    unsigned char a;

                    if (b == 'r')
                        a = '\r';
                    else if (b == 'n')
                        a = '\n';
                    else if (b == '\\')
                        a = '\\';
                    else
                        {
                            _dbus_warn ("bad backslashed byte %c\n", b);
                            return FALSE;
                        }

                    if (!_dbus_string_append_byte (dest, a))
                        return FALSE;

                    in_backslash = FALSE;
                }
            else if (b == '\\')
                {
                    in_backslash = TRUE;
                }
            else if (in_quotes)
                {
                    if (b == '\\')
                        in_quotes = FALSE;
                    else
                        {
                            if (!_dbus_string_append_byte (dest, b))
                                return FALSE;
                        }
                }
            else
                {
                    if (b == '\\')

```

```

        in_quotes = TRUE;
    else if (b == ' ' || b == '\n' || b == '\t')
        break; /* end on whitespace if not quoted */
    else
    {
        if (!_dbus_string_append_byte (dest, b))
            return FALSE;
    }
}

++i;
}

return TRUE;
}

static dbus_bool_t
same_first_word (const DBusString *a,
                const DBusString *b)
{
    int first_a_blank, first_b_blank;

    _dbus_string_find_blank (a, 0, &first_a_blank);
    _dbus_string_find_blank (b, 0, &first_b_blank);

    if (first_a_blank != first_b_blank)
        return FALSE;

    return _dbus_string_equal_len (a, b, first_a_blank);
}

static DBusAuthState
auth_state_from_string (const DBusString *str)
{
    if (_dbus_string_starts_with_c_str (str, "WAITING_FOR_INPUT"))
        return DBUS_AUTH_STATE_WAITING_FOR_INPUT;
    else if (_dbus_string_starts_with_c_str (str, "WAITING_FOR_MEMORY"))
        return DBUS_AUTH_STATE_WAITING_FOR_MEMORY;
    else if (_dbus_string_starts_with_c_str (str, "HAVE_BYTES_TO_SEND"))
        return DBUS_AUTH_STATE_HAVE_BYTES_TO_SEND;
    else if (_dbus_string_starts_with_c_str (str, "NEED_DISCONNECT"))
        return DBUS_AUTH_STATE_NEED_DISCONNECT;
    else if (_dbus_string_starts_with_c_str (str, "AUTHENTICATED"))
        return DBUS_AUTH_STATE_AUTHENTICATED;
    else
        return -1;
}

static const char*
auth_state_to_string (DBusAuthState state)
{
    switch (state)

```

```

    {
    case DBUS_AUTH_STATE_WAITING_FOR_INPUT:
        return "WAITING_FOR_INPUT";
    case DBUS_AUTH_STATE_WAITING_FOR_MEMORY:
        return "WAITING_FOR_MEMORY";
    case DBUS_AUTH_STATE_HAVE_BYTES_TO_SEND:
        return "HAVE_BYTES_TO_SEND";
    case DBUS_AUTH_STATE_NEED_DISCONNECT:
        return "NEED_DISCONNECT";
    case DBUS_AUTH_STATE_AUTHENTICATED:
        return "AUTHENTICATED";
    }

    return "unknown";
}

static char **
split_string (DBusString *str)
{
    int i, j, k, count, end;
    char **array;

    end = _dbus_string_get_length (str);

    i = 0;
    _dbus_string_skip_blank (str, i, &i);
    for (count = 0; i < end; count++)
    {
        _dbus_string_find_blank (str, i, &i);
        _dbus_string_skip_blank (str, i, &i);
    }

    array = dbus_new0 (char *, count + 1);
    if (array == NULL)
        return NULL;

    i = 0;
    _dbus_string_skip_blank (str, i, &i);
    for (k = 0; k < count; k++)
    {
        _dbus_string_find_blank (str, i, &j);

        array[k] = dbus_malloc (j - i + 1);
        if (array[k] == NULL)
        {
            dbus_free_string_array (array);
            return NULL;
        }
        memcpy (array[k],
                _dbus_string_get_const_data_len (str, i, j - i), j - i);
        array[k][j - i] = '\0';
    }
}

```

```

        _dbus_string_skip_blank (str, j, &i);
    }
    array[k] = NULL;

    return array;
}

static void
auth_set_unix_credentials(DBusAuth *auth,
                          dbus_uid_t uid,
                          dbus_pid_t pid)
{
    DBusCredentials *credentials;

    credentials = _dbus_credentials_new ();
    if (credentials == NULL)
        _dbus_assert_not_reached ("no memory");

    if (uid != DBUS_UID_UNSET)
        _dbus_credentials_add_unix_uid (credentials, uid);
    if (pid != DBUS_PID_UNSET)
        _dbus_credentials_add_unix_pid (credentials, pid);

    _dbus_auth_set_credentials (auth, credentials);

    _dbus_credentials_unref (credentials);
}

/**
 * Runs an "auth script" which is a script for testing the
 * authentication protocol. Scripts send and receive data, and then
 * include assertions about the state of both ends of the connection
 * after processing the data. A script succeeds if these assertions
 * hold.
 *
 * @param filename the file containing the script to run
 * @returns #TRUE if the script succeeds, #FALSE otherwise
 */
dbus_bool_t
_dbus_auth_script_run (const DBusString *filename)
{
    DBusString file;
    DBusError error = DBUS_ERROR_INIT;
    DBusString line;
    dbus_bool_t retval;
    int line_no;
    DBusAuth *auth;
    DBusString from_auth;
    DBusAuthState state;
    DBusString context;
    DBusString guid;

```

```

retval = FALSE;
auth = NULL;

_dbus_string_init_const (&guid, "5fa01f4202cd837709a3274ca0df9d00");
_dbus_string_init_const (&context, "org_freedesktop_test");

if (!_dbus_string_init (&file))
    return FALSE;

if (!_dbus_string_init (&line))
{
    _dbus_string_free (&file);
    return FALSE;
}

if (!_dbus_string_init (&from_auth))
{
    _dbus_string_free (&file);
    _dbus_string_free (&line);
    return FALSE;
}

if (!_dbus_file_get_contents (&file, filename, &error)) {
    _dbus_warn ("Getting contents of %s failed: %s\n",
                _dbus_string_get_const_data (filename),
error.message);
    dbus_error_free (&error);
    goto out;
}

state = DBUS_AUTH_STATE_NEED_DISCONNECT;
line_no = 0;

next_iteration:
while (_dbus_string_pop_line (&file, &line))
{
    line_no += 1;

    /* _dbus_warn ("%s\n", _dbus_string_get_const_data (&line)); */

    _dbus_string_delete_leading_blanks (&line);

    if (auth != NULL)
    {
        while ((state = _dbus_auth_do_work (auth)) ==
                DBUS_AUTH_STATE_HAVE_BYTES_TO_SEND)
        {
            const DBusString *tmp;
            if (_dbus_auth_get_bytes_to_send (auth, &tmp))
            {
                int count = _dbus_string_get_length (tmp);

```



```

        if (_dbus_string_copy (tmp, 0, &from_auth,
                                _dbus_string_get_length
(&from_auth)))
            _dbus_auth_bytes_sent (auth, count);
    }
}

if (_dbus_string_get_length (&line) == 0)
{
    /* empty line */
    goto next_iteration;
}
else if (_dbus_string_starts_with_c_str (&line,
                                         "#"))
{
    /* Ignore this comment */
    goto next_iteration;
}
#ifdef DBUS_WIN
else if (_dbus_string_starts_with_c_str (&line,
                                         "WIN_ONLY"))
{
    /* Ignore this line */
    goto next_iteration;
}
else if (_dbus_string_starts_with_c_str (&line,
                                         "UNIX_ONLY"))
{
    /* skip this file */
    _dbus_warn ("skipping unix only auth script\n");
    retval = TRUE;
    goto out;
}
#endif
#ifdef DBUS_UNIX
else if (_dbus_string_starts_with_c_str (&line,
                                         "UNIX_ONLY"))
{
    /* Ignore this line */
    goto next_iteration;
}
else if (_dbus_string_starts_with_c_str (&line,
                                         "WIN_ONLY"))
{
    /* skip this file */
    _dbus_warn ("skipping windows only auth script\n");
    retval = TRUE;
    goto out;
}
#endif
else if (_dbus_string_starts_with_c_str (&line,

```

```

                                                                    "CLIENT"))
{
    DBusCredentials *creds;

    if (auth != NULL)
    {
        _dbus_warn ("already created a DBusAuth (CLIENT or
SERVER given twice)\n");
        goto out;
    }

    auth = _dbus_auth_client_new ();
    if (auth == NULL)
    {
        _dbus_warn ("no memory to create DBusAuth\n");
        goto out;
    }

    /* test ref/unref */
    _dbus_auth_ref (auth);
    _dbus_auth_unref (auth);

    creds = _dbus_credentials_new_from_current_process ();
    if (creds == NULL)
    {
        _dbus_warn ("no memory for credentials\n");
        _dbus_auth_unref (auth);
        auth = NULL;
        goto out;
    }

    if (!_dbus_auth_set_credentials (auth, creds))
    {
        _dbus_warn ("no memory for setting credentials\n");
        _dbus_auth_unref (auth);
        auth = NULL;
        _dbus_credentials_unref (creds);
        goto out;
    }

    _dbus_credentials_unref (creds);
}
else if (_dbus_string_starts_with_c_str (&line,
                                                                    "SERVER"))
{
    DBusCredentials *creds;

    if (auth != NULL)
    {
        _dbus_warn ("already created a DBusAuth (CLIENT or
SERVER given twice)\n");
        goto out;
    }

```

```

    }

    auth = _dbus_auth_server_new (&guid);
    if (auth == NULL)
    {
        _dbus_warn ("no memory to create DBusAuth\n");
        goto out;
    }

    /* test ref/unref */
    _dbus_auth_ref (auth);
    _dbus_auth_unref (auth);

    creds = _dbus_credentials_new_from_current_process ();
    if (creds == NULL)
    {
        _dbus_warn ("no memory for credentials\n");
        _dbus_auth_unref (auth);
        auth = NULL;
        goto out;
    }

    if (!_dbus_auth_set_credentials (auth, creds))
    {
        _dbus_warn ("no memory for setting credentials\n");
        _dbus_auth_unref (auth);
        auth = NULL;
        _dbus_credentials_unref (creds);
        goto out;
    }

    _dbus_credentials_unref (creds);

    _dbus_auth_set_context (auth, &context);
}
else if (auth == NULL)
{
    _dbus_warn ("must specify CLIENT or SERVER\n");
    goto out;
}
else if (_dbus_string_starts_with_c_str (&line,
                                         "NO_CREDENTIALS"))
{
    auth_set_unix_credentials (auth, DBUS_UID_UNSET,
    DBUS_PID_UNSET);
}
else if (_dbus_string_starts_with_c_str (&line,
                                         "ROOT_CREDENTIALS"))
{
    auth_set_unix_credentials (auth, 0, DBUS_PID_UNSET);
}

```

```

else if (_dbus_string_starts_with_c_str (&line,
                                         "SILLY_CREDENTIALS"))
    {
        auth_set_unix_credentials (auth, 4312, DBUS_PID_UNSET);
    }
else if (_dbus_string_starts_with_c_str (&line,
                                         "ALLOWED_MECHS"))
    {
        char **mechs;

        _dbus_string_delete_first_word (&line);
        mechs = split_string (&line);
        _dbus_auth_set_mechanisms (auth, (const char **) mechs);
        _dbus_free_string_array (mechs);
    }
else if (_dbus_string_starts_with_c_str (&line,
                                         "SEND"))
    {
        DBusString to_send;

        _dbus_string_delete_first_word (&line);

        if (!_dbus_string_init (&to_send))
            {
                _dbus_warn ("no memory to allocate string\n");
                goto out;
            }

        if (!append_quoted_string (&to_send, &line))
            {
                _dbus_warn ("failed to append quoted string line %d\n",
                            line_no);
                _dbus_string_free (&to_send);
                goto out;
            }

        _dbus_verbose ("Sending '%s'\n", _dbus_string_get_const_data
(&to_send));

        if (!_dbus_string_append (&to_send, "\r\n"))
            {
                _dbus_warn ("failed to append \r\n from line %d\n",
                            line_no);
                _dbus_string_free (&to_send);
                goto out;
            }

        /* Replace USERID_HEX with our username in hex */
        {
            int where;

            if (_dbus_string_find (&to_send, 0,

```

```

                                "USERID_HEX", &where))
{
    DBusString username;

    if (!_dbus_string_init (&username))
    {
        _dbus_warn ("no memory for userid\n");
        _dbus_string_free (&to_send);
        goto out;
    }

    if (!_dbus_append_user_from_current_process
(&username))
    {
        _dbus_warn ("no memory for userid\n");
        _dbus_string_free (&username);
        _dbus_string_free (&to_send);
        goto out;
    }

    _dbus_string_delete (&to_send, where, strlen
("USERID_HEX"));

    if (!_dbus_string_hex_encode (&username, 0,
                                &to_send, where))
    {
        _dbus_warn ("no memory to subst USERID_HEX\n");
        _dbus_string_free (&username);
        _dbus_string_free (&to_send);
        goto out;
    }

    _dbus_string_free (&username);
}
else if (_dbus_string_find (&to_send, 0,
                            "USERNAME_HEX", &where))
{
    DBusString username;

    if (!_dbus_string_init (&username))
    {
        _dbus_warn ("no memory for username\n");
        _dbus_string_free (&to_send);
        goto out;
    }

    if (!_dbus_append_user_from_current_process
(&username))
    {
        _dbus_warn ("no memory for username\n");
        _dbus_string_free (&username);
        _dbus_string_free (&to_send);
    }
}

```

```

        goto out;
    }

    _dbus_string_delete (&to_send, where, strlen
("USERNAME_HEX"));

    if (!_dbus_string_hex_encode (&username, 0,
        &to_send, where))
    {
        _dbus_warn ("no memory to subst USERNAME_HEX\n");
        _dbus_string_free (&username);
        _dbus_string_free (&to_send);
        goto out;
    }

    _dbus_string_free (&username);
}

{
    DBusString *buffer;

    _dbus_auth_get_buffer (auth, &buffer);
    if (!_dbus_string_copy (&to_send, 0,
        buffer, _dbus_string_get_length
(buffer)))
    {
        _dbus_warn ("not enough memory to call bytes_received,
or can't add bytes to auth object already in end state\n");
        _dbus_string_free (&to_send);
        _dbus_auth_return_buffer (auth, buffer, 0);
        goto out;
    }

    _dbus_auth_return_buffer (auth, buffer,
_dbus_string_get_length (&to_send));
}

    _dbus_string_free (&to_send);
}
else if (_dbus_string_starts_with_c_str (&line,
        "EXPECT_STATE"))
{
    DBusAuthState expected;

    _dbus_string_delete_first_word (&line);

    expected = auth_state_from_string (&line);
    if (expected < 0)
    {
        _dbus_warn ("bad auth state given to EXPECT_STATE\n");
        goto parse_failed;
    }
}

```

```

    }

    if (expected != state)
    {
        _dbus_warn ("expected auth state %s but got %s on line
%d\n",
                    auth_state_to_string (expected),
                    auth_state_to_string (state),
                    line_no);
        goto out;
    }
}
else if (_dbus_string_starts_with_c_str (&line,
                                        "EXPECT_COMMAND"))
{
    DBusString received;

    _dbus_string_delete_first_word (&line);

    if (!_dbus_string_init (&received))
    {
        _dbus_warn ("no mem to allocate string received\n");
        goto out;
    }

    if (!_dbus_string_pop_line (&from_auth, &received))
    {
        _dbus_warn ("no line popped from the DBusAuth being
tested, expected command %s on line %d\n",
                    _dbus_string_get_const_data (&line),
                    line_no);
        _dbus_string_free (&received);
        goto out;
    }

    if (!same_first_word (&received, &line))
    {
        _dbus_warn ("line %d expected command '%s' and got
'%s'\n",
                    line_no,
                    _dbus_string_get_const_data (&line),
                    _dbus_string_get_const_data (&received));
        _dbus_string_free (&received);
        goto out;
    }

    _dbus_string_free (&received);
}
else if (_dbus_string_starts_with_c_str (&line,
                                        "EXPECT_UNUSED"))
{
    DBusString expected;

```

```

const DBusString *unused;

_dbus_string_delete_first_word (&line);

if (!_dbus_string_init (&expected))
{
    _dbus_warn ("no mem to allocate string expected\n");
    goto out;
}

if (!append_quoted_string (&expected, &line))
{
    _dbus_warn ("failed to append quoted string line %d\n",
                line_no);
    _dbus_string_free (&expected);
    goto out;
}

_dbus_auth_get_unused_bytes (auth, &unused);

if (_dbus_string_equal (&expected, unused))
{
    _dbus_auth_delete_unused_bytes (auth);
    _dbus_string_free (&expected);
}
else
{
    _dbus_warn ("Expected unused bytes '%s' and have
's'\n",
                _dbus_string_get_const_data (&expected),
                _dbus_string_get_const_data (unused));
    _dbus_string_free (&expected);
    goto out;
}
else if (_dbus_string_starts_with_c_str (&line,
"EXPECT_HAVE_NO_CREDENTIALS"))
{
    DBusCredentials *authorized_identity;

    authorized_identity = _dbus_auth_get_identity (auth);
    if (!_dbus_credentials_are_anonymous (authorized_identity))
    {
        _dbus_warn ("Expected anonymous login or failed login,
but some credentials were authorized\n");
        goto out;
    }
}
else if (_dbus_string_starts_with_c_str (&line,
"EXPECT_HAVE_SOME_CREDENTIALS"))

```



```

    {
        DBusCredentials *authorized_identity;

        authorized_identity = _dbus_auth_get_identity (auth);
        if (_dbus_credentials_are_anonymous (authorized_identity))
        {
            _dbus_warn ("Expected to have some credentials, but we
don't\n");
            goto out;
        }
    }
    else if (_dbus_string_starts_with_c_str (&line,
                                           "EXPECT"))
    {
        DBusString expected;

        _dbus_string_delete_first_word (&line);

        if (!_dbus_string_init (&expected))
        {
            _dbus_warn ("no mem to allocate string expected\n");
            goto out;
        }

        if (!append_quoted_string (&expected, &line))
        {
            _dbus_warn ("failed to append quoted string line %d\n",
                        line_no);
            _dbus_string_free (&expected);
            goto out;
        }

        if (_dbus_string_equal_len (&expected, &from_auth,
                                   _dbus_string_get_length
(&expected)))
        {
            _dbus_string_delete (&from_auth, 0,
                                _dbus_string_get_length
(&expected));
            _dbus_string_free (&expected);
        }
        else
        {
            _dbus_warn ("Expected exact string '%s' and have
'%s'\n",
                        _dbus_string_get_const_data (&expected),
                        _dbus_string_get_const_data (&from_auth));
            _dbus_string_free (&expected);
            goto out;
        }
    }
}
else

```

```

        goto parse_failed;

        goto next_iteration; /* skip parse_failed */

parse_failed:
    {
        _dbus_warn ("couldn't process line %d \"%s\"\n",
                    line_no, _dbus_string_get_const_data (&line));
        goto out;
    }
}

if (auth == NULL)
{
    _dbus_warn ("Auth script is bogus, did not even have CLIENT or
SERVER\n");
    goto out;
}
else if (state == DBUS_AUTH_STATE_AUTHENTICATED)
{
    const DBusString *unused;

    _dbus_auth_get_unused_bytes (auth, &unused);

    if (_dbus_string_get_length (unused) > 0)
    {
        _dbus_warn ("did not expect unused bytes (scripts must
specify explicitly if they are expected)\n");
        goto out;
    }
}

if (_dbus_string_get_length (&from_auth) > 0)
{
    _dbus_warn ("script did not have EXPECT_ statements for all the
data received from the DBusAuth\n");
    _dbus_warn ("Leftover data: %s\n", _dbus_string_get_const_data
(&from_auth));
    goto out;
}

retval = TRUE;

out:
if (auth)
    _dbus_auth_unref (auth);

_dbus_string_free (&file);
_dbus_string_free (&line);
_dbus_string_free (&from_auth);

return retval;

```

```

}

/** @} */
#endif /* DBUS_BUILD_TESTS */

File = dbus-auth-script.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-auth-script.h Test DBusAuth using a special script file
(internal to D-Bus implementation)
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef DBUS_AUTH_SCRIPT_H
#define DBUS_AUTH_SCRIPT_H

#include <dbus/dbus-memory.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-string.h>

DBUS_BEGIN_DECLS

dbus_bool_t _dbus_auth_script_run (const DBusString *filename);

DBUS_END_DECLS

#endif /* DBUS_AUTH_SCRIPT_H */

```

File = dbus-auth-util.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-auth-util.c Would be in dbus-auth.c, but only used for
tests/bus
*
* Copyright (C) 2002, 2003, 2004 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#include "dbus-internals.h"
#include "dbus-test.h"
#include "dbus-auth.h"

/**
 * @addtogroup DBusAuth
 * @{
 */

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include "dbus-auth-script.h"
#include <stdio.h>

static dbus_bool_t
process_test_subdir (const DBusString
                    const char
                    *test_base_dir,
                    *subdir)
{
    DBusString test_directory;
```

```

DBusString filename;
DBusDirIter *dir;
dbus_bool_t retval;
DBusError error = DBUS_ERROR_INIT;

retval = FALSE;
dir = NULL;

if (!_dbus_string_init (&test_directory))
    _dbus_assert_not_reached ("didn't allocate test_directory\n");

_dbus_string_init_const (&filename, subdir);

if (!_dbus_string_copy (test_base_dir, 0,
                       &test_directory, 0))
    _dbus_assert_not_reached ("couldn't copy test_base_dir to
test_directory");

if (!_dbus_concat_dir_and_file (&test_directory, &filename))
    _dbus_assert_not_reached ("couldn't allocate full path");

_dbus_string_free (&filename);
if (!_dbus_string_init (&filename))
    _dbus_assert_not_reached ("didn't allocate filename string\n");

dir = _dbus_directory_open (&test_directory, &error);
if (dir == NULL)
{
    _dbus_warn ("Could not open %s: %s\n",
               _dbus_string_get_const_data (&test_directory),
               error.message);
    dbus_error_free (&error);
    goto failed;
}

printf ("Testing %s:\n", subdir);

next:
while (_dbus_directory_get_next_file (dir, &filename, &error))
{
    DBusString full_path;

    if (!_dbus_string_init (&full_path))
        _dbus_assert_not_reached ("couldn't init string");

    if (!_dbus_string_copy (&test_directory, 0, &full_path, 0))
        _dbus_assert_not_reached ("couldn't copy dir to full_path");

    if (!_dbus_concat_dir_and_file (&full_path, &filename))
        _dbus_assert_not_reached ("couldn't concat file to dir");

    if (!_dbus_string_ends_with_c_str (&filename, ".auth-script"))

```

```

        {
            _dbus_verbose ("Skipping non-.auth-script file %s\n",
                           _dbus_string_get_const_data (&filename));
            _dbus_string_free (&full_path);
            goto next;
        }

printf ("    %s\n", _dbus_string_get_const_data (&filename));

if (!_dbus_auth_script_run (&full_path))
    {
        _dbus_string_free (&full_path);
        goto failed;
    }
else
    _dbus_string_free (&full_path);
}

if (dbus_error_is_set (&error))
    {
        _dbus_warn ("Could not get next file in %s: %s\n",
                    _dbus_string_get_const_data (&test_directory),
error.message);
        dbus_error_free (&error);
        goto failed;
    }

retval = TRUE;

failed:

if (dir)
    _dbus_directory_close (dir);
_dbus_string_free (&test_directory);
_dbus_string_free (&filename);

return retval;
}

static dbus_bool_t
process_test_dirs (const char *test_data_dir)
{
    DBusString test_directory;
    dbus_bool_t retval;

    retval = FALSE;

    _dbus_string_init_const (&test_directory, test_data_dir);

    if (!process_test_subdir (&test_directory, "auth"))
        goto failed;

```

```

    retval = TRUE;

failed:

    _dbus_string_free (&test_directory);

    return retval;
}

dbus_bool_t
_dbus_auth_test (const char *test_data_dir)
{

    if (test_data_dir == NULL)
        return TRUE;

    if (!process_test_dirs (test_data_dir))
        return FALSE;

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-auth.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-auth.c Authentication
 *
 * Copyright (C) 2002, 2003, 2004 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

```

```

*/

#include <config.h>
#include "dbus-auth.h"
#include "dbus-string.h"
#include "dbus-list.h"
#include "dbus-internals.h"
#include "dbus-keyring.h"
#include "dbus-sha.h"
#include "dbus-protocol.h"
#include "dbus-credentials.h"

/**
 * @defgroup DBusAuth Authentication
 * @ingroup DBusInternals
 * @brief DBusAuth object
 *
 * DBusAuth manages the authentication negotiation when a connection
 * is first established, and also manage any encryption used over a
 * connection.
 *
 * @todo some SASL profiles require sending the empty string as a
 * challenge/response, but we don't currently allow that in our
 * protocol.
 *
 * @todo right now sometimes both ends will block waiting for input
 * from the other end, e.g. if there's an error during
 * DBUS_COOKIE_SHA1.
 *
 * @todo the cookie keyring needs to be cached globally not just
 * per-auth (which raises threadsafety issues too)
 *
 * @todo grep FIXME in dbus-auth.c
 */

/**
 * @defgroup DBusAuthInternals Authentication implementation details
 * @ingroup DBusInternals
 * @brief DBusAuth implementation details
 *
 * Private details of authentication code.
 *
 * @{
 */

/**
 * This function appends an initial client response to the given
 * string
 */
typedef dbus_bool_t (* DBusInitialResponseFunction) (DBusAuth
*auth,

```



```

                                                                    DbusString
*response);

/**
 * This function processes a block of data received from the peer.
 * i.e. handles a DATA command.
 */
typedef dbus_bool_t (* DbusAuthDataFunction) (DBusAuth
*auth,
                                                                    const DbusString
*data);

/**
 * This function encodes a block of data from the peer.
 */
typedef dbus_bool_t (* DbusAuthEncodeFunction) (DBusAuth
*auth,
                                                                    const DbusString
*data,
                                                                    DbusString
*encoded);

/**
 * This function decodes a block of data from the peer.
 */
typedef dbus_bool_t (* DbusAuthDecodeFunction) (DBusAuth
*auth,
                                                                    const DbusString
*data,
                                                                    DbusString
*decoded);

/**
 * This function is called when the mechanism is abandoned.
 */
typedef void (* DbusAuthShutdownFunction) (DBusAuth
*auth);

/**
 * Virtual table representing a particular auth mechanism.
 */
typedef struct
{
    const char *mechanism; /**< Name of the mechanism */
    DbusAuthDataFunction server_data_func; /**< Function on server side
for DATA */
    DbusAuthEncodeFunction server_encode_func; /**< Function on server
side to encode */
    DbusAuthDecodeFunction server_decode_func; /**< Function on server
side to decode */
    DbusAuthShutdownFunction server_shutdown_func; /**< Function on
server side to shut down */

```

```

    DBusInitialResponseFunction client_initial_response_func; /**<
Function on client side to handle initial response */
    DBusAuthDataFunction client_data_func; /**< Function on client side
for DATA */
    DBusAuthEncodeFunction client_encode_func; /**< Function on client
side for encode */
    DBusAuthDecodeFunction client_decode_func; /**< Function on client
side for decode */
    DBusAuthShutdownFunction client_shutdown_func; /**< Function on
client side for shutdown */
} DBusAuthMechanismHandler;

/**
 * Enumeration for the known authentication commands.
 */
typedef enum {
    DBUS_AUTH_COMMAND_AUTH,
    DBUS_AUTH_COMMAND_CANCEL,
    DBUS_AUTH_COMMAND_DATA,
    DBUS_AUTH_COMMAND_BEGIN,
    DBUS_AUTH_COMMAND_REJECTED,
    DBUS_AUTH_COMMAND_OK,
    DBUS_AUTH_COMMAND_ERROR,
    DBUS_AUTH_COMMAND_UNKNOWN,
    DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD,
    DBUS_AUTH_COMMAND_AGREE_UNIX_FD
} DBusAuthCommand;

/**
 * Auth state function, determines the reaction to incoming events for
 * a particular state. Returns whether we had enough memory to
 * complete the operation.
 */
typedef dbus_bool_t (* DBusAuthStateFunction) (DBusAuth      *auth,
                                               DBusAuthCommand
command,
                                               const DBusString
*args);

/**
 * Information about a auth state.
 */
typedef struct
{
    const char *name;           /**< Name of the state */
    DBusAuthStateFunction handler; /**< State function for this state
*/
} DBusAuthStateData;

/**
 * Internal members of DBusAuth.
 */

```

```

struct DBusAuth
{
    int refcount;           /**< reference count */
    const char *side;      /**< Client or server */

    DBusString incoming;   /**< Incoming data buffer */
    DBusString outgoing;   /**< Outgoing data buffer */

    const DBusAuthStateData *state;      /**< Current protocol state
*/

    const DBusAuthMechanismHandler *mech; /**< Current auth mechanism
*/

    DBusString identity;   /**< Current identity we're
authorizing
                                *   as.
                                */

    DBusCredentials *credentials;        /**< Credentials read from
socket
                                */

    DBusCredentials *authorized_identity; /**< Credentials that are
authorized */

    DBusCredentials *desired_identity;   /**< Identity client has
requested */

    DBusString context;           /**< Cookie scope */
    DBusKeyring *keyring;         /**< Keyring for cookie mechanism.
*/
    int cookie_id;                /**< ID of cookie to use */
    DBusString challenge;         /**< Challenge sent to client */

    char **allowed_mechs;         /**< Mechanisms we're allowed to
use,
                                * or #NULL if we can use any
                                */

    unsigned int needed_memory : 1; /**< We needed memory to continue
since last
done
                                * successful getting something
                                */

    unsigned int already_got_mechanisms : 1; /**< Client already
got mech list */
    unsigned int already_asked_for_initial_response : 1; /**< Already
sent a blank challenge to get an initial response */
    unsigned int buffer_outstanding : 1; /**< Buffer is "checked out"
for reading data into */

```

```

    unsigned int unix_fd_possible : 1; /**< This side could do unix fd
    passing */
    unsigned int unix_fd_negotiated : 1; /**< Unix fd was successfully
    negotiated */
};

/**
 * "Subclass" of DbusAuth for client side
 */
typedef struct
{
    DbusAuth base;    /**< Parent class */

    DbusList *mechs_to_try; /**< Mechanisms we got from the server that
    we're going to try using */

    DbusString guid_from_server; /**< GUID received from server */
} DbusAuthClient;

/**
 * "Subclass" of DbusAuth for server side.
 */
typedef struct
{
    DbusAuth base;    /**< Parent class */

    int failures;    /**< Number of times client has been rejected */
    int max_failures; /**< Number of times we reject before disconnect
    */

    DbusString guid; /**< Our globally unique ID in hex encoding */
} DbusAuthServer;

static void          goto_state          (DbusAuth
*auth,

                                const DbusAuthStateData
*new_state);
static dbus_bool_t  send_auth           (DbusAuth *auth,
                                const
DbusAuthMechanismHandler *mech);
static dbus_bool_t  send_data           (DbusAuth *auth,
                                DbusString *data);
static dbus_bool_t  send_rejected       (DbusAuth *auth);
static dbus_bool_t  send_error          (DbusAuth *auth,
                                const char *message);
static dbus_bool_t  send_ok             (DbusAuth *auth);
static dbus_bool_t  send_begin          (DbusAuth *auth);
static dbus_bool_t  send_cancel         (DbusAuth *auth);
static dbus_bool_t  send_negotiate_unix_fd (DbusAuth *auth);
static dbus_bool_t  send_agree_unix_fd  (DbusAuth *auth);

```

```

/**
 * Client states
 */

static dbus_bool_t handle_server_state_waiting_for_auth (DBusAuth
*auth,

DBusAuthCommand  command,

                                                                    const
DBusString *args);
static dbus_bool_t handle_server_state_waiting_for_data (DBusAuth
*auth,

DBusAuthCommand  command,

                                                                    const
DBusString *args);
static dbus_bool_t handle_server_state_waiting_for_begin (DBusAuth
*auth,

DBusAuthCommand  command,

                                                                    const
DBusString *args);

static const DBusAuthStateData server_state_waiting_for_auth = {
    "WaitingForAuth", handle_server_state_waiting_for_auth
};
static const DBusAuthStateData server_state_waiting_for_data = {
    "WaitingForData", handle_server_state_waiting_for_data
};
static const DBusAuthStateData server_state_waiting_for_begin = {
    "WaitingForBegin", handle_server_state_waiting_for_begin
};

/**
 * Client states
 */

static dbus_bool_t handle_client_state_waiting_for_data (DBusAuth
*auth,

DBusAuthCommand  command,

                                                                    const
DBusString *args);
static dbus_bool_t handle_client_state_waiting_for_ok (DBusAuth
*auth,

DBusAuthCommand  command,

                                                                    const
DBusString *args);
static dbus_bool_t handle_client_state_waiting_for_reject (DBusAuth
*auth,

```

```

DBusAuthCommand  command,
                                                    const
DBusString *args);
static dbus_bool_t handle_client_state_waiting_for_agree_unix_fd
(DBusAuth *auth,
DBusAuthCommand  command,
                                                    const
DBusString *args);

static const DBusAuthStateData client_state_need_send_auth = {
    "NeedSendAuth", NULL
};
static const DBusAuthStateData client_state_waiting_for_data = {
    "WaitingForData", handle_client_state_waiting_for_data
};
static const DBusAuthStateData client_state_waiting_for_ok = {
    "WaitingForOK", handle_client_state_waiting_for_ok
};
static const DBusAuthStateData client_state_waiting_for_reject = {
    "WaitingForReject", handle_client_state_waiting_for_reject
};
static const DBusAuthStateData client_state_waiting_for_agree_unix_fd
= {
    "WaitingForAgreeUnixFD",
handle_client_state_waiting_for_agree_unix_fd
};

/**
 * Common terminal states.  Terminal states have handler == NULL.
 */

static const DBusAuthStateData common_state_authenticated = {
    "Authenticated",  NULL
};

static const DBusAuthStateData common_state_need_disconnect = {
    "NeedDisconnect",  NULL
};

static const char auth_side_client[] = "client";
static const char auth_side_server[] = "server";
/**
 * @param auth the auth conversation
 * @returns #TRUE if the conversation is the server side
 */
#define DBUS_AUTH_IS_SERVER(auth) ((auth)->side == auth_side_server)
/**
 * @param auth the auth conversation
 * @returns #TRUE if the conversation is the client side
 */

```

```

#define DBUS_AUTH_IS_CLIENT(auth) ((auth)->side == auth_side_client)
/**
 * @param auth the auth conversation
 * @returns auth cast to DBusAuthClient
 */
#define DBUS_AUTH_CLIENT(auth) ((DBusAuthClient*)(auth))
/**
 * @param auth the auth conversation
 * @returns auth cast to DBusAuthServer
 */
#define DBUS_AUTH_SERVER(auth) ((DBusAuthServer*)(auth))

/**
 * The name of the auth ("client" or "server")
 * @param auth the auth conversation
 * @returns a string
 */
#define DBUS_AUTH_NAME(auth) ((auth)->side)

static DBusAuth*
_dbus_auth_new (int size)
{
    DBusAuth *auth;

    auth = dbus_malloc0 (size);
    if (auth == NULL)
        return NULL;

    auth->refcount = 1;

    auth->keyring = NULL;
    auth->cookie_id = -1;

    /* note that we don't use the max string length feature,
     * because you can't use that feature if you're going to
     * try to recover from out-of-memory (it creates
     * what looks like unrecoverable inability to alloc
     * more space in the string). But we do handle
     * overlong buffers in _dbus_auth_do_work().
     */

    if (!_dbus_string_init (&auth->incoming))
        goto enomem_0;

    if (!_dbus_string_init (&auth->outgoing))
        goto enomem_1;

    if (!_dbus_string_init (&auth->identity))
        goto enomem_2;

    if (!_dbus_string_init (&auth->context))
        goto enomem_3;

```

```

    if (!_dbus_string_init (&auth->challenge))
        goto enomem_4;

    /* default context if none is specified */
    if (!_dbus_string_append (&auth->context,
"org_freedesktop_general"))
        goto enomem_5;

    auth->credentials = _dbus_credentials_new ();
    if (auth->credentials == NULL)
        goto enomem_6;

    auth->authorized_identity = _dbus_credentials_new ();
    if (auth->authorized_identity == NULL)
        goto enomem_7;

    auth->desired_identity = _dbus_credentials_new ();
    if (auth->desired_identity == NULL)
        goto enomem_8;

    return auth;

#if 0
    enomem_9:
        _dbus_credentials_unref (auth->desired_identity);
#endif
    enomem_8:
        _dbus_credentials_unref (auth->authorized_identity);
    enomem_7:
        _dbus_credentials_unref (auth->credentials);
    enomem_6:
        /* last alloc was an append to context, which is freed already below
        */ ;
    enomem_5:
        _dbus_string_free (&auth->challenge);
    enomem_4:
        _dbus_string_free (&auth->context);
    enomem_3:
        _dbus_string_free (&auth->identity);
    enomem_2:
        _dbus_string_free (&auth->outgoing);
    enomem_1:
        _dbus_string_free (&auth->incoming);
    enomem_0:
        dbus_free (auth);
    return NULL;
}

static void
shutdown_mech (DBusAuth *auth)
{

```



```

/* Cancel any auth */
auth->already_asked_for_initial_response = FALSE;
_dbus_string_set_length (&auth->identity, 0);

_dbus_credentials_clear (auth->authorized_identity);
_dbus_credentials_clear (auth->desired_identity);

if (auth->mech != NULL)
{
    _dbus_verbose ("%s: Shutting down mechanism %s\n",
                  DBUS_AUTH_NAME (auth), auth->mech->mechanism);

    if (DBUS_AUTH_IS_CLIENT (auth))
        (* auth->mech->client_shutdown_func) (auth);
    else
        (* auth->mech->server_shutdown_func) (auth);

    auth->mech = NULL;
}
}

/*
 * DBUS_COOKIE_SHA1 mechanism
 */

/* Returns TRUE but with an empty string hash if the
 * cookie_id isn't known. As with all this code
 * TRUE just means we had enough memory.
 */
static dbus_bool_t
shal_compute_hash (DBusAuth      *auth,
                  int             cookie_id,
                  const DBusString *server_challenge,
                  const DBusString *client_challenge,
                  DBusString      *hash)
{
    DBusString cookie;
    DBusString to_hash;
    dbus_bool_t retval;

    _dbus_assert (auth->keyring != NULL);

    retval = FALSE;

    if (!_dbus_string_init (&cookie))
        return FALSE;

    if (!_dbus_keyring_get_hex_key (auth->keyring, cookie_id,
                                   &cookie))
        goto out_0;

    if (_dbus_string_get_length (&cookie) == 0)

```



```

{
/* We haven't sent a challenge yet, we're expecting a desired
 * username from the client.
 */
DBusString tmp;
DBusString tmp2;
dbus_bool_t retval;
DBusError error;

retval = FALSE;

_dbus_string_set_length (&auth->challenge, 0);

if (_dbus_string_get_length (data) > 0)
{
    if (_dbus_string_get_length (&auth->identity) > 0)
    {
        /* Tried to send two auth identities, wtf */
        _dbus_verbose ("%s: client tried to send auth identity, but
we already have one\n",
                        DBUS_AUTH_NAME (auth));
        return send_rejected (auth);
    }
    else
    {
        /* this is our auth identity */
        if (!_dbus_string_copy (data, 0, &auth->identity, 0))
            return FALSE;
    }
}

if (!_dbus_credentials_add_from_user (auth->desired_identity, data))
{
    _dbus_verbose ("%s: Did not get a valid username from client\n",
                    DBUS_AUTH_NAME (auth));
    return send_rejected (auth);
}

if (!_dbus_string_init (&tmp))
    return FALSE;

if (!_dbus_string_init (&tmp2))
{
    _dbus_string_free (&tmp);
    return FALSE;
}

/* we cache the keyring for speed, so here we drop it if it's the
 * wrong one. FIXME caching the keyring here is useless since we use
 * a different DBusAuth for every connection.
 */
if (auth->keyring &&

```

```

        !_dbus_keyring_is_for_credentials (auth->keyring,
                                          auth->desired_identity))
    {
        _dbus_keyring_unref (auth->keyring);
        auth->keyring = NULL;
    }

    if (auth->keyring == NULL)
    {
        dbus_error_init (&error);
        auth->keyring = _dbus_keyring_new_for_credentials (auth-
>desired_identity,
                                                         &auth-
>context,
                                                         &error);

        if (auth->keyring == NULL)
        {
            if (dbus_error_has_name (&error,
                                     DBUS_ERROR_NO_MEMORY))
            {
                dbus_error_free (&error);
                goto out;
            }
            else
            {
                _DBUS_ASSERT_ERROR_IS_SET (&error);
                _dbus_verbose ("%s: Error loading keyring: %s\n",
                               DBUS_AUTH_NAME (auth), error.message);
                if (send_rejected (auth))
                    retval = TRUE; /* retval is only about mem */
                dbus_error_free (&error);
                goto out;
            }
        }
        else
        {
            _dbus_assert (!dbus_error_is_set (&error));
        }
    }

    _dbus_assert (auth->keyring != NULL);

    dbus_error_init (&error);
    auth->cookie_id = _dbus_keyring_get_best_key (auth->keyring,
&error);
    if (auth->cookie_id < 0)
    {
        _DBUS_ASSERT_ERROR_IS_SET (&error);
        _dbus_verbose ("%s: Could not get a cookie ID to send to client:
%s\n",
                      DBUS_AUTH_NAME (auth), error.message);

```



```

{
/* We are expecting a response which is the hex-encoded client
 * challenge, space, then SHA-1 hash of the concatenation of our
 * challenge, ":", client challenge, ":", secret key, all
 * hex-encoded.
 */
int i;
DBusString client_challenge;
DBusString client_hash;
dbus_bool_t retval;
DBusString correct_hash;

retval = FALSE;

if (!_dbus_string_find_blank (data, 0, &i))
{
    _dbus_verbose ("%s: no space separator in client response\n",
                    DBUS_AUTH_NAME (auth));
    return send_rejected (auth);
}

if (!_dbus_string_init (&client_challenge))
    goto out_0;

if (!_dbus_string_init (&client_hash))
    goto out_1;

if (!_dbus_string_copy_len (data, 0, i, &client_challenge,
                           0))
    goto out_2;

_dbus_string_skip_blank (data, i, &i);

if (!_dbus_string_copy_len (data, i,
                           _dbus_string_get_length (data) - i,
                           &client_hash,
                           0))
    goto out_2;

if (_dbus_string_get_length (&client_challenge) == 0 ||
    _dbus_string_get_length (&client_hash) == 0)
{
    _dbus_verbose ("%s: zero-length client challenge or hash\n",
                    DBUS_AUTH_NAME (auth));
    if (send_rejected (auth))
        retval = TRUE;
    goto out_2;
}

if (!_dbus_string_init (&correct_hash))
    goto out_2;

```

```

if (!sha1_compute_hash (auth, auth->cookie_id,
                        &auth->challenge,
                        &client_challenge,
                        &correct_hash))
    goto out_3;

/* if cookie_id was invalid, then we get an empty hash */
if (_dbus_string_get_length (&correct_hash) == 0)
{
    if (send_rejected (auth))
        retval = TRUE;
    goto out_3;
}

if (!_dbus_string_equal (&client_hash, &correct_hash))
{
    if (send_rejected (auth))
        retval = TRUE;
    goto out_3;
}

if (!_dbus_credentials_add_credentials (auth->authorized_identity,
                                       auth->desired_identity))
    goto out_3;

/* Copy process ID from the socket credentials if it's there
*/
if (!_dbus_credentials_add_credential (auth->authorized_identity,
DBUS_CREDENTIAL_UNIX_PROCESS_ID,
                                       auth->credentials))
    goto out_3;

if (!send_ok (auth))
    goto out_3;

_dbus_verbose ("%s: authenticated client using DBUS_COOKIE_SHA1\n",
               DBUS_AUTH_NAME (auth));

retval = TRUE;

out_3:
_dbus_string_zero (&correct_hash);
_dbus_string_free (&correct_hash);
out_2:
_dbus_string_zero (&client_hash);
_dbus_string_free (&client_hash);
out_1:
_dbus_string_free (&client_challenge);
out_0:
return retval;
}

```

```

static dbus_bool_t
handle_server_data_cookie_shal_mech (DBusAuth      *auth,
                                     const DBusString *data)
{
    if (auth->cookie_id < 0)
        return shal_handle_first_client_response (auth, data);
    else
        return shal_handle_second_client_response (auth, data);
}

static void
handle_server_shutdown_cookie_shal_mech (DBusAuth *auth)
{
    auth->cookie_id = -1;
    _dbus_string_set_length (&auth->challenge, 0);
}

static dbus_bool_t
handle_client_initial_response_cookie_shal_mech (DBusAuth  *auth,
                                                  DBusString *response)
{
    DBusString username;
    dbus_bool_t retval;

    retval = FALSE;

    if (!_dbus_string_init (&username))
        return FALSE;

    if (!_dbus_append_user_from_current_process (&username))
        goto out_0;

    if (!_dbus_string_hex_encode (&username, 0,
                                 response,
                                 _dbus_string_get_length (response)))
        goto out_0;

    retval = TRUE;

out_0:
    _dbus_string_free (&username);

    return retval;
}

static dbus_bool_t
handle_client_data_cookie_shal_mech (DBusAuth      *auth,
                                     const DBusString *data)
{
    /* The data we get from the server should be the cookie context
     * name, the cookie ID, and the server challenge, separated by

```



```

    * spaces. We send back our challenge string and the correct hash.
    */
dbus_bool_t retval;
DBusString context;
DBusString cookie_id_str;
DBusString server_challenge;
DBusString client_challenge;
DBusString correct_hash;
DBusString tmp;
int i, j;
long val;

retval = FALSE;

if (!_dbus_string_find_blank (data, 0, &i))
{
    if (send_error (auth,
                    "Server did not send context/ID/challenge
properly"))
        retval = TRUE;
    goto out_0;
}

if (!_dbus_string_init (&context))
    goto out_0;

if (!_dbus_string_copy_len (data, 0, i,
                           &context, 0))
    goto out_1;

_dbus_string_skip_blank (data, i, &i);
if (!_dbus_string_find_blank (data, i, &j))
{
    if (send_error (auth,
                    "Server did not send context/ID/challenge
properly"))
        retval = TRUE;
    goto out_1;
}

if (!_dbus_string_init (&cookie_id_str))
    goto out_1;

if (!_dbus_string_copy_len (data, i, j - i,
                           &cookie_id_str, 0))
    goto out_2;

if (!_dbus_string_init (&server_challenge))
    goto out_2;

i = j;
_dbus_string_skip_blank (data, i, &i);

```

```

j = _dbus_string_get_length (data);

if (!_dbus_string_copy_len (data, i, j - i,
                           &server_challenge, 0))
    goto out_3;

if (!_dbus_keyring_validate_context (&context))
{
    if (send_error (auth, "Server sent invalid cookie context"))
        retval = TRUE;
    goto out_3;
}

if (!_dbus_string_parse_int (&cookie_id_str, 0, &val, NULL))
{
    if (send_error (auth, "Could not parse cookie ID as an
integer"))
        retval = TRUE;
    goto out_3;
}

if (_dbus_string_get_length (&server_challenge) == 0)
{
    if (send_error (auth, "Empty server challenge string"))
        retval = TRUE;
    goto out_3;
}

if (auth->keyring == NULL)
{
    DBusError error;

    dbus_error_init (&error);
    auth->keyring = _dbus_keyring_new_for_credentials (NULL,
                                                       &context,
                                                       &error);

    if (auth->keyring == NULL)
    {
        if (dbus_error_has_name (&error,
                                DBUS_ERROR_NO_MEMORY))
        {
            dbus_error_free (&error);
            goto out_3;
        }
        else
        {
            _DBUS_ASSERT_ERROR_IS_SET (&error);

            _dbus_verbose ("%s: Error loading keyring: %s\n",
                           DBUS_AUTH_NAME (auth), error.message);
        }
    }
}

```

```

        if (send_error (auth, "Could not load cookie file"))
            retval = TRUE; /* retval is only about mem */

        dbus_error_free (&error);
        goto out_3;
    }
}
else
{
    _dbus_assert (!dbus_error_is_set (&error));
}
}

_dbus_assert (auth->keyring != NULL);

if (!_dbus_string_init (&tmp))
    goto out_3;

if (!_dbus_generate_random_bytes (&tmp, N_CHALLENGE_BYTES))
    goto out_4;

if (!_dbus_string_init (&client_challenge))
    goto out_4;

if (!_dbus_string_hex_encode (&tmp, 0, &client_challenge, 0))
    goto out_5;

if (!_dbus_string_init (&correct_hash))
    goto out_5;

if (!sha1_compute_hash (auth, val,
                        &server_challenge,
                        &client_challenge,
                        &correct_hash))
    goto out_6;

if (_dbus_string_get_length (&correct_hash) == 0)
{
    /* couldn't find the cookie ID or something */
    if (send_error (auth, "Don't have the requested cookie ID"))
        retval = TRUE;
    goto out_6;
}

_dbus_string_set_length (&tmp, 0);

if (!_dbus_string_copy (&client_challenge, 0, &tmp,
                      _dbus_string_get_length (&tmp)))
    goto out_6;

if (!_dbus_string_append (&tmp, " "))
    goto out_6;

```

```

    if (!_dbus_string_copy (&correct_hash, 0, &tmp,
                           _dbus_string_get_length (&tmp)))
        goto out_6;

    if (!send_data (auth, &tmp))
        goto out_6;

    retval = TRUE;

out_6:
    _dbus_string_zero (&correct_hash);
    _dbus_string_free (&correct_hash);
out_5:
    _dbus_string_free (&client_challenge);
out_4:
    _dbus_string_zero (&tmp);
    _dbus_string_free (&tmp);
out_3:
    _dbus_string_free (&server_challenge);
out_2:
    _dbus_string_free (&cookie_id_str);
out_1:
    _dbus_string_free (&context);
out_0:
    return retval;
}

static void
handle_client_shutdown_cookie_shal_mech (DBusAuth *auth)
{
    auth->cookie_id = -1;
    _dbus_string_set_length (&auth->challenge, 0);
}

/*
 * EXTERNAL mechanism
 */

static dbus_bool_t
handle_server_data_external_mech (DBusAuth *auth,
                                  const DBusString *data)
{
    if (_dbus_credentials_are_anonymous (auth->credentials))
    {
        _dbus_verbose ("%s: no credentials, mechanism EXTERNAL can't
authenticate\n",
                      DBUS_AUTH_NAME (auth));
        return send_rejected (auth);
    }

    if (_dbus_string_get_length (data) > 0)

```

```

    {
        if (_dbus_string_get_length (&auth->identity) > 0)
            {
                /* Tried to send two auth identities, wtf */
                _dbus_verbose ("%s: client tried to send auth identity, but
we already have one\n",
                    DBUS_AUTH_NAME (auth));
                return send_rejected (auth);
            }
        else
            {
                /* this is our auth identity */
                if (!_dbus_string_copy (data, 0, &auth->identity, 0))
                    return FALSE;
            }
    }

    /* Poke client for an auth identity, if none given */
    if (_dbus_string_get_length (&auth->identity) == 0 &&
        !auth->already_asked_for_initial_response)
        {
            if (send_data (auth, NULL))
                {
                    _dbus_verbose ("%s: sending empty challenge asking client
for auth identity\n",
                        DBUS_AUTH_NAME (auth));
                    auth->already_asked_for_initial_response = TRUE;
                    goto_state (auth, &server_state_waiting_for_data);
                    return TRUE;
                }
            else
                return FALSE;
        }

    _dbus_credentials_clear (auth->desired_identity);

    /* If auth->identity is still empty here, then client
    * responded with an empty string after we poked it for
    * an initial response. This means to try to auth the
    * identity provided in the credentials.
    */
    if (_dbus_string_get_length (&auth->identity) == 0)
        {
            if (!_dbus_credentials_add_credentials (auth->desired_identity,
                auth->credentials))
                {
                    return FALSE; /* OOM */
                }
        }
    else
        {
            if (!_dbus_credentials_add_from_user (auth->desired_identity,

```

```

                                &auth->identity))
    {
        _dbus_verbose ("%s: could not get credentials from uid
string\n",
                                DBUS_AUTH_NAME (auth));
        return send_rejected (auth);
    }
}

if (_dbus_credentials_are_anonymous (auth->desired_identity))
{
    _dbus_verbose ("%s: desired user %s is no good\n",
                                DBUS_AUTH_NAME (auth),
                                _dbus_string_get_const_data (&auth->identity));
    return send_rejected (auth);
}

if (_dbus_credentials_are_superset (auth->credentials,
                                auth->desired_identity))
{
    /* client has authenticated */
    if (!_dbus_credentials_add_credentials (auth-
>authorized_identity,
                                auth->desired_identity))
        return FALSE;

    /* also copy process ID from the socket credentials
    */
    if (!_dbus_credentials_add_credential (auth-
>authorized_identity,
DBUS_CREDENTIAL_UNIX_PROCESS_ID,
                                auth->credentials))
        return FALSE;

    /* also copy audit data from the socket credentials
    */
    if (!_dbus_credentials_add_credential (auth-
>authorized_identity,
DBUS_CREDENTIAL_ADT_AUDIT_DATA_ID,
                                auth->credentials))
        return FALSE;

    if (!send_ok (auth))
        return FALSE;

    _dbus_verbose ("%s: authenticated client based on socket
credentials\n",
                                DBUS_AUTH_NAME (auth));

    return TRUE;
}

```



```

    return TRUE;
}

static void
handle_client_shutdown_external_mech (DBusAuth *auth)
{
}

/*
 * ANONYMOUS mechanism
 */

static dbus_bool_t
handle_server_data_anonymous_mech (DBusAuth *auth,
                                   const DBusString *data)
{
    if (_dbus_string_get_length (data) > 0)
    {
        /* Client is allowed to send "trace" data, the only defined
         * meaning is that if it contains '@' it is an email address,
         * and otherwise it is anything else, and it's supposed to be
         * UTF-8
         */
        if (!_dbus_string_validate_utf8 (data, 0,
            _dbus_string_get_length (data)))
        {
            _dbus_verbose ("%s: Received invalid UTF-8 trace data from
ANONYMOUS client\n",
                           DBUS_AUTH_NAME (auth));
            return send_rejected (auth);
        }

        _dbus_verbose ("%s: ANONYMOUS client sent trace string: '%s'\n",
            DBUS_AUTH_NAME (auth),
            _dbus_string_get_const_data (data));
    }

    /* We want to be anonymous (clear in case some other protocol got
    midway through I guess) */
    _dbus_credentials_clear (auth->desired_identity);

    /* Copy process ID from the socket credentials
    */
    if (!_dbus_credentials_add_credential (auth->authorized_identity,
        DBUS_CREDENTIAL_UNIX_PROCESS_ID,
            auth->credentials))
        return FALSE;

    /* Anonymous is always allowed */

```



```

if (!send_ok (auth))
    return FALSE;

_dbus_verbose ("%s: authenticated client as anonymous\n",
                DBUS_AUTH_NAME (auth));

return TRUE;
}

static void
handle_server_shutdown_anonymous_mech (DBusAuth *auth)
{

}

static dbus_bool_t
handle_client_initial_response_anonymous_mech (DBusAuth *auth,
                                               DBusString
*response)
{
    /* Our initial response is a "trace" string which must be valid UTF-
8
    * and must be an email address if it contains '@'.
    * We just send the dbus implementation info, like a user-agent or
    * something, because... why not. There's nothing guaranteed here
    * though, we could change it later.
    */
    DBusString plaintext;

    if (!_dbus_string_init (&plaintext))
        return FALSE;

    if (!_dbus_string_append (&plaintext,
                             "libdbus " DBUS_VERSION_STRING))
        goto failed;

    if (!_dbus_string_hex_encode (&plaintext, 0,
                                  response,
                                  _dbus_string_get_length (response)))
        goto failed;

    _dbus_string_free (&plaintext);

    return TRUE;

failed:
    _dbus_string_free (&plaintext);
    return FALSE;
}

static dbus_bool_t
handle_client_data_anonymous_mech (DBusAuth *auth,

```

```

                                const DBusString *data)
{
    return TRUE;
}

static void
handle_client_shutdown_anonymous_mech (DBusAuth *auth)
{
}

/* Put mechanisms here in order of preference.
 * Right now we have:
 *
 * - EXTERNAL checks socket credentials (or in the future, other info
from the OS)
 * - DBUS_COOKIE_SHA1 uses a cookie in the home directory, like xauth
or ICE
 * - ANONYMOUS checks nothing but doesn't auth the person as a user
 *
 * We might ideally add a mechanism to chain to Cyrus SASL so we can
 * use its mechanisms as well.
 *
 */
static const DBusAuthMechanismHandler
all_mechanisms[] = {
    { "EXTERNAL",
      handle_server_data_external_mech,
      NULL, NULL,
      handle_server_shutdown_external_mech,
      handle_client_initial_response_external_mech,
      handle_client_data_external_mech,
      NULL, NULL,
      handle_client_shutdown_external_mech },
    { "DBUS_COOKIE_SHA1",
      handle_server_data_cookie_shal_mech,
      NULL, NULL,
      handle_server_shutdown_cookie_shal_mech,
      handle_client_initial_response_cookie_shal_mech,
      handle_client_data_cookie_shal_mech,
      NULL, NULL,
      handle_client_shutdown_cookie_shal_mech },
    { "ANONYMOUS",
      handle_server_data_anonymous_mech,
      NULL, NULL,
      handle_server_shutdown_anonymous_mech,
      handle_client_initial_response_anonymous_mech,
      handle_client_data_anonymous_mech,
      NULL, NULL,
      handle_client_shutdown_anonymous_mech },
    { NULL, NULL }
}

```

```

};

static const DBusAuthMechanismHandler*
find_mech (const DBusString *name,
           char **allowed_mechs)
{
    int i;

    if (allowed_mechs != NULL &&
        !_dbus_string_array_contains ((const char**) allowed_mechs,
                                     _dbus_string_get_const_data
(name)))
        return NULL;

    i = 0;
    while (all_mechanisms[i].mechanism != NULL)
    {
        if (_dbus_string_equal_c_str (name,
                                     all_mechanisms[i].mechanism))

            return &all_mechanisms[i];

        ++i;
    }

    return NULL;
}

static dbus_bool_t
send_auth (DBusAuth *auth, const DBusAuthMechanismHandler *mech)
{
    DBusString auth_command;

    if (!_dbus_string_init (&auth_command))
        return FALSE;

    if (!_dbus_string_append (&auth_command,
                             "AUTH "))
    {
        _dbus_string_free (&auth_command);
        return FALSE;
    }

    if (!_dbus_string_append (&auth_command,
                             mech->mechanism))
    {
        _dbus_string_free (&auth_command);
        return FALSE;
    }

    if (mech->client_initial_response_func != NULL)
    {

```

```

        if (!_dbus_string_append (&auth_command, " "))
        {
            _dbus_string_free (&auth_command);
            return FALSE;
        }

        if (!(* mech->client_initial_response_func) (auth,
&auth_command))
        {
            _dbus_string_free (&auth_command);
            return FALSE;
        }
    }

    if (!_dbus_string_append (&auth_command,
                            "\r\n"))
    {
        _dbus_string_free (&auth_command);
        return FALSE;
    }

    if (!_dbus_string_copy (&auth_command, 0,
                            &auth->outgoing,
                            _dbus_string_get_length (&auth->outgoing)))
    {
        _dbus_string_free (&auth_command);
        return FALSE;
    }

    _dbus_string_free (&auth_command);
    shutdown_mech (auth);
    auth->mech = mech;
    goto_state (auth, &client_state_waiting_for_data);

    return TRUE;
}

static dbus_bool_t
send_data (DBusAuth *auth, DBusString *data)
{
    int old_len;

    if (data == NULL || _dbus_string_get_length (data) == 0)
        return _dbus_string_append (&auth->outgoing, "DATA\r\n");
    else
    {
        old_len = _dbus_string_get_length (&auth->outgoing);
        if (!_dbus_string_append (&auth->outgoing, "DATA "))
            goto out;

        if (!_dbus_string_hex_encode (data, 0, &auth->outgoing,

```

```

                                _dbus_string_get_length (&auth-
>outgoing)))
    goto out;

    if (!_dbus_string_append (&auth->outgoing, "\r\n"))
        goto out;

    return TRUE;

out:
    _dbus_string_set_length (&auth->outgoing, old_len);

    return FALSE;
}
}

static dbus_bool_t
send_rejected (DBusAuth *auth)
{
    DBusString command;
    DBusAuthServer *server_auth;
    int i;

    if (!_dbus_string_init (&command))
        return FALSE;

    if (!_dbus_string_append (&command,
                            "REJECTED"))
        goto nomem;

    i = 0;
    while (all_mechanisms[i].mechanism != NULL)
    {
        if (!_dbus_string_append (&command,
                                " "))
            goto nomem;

        if (!_dbus_string_append (&command,
                                all_mechanisms[i].mechanism))
            goto nomem;

        ++i;
    }

    if (!_dbus_string_append (&command, "\r\n"))
        goto nomem;

    if (!_dbus_string_copy (&command, 0, &auth->outgoing,
                            _dbus_string_get_length (&auth->outgoing)))
        goto nomem;

    shutdown_mech (auth);
}

```

```

_dbus_assert (DBUS_AUTH_IS_SERVER (auth));
server_auth = DBUS_AUTH_SERVER (auth);
server_auth->failures += 1;

if (server_auth->failures >= server_auth->max_failures)
    goto_state (auth, &common_state_need_disconnect);
else
    goto_state (auth, &server_state_waiting_for_auth);

_dbus_string_free (&command);

return TRUE;

nomem:
_dbus_string_free (&command);
return FALSE;
}

static dbus_bool_t
send_error (DBusAuth *auth, const char *message)
{
    return _dbus_string_append_printf (&auth->outgoing,
                                       "ERROR \"%s\"\r\n", message);
}

static dbus_bool_t
send_ok (DBusAuth *auth)
{
    int orig_len;

    orig_len = _dbus_string_get_length (&auth->outgoing);

    if (_dbus_string_append (&auth->outgoing, "OK ") &&
        _dbus_string_copy (& DBUS_AUTH_SERVER (auth)->guid,
                          0,
                          &auth->outgoing,
                          _dbus_string_get_length (&auth->outgoing)) &&
        _dbus_string_append (&auth->outgoing, "\r\n"))
    {
        goto_state (auth, &server_state_waiting_for_begin);
        return TRUE;
    }
    else
    {
        _dbus_string_set_length (&auth->outgoing, orig_len);
        return FALSE;
    }
}

static dbus_bool_t
send_begin (DBusAuth *auth)

```

```

{
    if (!_dbus_string_append (&auth->outgoing,
                              "BEGIN\r\n"))
        return FALSE;

    goto_state (auth, &common_state_authenticated);
    return TRUE;
}

static dbus_bool_t
process_ok(DBusAuth *auth,
          const DBusString *args_from_ok) {

    int end_of_hex;

    /* "args_from_ok" should be the GUID, whitespace already pulled off
    the front */
    _dbus_assert (_dbus_string_get_length (& DBUS_AUTH_CLIENT (auth)-
>guid_from_server) == 0);

    /* We decode the hex string to binary, using guid_from_server as
    scratch... */

    end_of_hex = 0;
    if (!_dbus_string_hex_decode (args_from_ok, 0, &end_of_hex,
                                  & DBUS_AUTH_CLIENT (auth)-
>guid_from_server, 0))
        return FALSE;

    /* now clear out the scratch */
    _dbus_string_set_length (& DBUS_AUTH_CLIENT (auth)-
>guid_from_server, 0);

    if (end_of_hex != _dbus_string_get_length (args_from_ok) ||
        end_of_hex == 0)
    {
        _dbus_verbose ("Bad GUID from server, parsed %d bytes and had %d
bytes from server\n",
                      end_of_hex, _dbus_string_get_length
(args_from_ok));
        goto_state (auth, &common_state_need_disconnect);
        return TRUE;
    }

    if (!_dbus_string_copy (args_from_ok, 0, &DBUS_AUTH_CLIENT (auth)-
>guid_from_server, 0)) {
        _dbus_string_set_length (& DBUS_AUTH_CLIENT (auth)-
>guid_from_server, 0);
        return FALSE;
    }
}

```

```

    _dbus_verbose ("Got GUID '%s' from the server\n",
                  _dbus_string_get_const_data (& DBUS_AUTH_CLIENT
(auth)->guid_from_server));

    if (auth->unix_fd_possible)
        return send_negotiate_unix_fd(auth);

    _dbus_verbose("Not negotiating unix fd passing, since not
possible\n");
    return send_begin (auth);
}

static dbus_bool_t
send_cancel (DBusAuth *auth)
{
    if (_dbus_string_append (&auth->outgoing, "CANCEL\r\n"))
        {
            goto_state (auth, &client_state_waiting_for_reject);
            return TRUE;
        }
    else
        return FALSE;
}

static dbus_bool_t
process_data (DBusAuth *auth,
              const DBusString *args,
              DBusAuthDataFunction data_func)
{
    int end;
    DBusString decoded;

    if (!_dbus_string_init (&decoded))
        return FALSE;

    if (!_dbus_string_hex_decode (args, 0, &end, &decoded, 0))
        {
            _dbus_string_free (&decoded);
            return FALSE;
        }

    if (_dbus_string_get_length (args) != end)
        {
            _dbus_string_free (&decoded);
            if (!send_error (auth, "Invalid hex encoding"))
                return FALSE;

            return TRUE;
        }

#ifdef DBUS_ENABLE_VERBOSE_MODE
    if (_dbus_string_validate_ascii (&decoded, 0,

```



```

                                _dbus_string_get_length
(&decoded))
    _dbus_verbose ("%s: data: '%s'\n",
                   DBUS_AUTH_NAME (auth),
                   _dbus_string_get_const_data (&decoded));
#endif

    if (!(* data_func) (auth, &decoded))
    {
        _dbus_string_free (&decoded);
        return FALSE;
    }

    _dbus_string_free (&decoded);
    return TRUE;
}

static dbus_bool_t
send_negotiate_unix_fd (DBusAuth *auth)
{
    if (!_dbus_string_append (&auth->outgoing,
                             "NEGOTIATE_UNIX_FD\r\n"))
        return FALSE;

    goto_state (auth, &client_state_waiting_for_agree_unix_fd);
    return TRUE;
}

static dbus_bool_t
send_agree_unix_fd (DBusAuth *auth)
{
    _dbus_assert (auth->unix_fd_possible);

    auth->unix_fd_negotiated = TRUE;
    _dbus_verbose ("Agreed to UNIX FD passing\n");

    if (!_dbus_string_append (&auth->outgoing,
                             "AGREE_UNIX_FD\r\n"))
        return FALSE;

    goto_state (auth, &server_state_waiting_for_begin);
    return TRUE;
}

static dbus_bool_t
handle_auth (DBusAuth *auth, const DBusString *args)
{
    if (_dbus_string_get_length (args) == 0)
    {
        /* No args to the auth, send mechanisms */
        if (!send_rejected (auth))
            return FALSE;
    }
}

```

```

    return TRUE;
}
else
{
    int i;
    DBusString mech;
    DBusString hex_response;

    _dbus_string_find_blank (args, 0, &i);

    if (!_dbus_string_init (&mech))
        return FALSE;

    if (!_dbus_string_init (&hex_response))
    {
        _dbus_string_free (&mech);
        return FALSE;
    }

    if (!_dbus_string_copy_len (args, 0, i, &mech, 0))
        goto failed;

    _dbus_string_skip_blank (args, i, &i);
    if (!_dbus_string_copy (args, i, &hex_response, 0))
        goto failed;

    auth->mech = find_mech (&mech, auth->allowed_mechs);
    if (auth->mech != NULL)
    {
        _dbus_verbose ("%s: Trying mechanism %s\n",
                       DBUS_AUTH_NAME (auth),
                       auth->mech->mechanism);

        if (!process_data (auth, &hex_response,
                          auth->mech->server_data_func))
            goto failed;
    }
    else
    {
        /* Unsupported mechanism */
        _dbus_verbose ("%s: Unsupported mechanism %s\n",
                       DBUS_AUTH_NAME (auth),
                       _dbus_string_get_const_data (&mech));

        if (!send_rejected (auth))
            goto failed;
    }

    _dbus_string_free (&mech);
    _dbus_string_free (&hex_response);
}

```

```

        return TRUE;

failed:
    auth->mech = NULL;
    _dbus_string_free (&mech);
    _dbus_string_free (&hex_response);
    return FALSE;
}
}

static dbus_bool_t
handle_server_state_waiting_for_auth (DBusAuth *auth,
                                       DBusAuthCommand command,
                                       const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_AUTH:
            return handle_auth (auth, args);

        case DBUS_AUTH_COMMAND_CANCEL:
        case DBUS_AUTH_COMMAND_DATA:
            return send_error (auth, "Not currently in an auth
conversation");

        case DBUS_AUTH_COMMAND_BEGIN:
            goto_state (auth, &common_state_need_disconnect);
            return TRUE;

        case DBUS_AUTH_COMMAND_ERROR:
            return send_rejected (auth);

        case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
            return send_error (auth, "Need to authenticate first");

        case DBUS_AUTH_COMMAND_REJECTED:
        case DBUS_AUTH_COMMAND_OK:
        case DBUS_AUTH_COMMAND_UNKNOWN:
        case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
        default:
            return send_error (auth, "Unknown command");
    }
}

static dbus_bool_t
handle_server_state_waiting_for_data (DBusAuth *auth,
                                       DBusAuthCommand command,
                                       const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_AUTH:

```

```

    return send_error (auth, "Sent AUTH while another AUTH in
progress");

case DBUS_AUTH_COMMAND_CANCEL:
case DBUS_AUTH_COMMAND_ERROR:
    return send_rejected (auth);

case DBUS_AUTH_COMMAND_DATA:
    return process_data (auth, args, auth->mech->server_data_func);

case DBUS_AUTH_COMMAND_BEGIN:
    goto_state (auth, &common_state_need_disconnect);
    return TRUE;

case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
    return send_error (auth, "Need to authenticate first");

case DBUS_AUTH_COMMAND_REJECTED:
case DBUS_AUTH_COMMAND_OK:
case DBUS_AUTH_COMMAND_UNKNOWN:
case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
default:
    return send_error (auth, "Unknown command");
}
}

static dbus_bool_t
handle_server_state_waiting_for_begin (DBusAuth      *auth,
                                       DBusAuthCommand  command,
                                       const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_AUTH:
            return send_error (auth, "Sent AUTH while expecting BEGIN");

        case DBUS_AUTH_COMMAND_DATA:
            return send_error (auth, "Sent DATA while expecting BEGIN");

        case DBUS_AUTH_COMMAND_BEGIN:
            goto_state (auth, &common_state_authenticated);
            return TRUE;

        case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
            if (auth->unix_fd_possible)
                return send_agree_unix_fd(auth);
            else
                return send_error(auth, "Unix FD passing not supported, not
authenticated or otherwise not possible");

        case DBUS_AUTH_COMMAND_REJECTED:
        case DBUS_AUTH_COMMAND_OK:
    }
}

```

```

    case DBUS_AUTH_COMMAND_UNKNOWN:
    case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
    default:
        return send_error (auth, "Unknown command");

    case DBUS_AUTH_COMMAND_CANCEL:
    case DBUS_AUTH_COMMAND_ERROR:
        return send_rejected (auth);
    }
}

/* return FALSE if no memory, TRUE if all OK */
static dbus_bool_t
get_word (const DBusString *str,
          int *start,
          DBusString *word)
{
    int i;

    _dbus_string_skip_blank (str, *start, start);
    _dbus_string_find_blank (str, *start, &i);

    if (i > *start)
    {
        if (!_dbus_string_copy_len (str, *start, i - *start, word, 0))
            return FALSE;

        *start = i;
    }

    return TRUE;
}

static dbus_bool_t
record_mechanisms (DBusAuth *auth,
                  const DBusString *args)
{
    int next;
    int len;

    if (auth->already_got_mechanisms)
        return TRUE;

    len = _dbus_string_get_length (args);

    next = 0;
    while (next < len)
    {
        DBusString m;
        const DBusAuthMechanismHandler *mech;

        if (!_dbus_string_init (&m))

```

```

    goto nomem;

if (!get_word (args, &next, &m))
{
    _dbus_string_free (&m);
    goto nomem;
}

mech = find_mech (&m, auth->allowed_mechs);

if (mech != NULL)
{
    /* FIXME right now we try mechanisms in the order
    * the server lists them; should we do them in
    * some more deterministic order?
    *
    * Probably in all_mechanisms order, our order of
    * preference. Of course when the server is us,
    * it lists things in that order anyhow.
    */

    if (mech != &all_mechanisms[0])
    {
        try\n",
        _dbus_verbose ("%s: Adding mechanism %s to list we will
        DBUS_AUTH_NAME (auth), mech->mechanism);

        if (!_dbus_list_append (& DBUS_AUTH_CLIENT (auth)-
        >mechs_to_try,
        (void*) mech))
        {
            _dbus_string_free (&m);
            goto nomem;
        }
    }
    else
    {
        adding to list we will try\n",
        DBUS_AUTH_NAME (auth), mech->mechanism);
    }
}
else
{
    don't know how to use\n",
    DBUS_AUTH_NAME (auth),
    _dbus_string_get_const_data (&m));
}

_dbus_string_free (&m);
}

```

```

auth->already_got_mechanisms = TRUE;

return TRUE;

nomen:
_dbus_list_clear (& DBUS_AUTH_CLIENT (auth)->mechs_to_try);

return FALSE;
}

static dbus_bool_t
process_rejected (DBusAuth *auth, const DBusString *args)
{
    const DBusAuthMechanismHandler *mech;
    DBusAuthClient *client;

    client = DBUS_AUTH_CLIENT (auth);

    if (!auth->already_got_mechanisms)
    {
        if (!record_mechanisms (auth, args))
            return FALSE;
    }

    if (DBUS_AUTH_CLIENT (auth)->mechs_to_try != NULL)
    {
        mech = client->mechs_to_try->data;

        if (!send_auth (auth, mech))
            return FALSE;

        _dbus_list_pop_first (&client->mechs_to_try);

        _dbus_verbose ("%s: Trying mechanism %s\n",
                       DBUS_AUTH_NAME (auth),
                       mech->mechanism);
    }
    else
    {
        /* Give up */
        _dbus_verbose ("%s: Disconnecting because we are out of
mechanisms to try using\n",
                       DBUS_AUTH_NAME (auth));
        goto_state (auth, &common_state_need_disconnect);
    }

    return TRUE;
}

static dbus_bool_t

```

```

handle_client_state_waiting_for_data (DBusAuth          *auth,
                                       DBusAuthCommand  command,
                                       const DBusString *args)
{
    _dbus_assert (auth->mech != NULL);

    switch (command)
    {
        case DBUS_AUTH_COMMAND_DATA:
            return process_data (auth, args, auth->mech->client_data_func);

        case DBUS_AUTH_COMMAND_REJECTED:
            return process_rejected (auth, args);

        case DBUS_AUTH_COMMAND_OK:
            return process_ok(auth, args);

        case DBUS_AUTH_COMMAND_ERROR:
            return send_cancel (auth);

        case DBUS_AUTH_COMMAND_AUTH:
        case DBUS_AUTH_COMMAND_CANCEL:
        case DBUS_AUTH_COMMAND_BEGIN:
        case DBUS_AUTH_COMMAND_UNKNOWN:
        case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
        case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
        default:
            return send_error (auth, "Unknown command");
    }
}

```

```

static dbus_bool_t
handle_client_state_waiting_for_ok (DBusAuth          *auth,
                                    DBusAuthCommand  command,
                                    const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_REJECTED:
            return process_rejected (auth, args);

        case DBUS_AUTH_COMMAND_OK:
            return process_ok(auth, args);

        case DBUS_AUTH_COMMAND_DATA:
        case DBUS_AUTH_COMMAND_ERROR:
            return send_cancel (auth);

        case DBUS_AUTH_COMMAND_AUTH:
        case DBUS_AUTH_COMMAND_CANCEL:
        case DBUS_AUTH_COMMAND_BEGIN:
        case DBUS_AUTH_COMMAND_UNKNOWN:

```



```

    case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
    case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
    default:
        return send_error (auth, "Unknown command");
    }
}

static dbus_bool_t
handle_client_state_waiting_for_reject (DBusAuth          *auth,
                                       DBusAuthCommand    command,
                                       const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_REJECTED:
            return process_rejected (auth, args);

        case DBUS_AUTH_COMMAND_AUTH:
        case DBUS_AUTH_COMMAND_CANCEL:
        case DBUS_AUTH_COMMAND_DATA:
        case DBUS_AUTH_COMMAND_BEGIN:
        case DBUS_AUTH_COMMAND_OK:
        case DBUS_AUTH_COMMAND_ERROR:
        case DBUS_AUTH_COMMAND_UNKNOWN:
        case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
        case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
        default:
            goto_state (auth, &common_state_need_disconnect);
            return TRUE;
    }
}

static dbus_bool_t
handle_client_state_waiting_for_agree_unix_fd(DBusAuth          *auth,
                                              DBusAuthCommand
command,
                                              const DBusString *args)
{
    switch (command)
    {
        case DBUS_AUTH_COMMAND_AGREE_UNIX_FD:
            _dbus_assert(auth->unix_fd_possible);
            auth->unix_fd_negotiated = TRUE;
            _dbus_verbose("Successfully negotiated UNIX FD passing\n");
            return send_begin (auth);

        case DBUS_AUTH_COMMAND_ERROR:
            _dbus_assert(auth->unix_fd_possible);
            auth->unix_fd_negotiated = FALSE;
            _dbus_verbose("Failed to negotiate UNIX FD passing\n");
            return send_begin (auth);
    }
}

```

```

    case DBUS_AUTH_COMMAND_OK:
    case DBUS_AUTH_COMMAND_DATA:
    case DBUS_AUTH_COMMAND_REJECTED:
    case DBUS_AUTH_COMMAND_AUTH:
    case DBUS_AUTH_COMMAND_CANCEL:
    case DBUS_AUTH_COMMAND_BEGIN:
    case DBUS_AUTH_COMMAND_UNKNOWN:
    case DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD:
    default:
        return send_error (auth, "Unknown command");
    }
}

/**
 * Mapping from command name to enum
 */
typedef struct {
    const char *name;          /**< Name of the command */
    DBusAuthCommand command;  /**< Corresponding enum */
} DBusAuthCommandName;

static const DBusAuthCommandName auth_command_names[] = {
    { "AUTH",          DBUS_AUTH_COMMAND_AUTH },
    { "CANCEL",       DBUS_AUTH_COMMAND_CANCEL },
    { "DATA",         DBUS_AUTH_COMMAND_DATA },
    { "BEGIN",        DBUS_AUTH_COMMAND_BEGIN },
    { "REJECTED",     DBUS_AUTH_COMMAND_REJECTED },
    { "OK",           DBUS_AUTH_COMMAND_OK },
    { "ERROR",        DBUS_AUTH_COMMAND_ERROR },
    { "NEGOTIATE_UNIX_FD", DBUS_AUTH_COMMAND_NEGOTIATE_UNIX_FD },
    { "AGREE_UNIX_FD",  DBUS_AUTH_COMMAND_AGREE_UNIX_FD }
};

static DBusAuthCommand
lookup_command_from_name (DBusString *command)
{
    int i;

    for (i = 0; i < _DBUS_N_ELEMENTS (auth_command_names); i++)
    {
        if (_dbus_string_equal_c_str (command,
                                     auth_command_names[i].name))
            return auth_command_names[i].command;
    }

    return DBUS_AUTH_COMMAND_UNKNOWN;
}

static void
goto_state (DBusAuth *auth,
            const DBusAuthStateData *state)
{

```

```

_dbus_verbose ("%s: going from state %s to state %s\n",
               DBUS_AUTH_NAME (auth),
               auth->state->name,
               state->name);

auth->state = state;
}

/* returns whether to call it again right away */
static dbus_bool_t
process_command (DBusAuth *auth)
{
    DBusAuthCommand command;
    DBusString line;
    DBusString args;
    int eol;
    int i, j;
    dbus_bool_t retval;

    /* _dbus_verbose ("%s: trying process_command()\n"); */

    retval = FALSE;

    eol = 0;
    if (!_dbus_string_find (&auth->incoming, 0, "\r\n", &eol))
        return FALSE;

    if (!_dbus_string_init (&line))
    {
        auth->needed_memory = TRUE;
        return FALSE;
    }

    if (!_dbus_string_init (&args))
    {
        _dbus_string_free (&line);
        auth->needed_memory = TRUE;
        return FALSE;
    }

    if (!_dbus_string_copy_len (&auth->incoming, 0, eol, &line, 0))
        goto out;

    if (!_dbus_string_validate_ascii (&line, 0,
                                     _dbus_string_get_length (&line)))
    {
        _dbus_verbose ("%s: Command contained non-ASCII chars or
embedded nul\n",
                       DBUS_AUTH_NAME (auth));
        if (!send_error (auth, "Command contained non-ASCII"))
            goto out;
        else

```

```

        goto next_command;
    }

    _dbus_verbose ("%s: got command \"%s\"\n",
                  DBUS_AUTH_NAME (auth),
                  _dbus_string_get_const_data (&line));

    _dbus_string_find_blank (&line, 0, &i);
    _dbus_string_skip_blank (&line, i, &j);

    if (j > i)
        _dbus_string_delete (&line, i, j - i);

    if (!_dbus_string_move (&line, i, &args, 0))
        goto out;

    /* FIXME 1.0 we should probably validate that only the allowed
     * chars are in the command name
     */

    command = lookup_command_from_name (&line);
    if (!(* auth->state->handler) (auth, command, &args))
        goto out;

next_command:

    /* We've succeeded in processing the whole command so drop it out
     * of the incoming buffer and return TRUE to try another command.
     */

    _dbus_string_delete (&auth->incoming, 0, eol);

    /* kill the \r\n */
    _dbus_string_delete (&auth->incoming, 0, 2);

    retval = TRUE;

out:
    _dbus_string_free (&args);
    _dbus_string_free (&line);

    if (!retval)
        auth->needed_memory = TRUE;
    else
        auth->needed_memory = FALSE;

    return retval;
}

/** @} */

```

```

/**
 * @addtogroup DBusAuth
 * @{}
 */

/**
 * Creates a new auth conversation object for the server side.
 * See doc/dbus-sasl-profile.txt for full details on what
 * this object does.
 *
 * @returns the new object or #NULL if no memory
 */
DBusAuth*
_dbus_auth_server_new (const DBusString *guid)
{
    DBusAuth *auth;
    DBusAuthServer *server_auth;
    DBusString guid_copy;

    if (!_dbus_string_init (&guid_copy))
        return NULL;

    if (!_dbus_string_copy (guid, 0, &guid_copy, 0))
    {
        _dbus_string_free (&guid_copy);
        return NULL;
    }

    auth = _dbus_auth_new (sizeof (DBusAuthServer));
    if (auth == NULL)
    {
        _dbus_string_free (&guid_copy);
        return NULL;
    }

    auth->side = auth_side_server;
    auth->state = &server_state_waiting_for_auth;

    server_auth = DBUS_AUTH_SERVER (auth);

    server_auth->guid = guid_copy;

    /* perhaps this should be per-mechanism with a lower
     * max
     */
    server_auth->failures = 0;
    server_auth->max_failures = 6;

    return auth;
}

/**

```

```

* Creates a new auth conversation object for the client side.
* See doc/dbus-sasl-profile.txt for full details on what
* this object does.
*
* @returns the new object or #NULL if no memory
*/
DBusAuth*
_dbus_auth_client_new (void)
{
    DBusAuth *auth;
    DBusString guid_str;

    if (!_dbus_string_init (&guid_str))
        return NULL;

    auth = _dbus_auth_new (sizeof (DBusAuthClient));
    if (auth == NULL)
    {
        _dbus_string_free (&guid_str);
        return NULL;
    }

    DBUS_AUTH_CLIENT (auth)->guid_from_server = guid_str;

    auth->side = auth_side_client;
    auth->state = &client_state_need_send_auth;

    /* Start the auth conversation by sending AUTH for our default
     * mechanism */
    if (!send_auth (auth, &all_mechanisms[0]))
    {
        _dbus_auth_unref (auth);
        return NULL;
    }

    return auth;
}

/**
 * Increments the refcount of an auth object.
 *
 * @param auth the auth conversation
 * @returns the auth conversation
 */
DBusAuth *
_dbus_auth_ref (DBusAuth *auth)
{
    _dbus_assert (auth != NULL);

    auth->refcount += 1;

    return auth;
}

```

```

}

/**
 * Decrements the refcount of an auth object.
 *
 * @param auth the auth conversation
 */
void
_dbus_auth_unref (DBusAuth *auth)
{
    _dbus_assert (auth != NULL);
    _dbus_assert (auth->refcount > 0);

    auth->refcount -= 1;
    if (auth->refcount == 0)
    {
        shutdown_mech (auth);

        if (DBUS_AUTH_IS_CLIENT (auth))
        {
            _dbus_string_free (& DBUS_AUTH_CLIENT (auth)-
>guid_from_server);
            _dbus_list_clear (& DBUS_AUTH_CLIENT (auth)->mechs_to_try);
        }
        else
        {
            _dbus_assert (DBUS_AUTH_IS_SERVER (auth));

            _dbus_string_free (& DBUS_AUTH_SERVER (auth)->guid);
        }

        if (auth->keyring)
            _dbus_keyring_unref (auth->keyring);

        _dbus_string_free (&auth->context);
        _dbus_string_free (&auth->challenge);
        _dbus_string_free (&auth->identity);
        _dbus_string_free (&auth->incoming);
        _dbus_string_free (&auth->outgoing);

        dbus_free_string_array (auth->allowed_mechs);

        _dbus_credentials_unref (auth->credentials);
        _dbus_credentials_unref (auth->authorized_identity);
        _dbus_credentials_unref (auth->desired_identity);

        dbus_free (auth);
    }
}

/**
 * Sets an array of authentication mechanism names

```

```

* that we are willing to use.
*
* @param auth the auth conversation
* @param mechanisms #NULL-terminated array of mechanism names
* @returns #FALSE if no memory
*/
dbus_bool_t
_dbus_auth_set_mechanisms (DBusAuth *auth,
                          const char **mechanisms)
{
    char **copy;

    if (mechanisms != NULL)
    {
        copy = _dbus_dup_string_array (mechanisms);
        if (copy == NULL)
            return FALSE;
    }
    else
        copy = NULL;

    dbus_free_string_array (auth->allowed_mechs);

    auth->allowed_mechs = copy;

    return TRUE;
}

/**
 * @param auth the auth conversation object
 * @returns #TRUE if we're in a final state
 */
#define DBUS_AUTH_IN_END_STATE(auth) ((auth)->state->handler == NULL)

/**
 * Analyzes buffered input and moves the auth conversation forward,
 * returning the new state of the auth conversation.
 *
 * @param auth the auth conversation
 * @returns the new state
 */
DBusAuthState
_dbus_auth_do_work (DBusAuth *auth)
{
    auth->needed_memory = FALSE;

    /* Max amount we'll buffer up before deciding someone's on crack */
#define MAX_BUFFER (16 * _DBUS_ONE_KILOBYTE)

    do
    {
        if (DBUS_AUTH_IN_END_STATE (auth))

```



```

        break;

        if (_dbus_string_get_length (&auth->incoming) > MAX_BUFFER ||
            _dbus_string_get_length (&auth->outgoing) > MAX_BUFFER)
        {
            goto_state (auth, &common_state_need_disconnect);
            _dbus_verbose ("%s: Disconnecting due to excessive data
buffered in auth phase\n",
                        DBUS_AUTH_NAME (auth));
            break;
        }
    }
    while (process_command (auth));

    if (auth->needed_memory)
        return DBUS_AUTH_STATE_WAITING_FOR_MEMORY;
    else if (_dbus_string_get_length (&auth->outgoing) > 0)
        return DBUS_AUTH_STATE_HAVE_BYTES_TO_SEND;
    else if (auth->state == &common_state_need_disconnect)
        return DBUS_AUTH_STATE_NEED_DISCONNECT;
    else if (auth->state == &common_state_authenticated)
        return DBUS_AUTH_STATE_AUTHENTICATED;
    else return DBUS_AUTH_STATE_WAITING_FOR_INPUT;
}

/**
 * Gets bytes that need to be sent to the peer we're conversing with.
 * After writing some bytes, _dbus_auth_bytes_sent() must be called
 * to notify the auth object that they were written.
 *
 * @param auth the auth conversation
 * @param str return location for a ref to the buffer to send
 * @returns #FALSE if nothing to send
 */
dbus_bool_t
_dbus_auth_get_bytes_to_send (DBusAuth *auth,
                             const DBusString **str)
{
    _dbus_assert (auth != NULL);
    _dbus_assert (str != NULL);

    *str = NULL;

    if (_dbus_string_get_length (&auth->outgoing) == 0)
        return FALSE;

    *str = &auth->outgoing;

    return TRUE;
}

/**

```

```

* Notifies the auth conversation object that
* the given number of bytes of the outgoing buffer
* have been written out.
*
* @param auth the auth conversation
* @param bytes_sent number of bytes written out
*/
void
_dbus_auth_bytes_sent (DBusAuth *auth,
                      int        bytes_sent)
{
    _dbus_verbose ("%s: Sent %d bytes of: %s\n",
                  DBUS_AUTH_NAME (auth),
                  bytes_sent,
                  _dbus_string_get_const_data (&auth->outgoing));

    _dbus_string_delete (&auth->outgoing,
                        0, bytes_sent);
}

/**
 * Get a buffer to be used for reading bytes from the peer we're
 * conversing
 * with. Bytes should be appended to this buffer.
 *
 * @param auth the auth conversation
 * @param buffer return location for buffer to append bytes to
 */
void
_dbus_auth_get_buffer (DBusAuth *auth,
                      DBusString **buffer)
{
    _dbus_assert (auth != NULL);
    _dbus_assert (!auth->buffer_outstanding);

    *buffer = &auth->incoming;

    auth->buffer_outstanding = TRUE;
}

/**
 * Returns a buffer with new data read into it.
 *
 * @param auth the auth conversation
 * @param buffer the buffer being returned
 * @param bytes_read number of new bytes added
 */
void
_dbus_auth_return_buffer (DBusAuth *auth,
                         DBusString *buffer,
                         int        bytes_read)
{

```

```

    _dbus_assert (buffer == &auth->incoming);
    _dbus_assert (auth->buffer_outstanding);

    auth->buffer_outstanding = FALSE;
}

/**
 * Returns leftover bytes that were not used as part of the auth
 * conversation.  These bytes will be part of the message stream
 * instead.  This function may not be called until authentication has
 * succeeded.
 *
 * @param auth the auth conversation
 * @param str return location for pointer to string of unused bytes
 */
void
_dbus_auth_get_unused_bytes (DBusAuth *auth,
                             const DBusString **str)
{
    if (!DBUS_AUTH_IN_END_STATE (auth))
        return;

    *str = &auth->incoming;
}

/**
 * Gets rid of unused bytes returned by _dbus_auth_get_unused_bytes()
 * after we've gotten them and successfully moved them elsewhere.
 *
 * @param auth the auth conversation
 */
void
_dbus_auth_delete_unused_bytes (DBusAuth *auth)
{
    if (!DBUS_AUTH_IN_END_STATE (auth))
        return;

    _dbus_string_set_length (&auth->incoming, 0);
}

/**
 * Called post-authentication, indicates whether we need to encode
 * the message stream with _dbus_auth_encode_data() prior to
 * sending it to the peer.
 *
 * @param auth the auth conversation
 * @returns #TRUE if we need to encode the stream
 */
dbus_bool_t
_dbus_auth_needs_encoding (DBusAuth *auth)
{

```

```

if (auth->state != &common_state_authenticated)
    return FALSE;

if (auth->mech != NULL)
{
    if (DBUS_AUTH_IS_CLIENT (auth))
        return auth->mech->client_encode_func != NULL;
    else
        return auth->mech->server_encode_func != NULL;
}
else
    return FALSE;
}

/**
 * Called post-authentication, encodes a block of bytes for sending to
 * the peer. If no encoding was negotiated, just copies the bytes
 * (you can avoid this by checking _dbus_auth_needs_encoding()).
 *
 * @param auth the auth conversation
 * @param plaintext the plain text data
 * @param encoded initialized string to where encoded data is appended
 * @returns #TRUE if we had enough memory and successfully encoded
 */
dbus_bool_t
_dbus_auth_encode_data (DBusAuth      *auth,
                       const DBusString *plaintext,
                       DBusString      *encoded)
{
    _dbus_assert (plaintext != encoded);

    if (auth->state != &common_state_authenticated)
        return FALSE;

    if (_dbus_auth_needs_encoding (auth))
    {
        if (DBUS_AUTH_IS_CLIENT (auth))
            return (* auth->mech->client_encode_func) (auth, plaintext,
encoded);
        else
            return (* auth->mech->server_encode_func) (auth, plaintext,
encoded);
    }
    else
    {
        return _dbus_string_copy (plaintext, 0, encoded,
                                _dbus_string_get_length (encoded));
    }
}

/**
 * Called post-authentication, indicates whether we need to decode

```

```

* the message stream with _dbus_auth_decode_data() after
* receiving it from the peer.
*
* @param auth the auth conversation
* @returns #TRUE if we need to encode the stream
*/
dbus_bool_t
_dbus_auth_needs_decoding (DBusAuth *auth)
{
    if (auth->state != &common_state_authenticated)
        return FALSE;

    if (auth->mech != NULL)
    {
        if (DBUS_AUTH_IS_CLIENT (auth))
            return auth->mech->client_decode_func != NULL;
        else
            return auth->mech->server_decode_func != NULL;
    }
    else
        return FALSE;
}

/**
 * Called post-authentication, decodes a block of bytes received from
 * the peer. If no encoding was negotiated, just copies the bytes (you
 * can avoid this by checking _dbus_auth_needs_decoding()).
 *
 * @todo 1.0? We need to be able to distinguish "out of memory" error
 * from "the data is hosed" error.
 *
 * @param auth the auth conversation
 * @param encoded the encoded data
 * @param plaintext initialized string where decoded data is appended
 * @returns #TRUE if we had enough memory and successfully decoded
 */
dbus_bool_t
_dbus_auth_decode_data (DBusAuth      *auth,
                        const DBusString *encoded,
                        DBusString      *plaintext)
{
    _dbus_assert (plaintext != encoded);

    if (auth->state != &common_state_authenticated)
        return FALSE;

    if (_dbus_auth_needs_decoding (auth))
    {
        if (DBUS_AUTH_IS_CLIENT (auth))
            return (* auth->mech->client_decode_func) (auth, encoded,
plaintext);
    }
}

```

```

        else
            return (* auth->mech->server_decode_func) (auth, encoded,
plaintext);
        }
    else
    {
        return _dbus_string_copy (encoded, 0, plaintext,
                                _dbus_string_get_length (plaintext));
    }
}

/**
 * Sets credentials received via reliable means from the operating
 * system.
 *
 * @param auth the auth conversation
 * @param credentials the credentials received
 * @returns #FALSE on OOM
 */
dbus_bool_t
_dbus_auth_set_credentials (DBusAuth          *auth,
                           DBusCredentials   *credentials)
{
    _dbus_credentials_clear (auth->credentials);
    return _dbus_credentials_add_credentials (auth->credentials,
                                             credentials);
}

/**
 * Gets the identity we authorized the client as.  Apps may have
 * different policies as to what identities they allow.
 *
 * Returned credentials are not a copy and should not be modified
 *
 * @param auth the auth conversation
 * @returns the credentials we've authorized BY REFERENCE do not
modify
 */
DBusCredentials*
_dbus_auth_get_identity (DBusAuth          *auth)
{
    if (auth->state == &common_state_authenticated)
    {
        return auth->authorized_identity;
    }
    else
    {
        /* FIXME instead of this, keep an empty credential around that
         * doesn't require allocation or something
         */
        /* return empty credentials */
    }
}

```

```

        _dbus_assert (_dbus_credentials_are_empty (auth-
>authorized_identity));
        return auth->authorized_identity;
    }
}

/**
 * Gets the GUID from the server if we've authenticated; gets
 * #NULL otherwise.
 * @param auth the auth object
 * @returns the GUID in ASCII hex format
 */
const char*
_dbus_auth_get_guid_from_server (DBusAuth *auth)
{
    _dbus_assert (DBUS_AUTH_IS_CLIENT (auth));

    if (auth->state == &common_state_authenticated)
        return _dbus_string_get_const_data (& DBUS_AUTH_CLIENT (auth)-
>guid_from_server);
    else
        return NULL;
}

/**
 * Sets the "authentication context" which scopes cookies
 * with the DBUS_COOKIE_SHA1 auth mechanism for example.
 *
 * @param auth the auth conversation
 * @param context the context
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_auth_set_context (DBusAuth          *auth,
                       const DBusString  *context)
{
    return _dbus_string_replace_len (context, 0, _dbus_string_get_length
(context),
                                   &auth->context, 0,
_dbus_string_get_length (context));
}

/**
 * Sets whether unix fd passing is potentially on the transport and
 * hence shall be negotiated.
 *
 * @param auth the auth conversation
 * @param b TRUE when unix fd passing shall be negotiated, otherwise
FALSE
 */
void
_dbus_auth_set_unix_fd_possible (DBusAuth *auth, dbus_bool_t b)

```

```

{
    auth->unix_fd_possible = b;
}

/**
 * Queries whether unix fd passing was successfully negotiated.
 *
 * @param auth the auth conversion
 * @returns #TRUE when unix fd passing was negotiated.
 */
dbus_bool_t
_dbus_auth_get_unix_fd_negotiated(DBusAuth *auth)
{
    return auth->unix_fd_negotiated;
}

/** @} */

/* tests in dbus-auth-util.c */

File = dbus-auth.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-auth.h Authentication
 *
 * Copyright (C) 2002 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#endif DBUS_AUTH_H
#define DBUS_AUTH_H

```



```

#include <dbus/dbus-macros.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-sysdeps.h>

DBUS_BEGIN_DECLS

typedef struct DBusAuth DBusAuth;

typedef enum
{
    DBUS_AUTH_STATE_WAITING_FOR_INPUT,
    DBUS_AUTH_STATE_WAITING_FOR_MEMORY,
    DBUS_AUTH_STATE_HAVE_BYTES_TO_SEND,
    DBUS_AUTH_STATE_NEED_DISCONNECT,
    DBUS_AUTH_STATE_AUTHENTICATED
} DBusAuthState;

DBusAuth*      _dbus_auth_server_new      (const DBusString
*guid);
DBusAuth*      _dbus_auth_client_new      (void);
DBusAuth*      _dbus_auth_ref            (DBusAuth
*auth);
void           _dbus_auth_unref          (DBusAuth
*auth);
dbus_bool_t    _dbus_auth_set_mechanisms  (DBusAuth
*auth,
                                           const char
**mechanisms);
DBusAuthState _dbus_auth_do_work         (DBusAuth
*auth);
dbus_bool_t    _dbus_auth_get_bytes_to_send (DBusAuth
*auth,
                                           const DBusString
**str);
void           _dbus_auth_bytes_sent      (DBusAuth
*auth,
                                           int
bytes_sent);
void           _dbus_auth_get_buffer      (DBusAuth
*auth,
                                           DBusString
**buffer);
void           _dbus_auth_return_buffer   (DBusAuth
*auth,
                                           DBusString
*buffer,
                                           int
bytes_read);
void           _dbus_auth_get_unused_bytes (DBusAuth
*auth,

```

```

const DBusString
**str);
void      _dbus_auth_delete_unused_bytes (DBusAuth
*auth);
dbus_bool_t  _dbus_auth_needs_encoding      (DBusAuth
*auth);
dbus_bool_t  _dbus_auth_encode_data      (DBusAuth
*auth,
const DBusString
*plaintext,
DBusString
*encoded);
dbus_bool_t  _dbus_auth_needs_decoding      (DBusAuth
*auth);
dbus_bool_t  _dbus_auth_decode_data      (DBusAuth
*auth,
const DBusString
*encoded,
DBusString
*plaintext);
dbus_bool_t  _dbus_auth_set_credentials      (DBusAuth
*auth,
DBusCredentials
*credentials);
DBusCredentials* _dbus_auth_get_identity      (DBusAuth
*auth);
dbus_bool_t  _dbus_auth_set_context      (DBusAuth
*auth,
const DBusString
*context);
const char*  _dbus_auth_get_guid_from_server (DBusAuth
*auth);

void      _dbus_auth_set_unix_fd_possible (DBusAuth
*auth, dbus_bool_t b);
dbus_bool_t  _dbus_auth_get_unix_fd_negotiated (DBusAuth
*auth);

```

```
DBUS_END_DECLS
```

```
#endif /* DBUS_AUTH_H */
```

```
File = dbus-bash-completion.sh.in
```

```
# Check for bash
[ -z "$BASH_VERSION" ] && return
```

```
#####
#####
```

```

__dbus_send() {
    local IFS=$'\n'
    local cur="${COMP_WORDS[COMP_CWORD]}"

    # --name=value style option
    if [[ "$cur" == *=* ]] ; then
        cur=${cur/*=/}
    fi

    COMPREPLY=( $(compgen -W "$(@libexecdir@/dbus-bash-completion-
helper dbus-send ${COMP_WORDS[@]:0})" -- $cur) )
}

#####
#####

complete -o nospace -F __dbus_send dbus-send

File = dbus-binding-tool-glib.c

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-binding-tool-glib.c: Output C glue
 *
 * Copyright (C) 2003, 2004, 2005 Red Hat, Inc.
 * Copyright (C) 2005 Nokia
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

#include <config.h>
#include "dbus/dbus-glib.h"

```

```

#include "dbus-gidl.h"
#include "dbus-gparser.h"
#include "dbus-gutils.h"
#include "dbus-gtype-specialized.h"
#include "dbus-gsignature.h"
#include "dbus-gvalue-utils.h"
#include "dbus-glib-tool.h"
#include "dbus-binding-tool-glib.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

/* Remember to grep for ->format_version in the code if you change
this,
 * most changes should be in dbus-gobject.c. */
#define FORMAT_VERSION 1

#define MARSHAL_PREFIX "dbus_glib_marshal_"

typedef struct
{
    gboolean ignore_unsupported;
    const char* prefix;
    GIOChannel *channel;

    GError **error;

    GHashTable *generated;
    GString *blob;
    GString *signal_blob;
    GString *property_blob;
    guint count;
} DBusBindingToolCData;

static gboolean gather_marshallers (BaseInfo *base,
DBusBindingToolCData *data, GError **error);
static gboolean generate_glue_toplevel (BaseInfo *base,
DBusBindingToolCData *data, GError **error);
static gboolean generate_glue (BaseInfo *base, DBusBindingToolCData
*data, GError **error);
static gboolean generate_client_glue (BaseInfo *base,
DBusBindingToolCData *data, GError **error);

static const char *
dbus_g_type_get_marshal_name (GType gtype)
{
    switch (G_TYPE_FUNDAMENTAL (gtype))
    {
        {
            case G_TYPE_NONE:
                return "NONE";
            case G_TYPE_BOOLEAN:

```

```

    return "BOOLEAN";
case G_TYPE_UCHAR:
    return "UCHAR";
case G_TYPE_INT:
    return "INT";
case G_TYPE_UINT:
    return "UINT";
case G_TYPE_INT64:
    return "INT64";
case G_TYPE_UINT64:
    return "UINT64";
case G_TYPE_DOUBLE:
    return "DOUBLE";
case G_TYPE_STRING:
    return "STRING";
case G_TYPE_POINTER:
    return "POINTER";
case G_TYPE_BOXED:
    return "BOXED";
case G_TYPE_OBJECT:
    return "OBJECT";
default:
    return NULL;
}
}

/* This entire function is kind of...ugh. */
static const char *
dbus_g_type_get_c_name (GType gtype)
{
    GType subtype;
    if (dbus_g_type_is_struct (gtype))
        {
            return "GValueArray";
        }
    if (dbus_g_type_is_collection (gtype))
        {
            subtype = dbus_g_type_get_collection_specialization(gtype);
            if (_dbus_g_type_is_fixed (subtype))
                return "GArray";
            else
                return "GPtrArray";
        }

    if (dbus_g_type_is_map (gtype))
        return "GHashTable";

    if (g_type_is_a (gtype, G_TYPE_STRING))
        return "char *";

    /* This one is even more hacky...we get an extra *
    * because G_TYPE_STRV is a G_TYPE_BOXED

```

```

    */
    if (g_type_is_a (gtype, G_TYPE_STRV))
        return "char *";

    if (g_type_is_a (gtype, DBUS_TYPE_G_OBJECT_PATH))
        return "char";

    if (g_type_is_a (gtype, DBUS_TYPE_G_SIGNATURE))
        return "char";

    return g_type_name (gtype);
}

static gboolean
compute_gsignature (MethodInfo *method, GType *rettype, GArray
**params, GError **error)
{
    GSList *elt;
    GType retval_type;
    GArray *ret;
    gboolean is_async;
    const char *arg_type;
    gboolean retval_signals_error;

    is_async = method_info_get_annotation (method,
DBUS_GLIB_ANNOTATION_ASYNC) != NULL;
    retval_signals_error = FALSE;

    ret = g_array_new (TRUE, TRUE, sizeof (GType));

    if (is_async)
        retval_type = G_TYPE_NONE;
    else
    {
        gboolean found_retval;

        /* Look for return value */
        found_retval = FALSE;
        for (elt = method_info_get_args (method); elt; elt = elt->next)
        {
            ArgInfo *arg = elt->data;
            const char *returnval_annotation;

            returnval_annotation = arg_info_get_annotation (arg,
DBUS_GLIB_ANNOTATION_RETURNVAL);
            if (returnval_annotation != NULL)
            {
                arg_type = arg_info_get_type (arg);
                retval_type = _dbus_gtype_from_signature (arg_type, FALSE);
                if (retval_type == G_TYPE_INVALID)
                    goto invalid_type;
                found_retval = TRUE;
            }
        }
    }
}

```

```

        if (!strcmp (returnval_annotation, "error"))
            retval_signals_error = TRUE;
        break;
    }
}
if (!found_retval)
{
    retval_type = G_TYPE_BOOLEAN;
    retval_signals_error = TRUE;
}
}

*rettype = retval_type;

/* Handle all input arguments */
for (elt = method_info_get_args (method); elt; elt = elt->next)
{
    ArgInfo *arg = elt->data;
    if (arg_info_get_direction (arg) == ARG_IN)
    {
        GType gtype;

        arg_type = arg_info_get_type (arg);
        gtype = _dbus_gtype_from_signature (arg_type, FALSE);
        if (gtype == G_TYPE_INVALID)
            goto invalid_type;

        g_array_append_val (ret, gtype);
    }
}

if (!is_async)
{
    /* Append pointer for each out arg storage */
    for (elt = method_info_get_args (method); elt; elt = elt->next)
    {
        ArgInfo *arg = elt->data;

        /* Skip return value */
        if (arg_info_get_annotation (arg,
            DBUS_GLIB_ANNOTATION_RETURNVAL) != NULL)
            continue;

        if (arg_info_get_direction (arg) == ARG_OUT)
        {
            GType gtype;
            arg_type = arg_info_get_type (arg);
            gtype = _dbus_gtype_from_signature (arg_type, FALSE);
            if (gtype == G_TYPE_INVALID)
                goto invalid_type;
            /* We actually just need a pointer for the return value
            storage */

```

```

        gtype = G_TYPE_POINTER;
        g_array_append_val (ret, gtype);
    }
}

if (retval_signals_error)
{
    /* Final GError parameter */
    GType gtype = G_TYPE_POINTER;
    g_array_append_val (ret, gtype);
}
}
else
{
    /* Context pointer */
    GType gtype = G_TYPE_POINTER;
    g_array_append_val (ret, gtype);
}

*params = ret;
return TRUE;

invalid_type:
g_set_error (error,
            DBUS_BINDING_TOOL_ERROR,
            DBUS_BINDING_TOOL_ERROR_UNSUPPORTED_CONVERSION,
            "Unsupported conversion from D-BUS type %s to glib-
genmarshal type",
            arg_type);
return FALSE;
}

static char *
compute_marshall (MethodInfo *method, GError **error)
{
    GArray *signature;
    GType rettype;
    const char *marshal_name;
    GString *ret;
    guint i;

    if (!compute_gsignature (method, &rettype, &signature, error))
        return NULL;

    ret = g_string_new ("");
    marshal_name = dbus_g_type_get_marshal_name (rettype);
    g_assert (marshal_name != NULL);
    g_string_append (ret, marshal_name);
    g_string_append_c (ret, ':');
    for (i = 0; i < signature->len; i++)
    {

```



```

        marshal_name = dbus_g_type_get_marshal_name (g_array_index
(signature, GType, i));
        g_assert (marshal_name != NULL);
        g_string_append (ret, marshal_name);
        if (i < signature->len - 1)
            g_string_append_c (ret, ',');
    }
    if (signature->len == 0)
    {
        marshal_name = dbus_g_type_get_marshal_name (G_TYPE_NONE);
        g_assert (marshal_name != NULL);
        g_string_append (ret, marshal_name);
    }
    g_array_free (signature, TRUE);
    return g_string_free (ret, FALSE);
}

```

```

static char *
compute_marshall_name (MethodInfo *method, const char *prefix,
GError **error)
{
    GString *ret;
    GArray *signature;
    GType rettype;
    const char *marshal_name;
    guint i;

    if (!compute_gsignature (method, &rettype, &signature, error))
        return NULL;

    ret = g_string_new (MARSHAL_PREFIX);
    g_string_append (ret, prefix);
    g_string_append_c (ret, '_');

    marshal_name = dbus_g_type_get_marshal_name (rettype);
    g_assert (marshal_name != NULL);
    g_string_append (ret, marshal_name);
    g_string_append (ret, "__");
    for (i = 0; i < signature->len; i++)
    {
        marshal_name = dbus_g_type_get_marshal_name (g_array_index
(signature, GType, i));
        g_assert (marshal_name != NULL);
        g_string_append (ret, marshal_name);
        if (i < signature->len - 1)
            g_string_append_c (ret, '_');
    }
    if (signature->len == 0)
    {
        marshal_name = dbus_g_type_get_marshal_name (G_TYPE_NONE);
        g_assert (marshal_name != NULL);
        g_string_append (ret, marshal_name);
    }
}

```

```

    }
    g_array_free (signature, TRUE);
    return g_string_free (ret, FALSE);
}

static gboolean
gather_marshallers_list (GSLList *list, DBusBindingToolCData *data,
GError **error)
{
    GSLList *tmp;

    tmp = list;
    while (tmp != NULL)
    {
        if (!gather_marshallers (tmp->data, data, error))
            return FALSE;
        tmp = tmp->next;
    }
    return TRUE;
}

static gboolean
gather_marshallers (BaseInfo *base, DBusBindingToolCData *data, GError
**error)
{
    if (base_info_get_type (base) == INFO_TYPE_NODE)
    {
        if (!gather_marshallers_list (node_info_get_nodes ((NodeInfo *)
base),
                                data, error))
            return FALSE;
        if (!gather_marshallers_list (node_info_get_interfaces
((NodeInfo *) base),
                                data, error))
            return FALSE;
    }
    else
    {
        InterfaceInfo *interface;
        GSLList *methods;
        GSLList *tmp;
        const char *interface_c_name;

        interface = (InterfaceInfo *) base;
        interface_c_name = interface_info_get_annotation (interface,
DBUS_GLIB_ANNOTATION_C_SYMBOL);
        if (interface_c_name == NULL)
        {
            if (!data->prefix)
                return TRUE;
        }
    }
}

```

```

methods = interface_info_get_methods (interface);

/* Generate the necessarymarshallers for the methods. */
for (tmp = methods; tmp != NULL; tmp = g_slist_next (tmp))
{
    MethodInfo *method;
    char *marshaller_name;

    method = (MethodInfo *) tmp->data;

    marshaller_name = compute_marshallers (method, error);
    if (!marshaller_name)
        return FALSE;

    if (g_hash_table_lookup (data->generated, marshaller_name))
    {
        g_free (marshaller_name);
        continue;
    }

    g_hash_table_insert (data->generated, marshaller_name, NULL);
}

}
return TRUE;
}

static gboolean
generate_glue_list (GSList *list, DBusBindingToolCData *data, GError
**error)
{
    GSList *tmp;

    tmp = list;
    while (tmp != NULL)
    {
        if (!generate_glue (tmp->data, data, error))
            return FALSE;
        tmp = tmp->next;
    }
    return TRUE;
}

#define WRITE_OR_LOSE(x) do { gsize bytes_written; if
(!g_io_channel_write_chars (channel, x, -1, &bytes_written, error))
goto io_lose; } while (0)

static gboolean
write_printf_to_iochannel (const char *fmt, GIOChannel *channel,
GError **error, ...)
{

```

```

char *str;
va_list args;
GIOStatus status;
gsize written;
gboolean ret;

va_start (args, error);

str = g_strdup_vprintf (fmt, args);
if ((status = g_io_channel_write_chars (channel, str, -1, &written,
error)) == G_IO_STATUS_NORMAL)
    ret = TRUE;
else
    ret = FALSE;

g_free (str);

va_end (args);

return ret;
}

static gboolean
write_quoted_string (GIOChannel *channel, GString *string, GError
**error)
{
    guint i;

    WRITE_OR_LOSE ("\"");
    for (i = 0; i < string->len; i++)
        {
            if (string->str[i] != '\\0')
                {
                    if (!g_io_channel_write_chars (channel, string->str + i, 1,
NULL, error))
                        return FALSE;
                }
            else
                {
                    if (!g_io_channel_write_chars (channel, "\\0", -1, NULL,
error))
                        return FALSE;
                }
        }
    WRITE_OR_LOSE ("\\0\"");
    return TRUE;
io_lose:
    return FALSE;
}

static gboolean

```

```

generate_glue_toplevel (BaseInfo *base, DBusBindingToolCData *data,
GError **error)
{
    GString *object_introspection_data_blob;
    GIOChannel *channel;

    channel = data->channel;

    object_introspection_data_blob = g_string_new_len ("", 0);
    data->blob = object_introspection_data_blob;
    data->count = 0;

    data->signal_blob = g_string_new_len ("", 0);
    data->property_blob = g_string_new_len ("", 0);

    if (!write_printf_to_iochannel ("static const DBusGMethodInfo
dbus_glib_%s_methods[] = {\n", channel, error, data->prefix))
        goto io_lose;

    if (!generate_glue_list (node_info_get_nodes ((NodeInfo *) base),
                            data, error))
        return FALSE;
    if (!generate_glue_list (node_info_get_interfaces ((NodeInfo *)
base),
                            data, error))
        return FALSE;

    WRITE_OR_LOSE ("};\n\n");
    /* Information about the object. */

    if (!write_printf_to_iochannel ("const DBusGObjectInfo
dbus_glib_%s_object_info = { %d,\n",
                                channel, error, data->prefix,
FORMAT_VERSION))
        goto io_lose;
    if (!write_printf_to_iochannel ("  dbus_glib_%s_methods,\n",
channel, error, data->prefix))
        goto io_lose;
    if (!write_printf_to_iochannel ("  %d,\n", channel, error, data-
>count))
        goto io_lose;
    if (!write_quoted_string (channel, object_introspection_data_blob,
error))
        goto io_lose;
    WRITE_OR_LOSE (" ,\n");
    if (!write_quoted_string (channel, data->signal_blob, error))
        goto io_lose;
    WRITE_OR_LOSE (" ,\n");
    if (!write_quoted_string (channel, data->property_blob, error))
        goto io_lose;
    WRITE_OR_LOSE (" \n};\n\n");
    g_string_free (object_introspection_data_blob, TRUE);

```

```

    g_string_free (data->signal_blob, TRUE);
    g_string_free (data->property_blob, TRUE);
    data->signal_blob = NULL;
    data->property_blob = NULL;
    return TRUE;
io_lose:
    return FALSE;
}

static gboolean
generate_glue (BaseInfo *base, DBusBindingToolCData *data, GError
**error)
{
    if (base_info_get_type (base) == INFO_TYPE_NODE)
    {
        if (!generate_glue_list (node_info_get_nodes ((NodeInfo *)
base),
                                data, error))
            return FALSE;
        if (!generate_glue_list (node_info_get_interfaces ((NodeInfo *)
base),
                                data, error))
            return FALSE;
    }
    else
    {
        GIOChannel *channel;
        InterfaceInfo *interface;
        GSList *methods;
        GSList *signals;
        GSList *properties;
        GSList *tmp;
        const char *interface_c_name;
        GString *object_introspection_data_blob;

        channel = data->channel;
        object_introspection_data_blob = data->blob;

        interface = (InterfaceInfo *) base;
        interface_c_name = interface_info_get_annotation (interface,
DBUS_GLIB_ANNOTATION_C_SYMBOL);
        if (interface_c_name == NULL)
        {
            if (data->prefix == NULL)
                return TRUE;
            interface_c_name = data->prefix;
        }

        methods = interface_info_get_methods (interface);

        /* Table of marshalled methods. */

```

```

for (tmp = methods; tmp != NULL; tmp = g_slist_next (tmp))
{
    MethodInfo *method;
    char *marshaller_name;
    char *method_c_name;
    gboolean async = FALSE;
    GSList *args;
    gboolean found_retval = FALSE;
    guint found_out_args = 0;

    method = (MethodInfo *) tmp->data;
    method_c_name = g_strdup (method_info_get_annotation (method,
DBUS_GLIB_ANNOTATION_C_SYMBOL));
    if (method_c_name == NULL)
    {
        char *method_name_uscored;
        method_name_uscored = _dbus_gutils_wincaps_to_uscore
(method_info_get_name (method));
        method_c_name = g_strdup_printf ("%s_%s",
interface_c_name,
method_name_uscored);
        g_free (method_name_uscored);
    }

    if (!write_printf_to_iochannel (" { (GCallback) %s, ",
channel, error,
method_c_name))
    {
        g_free (method_c_name);
        goto io_lose;
    }
    g_free (method_c_name);

    marshaller_name = compute_marshall_name (method, data-
>prefix, error);
    if (!marshaller_name)
        goto io_lose;

    if (!write_printf_to_iochannel ("%s, %d },\n", channel,
error,
marshaller_name,
object_introspection_data_blob->len))
    {
        g_free (marshaller_name);
        goto io_lose;
    }

    if (method_info_get_annotation (method,
DBUS_GLIB_ANNOTATION_ASYNC) != NULL)
        async = TRUE;

    /* Object method data blob format:

```

```

    * <iface>\0<name>\0(<argname>\0<argdirection>\0<argtype>\0)*\0
    */

    g_string_append (object_introspection_data_blob,
interface_info_get_name (interface));
    g_string_append_c (object_introspection_data_blob, '\0');

    g_string_append (object_introspection_data_blob,
method_info_get_name (method));
    g_string_append_c (object_introspection_data_blob, '\0');

    g_string_append_c (object_introspection_data_blob, async ? 'A'
: 'S');
    g_string_append_c (object_introspection_data_blob, '\0');

    for (args = method_info_get_args (method); args; args = args-
>next)
    {
        ArgInfo *arg;
        char direction;
        const char *returnval_annotation;

        arg = args->data;

        g_string_append (object_introspection_data_blob,
arg_info_get_name (arg));
        g_string_append_c (object_introspection_data_blob, '\0');

        switch (arg_info_get_direction (arg))
        {
        case ARG_IN:
            direction = 'I';
            break;
        case ARG_OUT:
            direction = 'O';
            found_out_args++;
            break;
        case ARG_INVALID:
        default:
            g_assert_not_reached ();
            direction = 0; /* silence gcc */
            break;
        }
        g_string_append_c (object_introspection_data_blob,
direction);
        g_string_append_c (object_introspection_data_blob, '\0');

        if (arg_info_get_annotation (arg,
DBUS_GLIB_ANNOTATION_CONST) != NULL)
        {
            if (arg_info_get_direction (arg) == ARG_IN)
            {

```



```

        g_set_error (error,
                    DBUS_BINDING_TOOL_ERROR,
                    DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                    "Input argument \"%s\" cannot have const
annotation in method \"%s\" of interface \"%s\"\\n",
                    arg_info_get_name (arg),
                    method_info_get_name (method),
                    interface_info_get_name (interface));
        return FALSE;
    }
    g_string_append_c (object_introspection_data_blob, 'C');
    g_string_append_c (object_introspection_data_blob, '\\0');
}
else if (arg_info_get_direction (arg) == ARG_OUT)
{
    g_string_append_c (object_introspection_data_blob, 'F');
    g_string_append_c (object_introspection_data_blob, '\\0');
}

    returnval_annotation = arg_info_get_annotation (arg,
DBUS_GLIB_ANNOTATION_RETURNVAL);
    if (returnval_annotation != NULL)
    {
        GType gtype;

        if (found_retval)
        {
            g_set_error (error,
                        DBUS_BINDING_TOOL_ERROR,
                        DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                        "Multiple arguments with return value
annotation in method \"%s\" of interface \"%s\"\\n",
                        method_info_get_name (method),
                        interface_info_get_name (interface));
            return FALSE;
        }
        found_retval = TRUE;
        if (arg_info_get_direction (arg) == ARG_IN)
        {
            g_set_error (error,
                        DBUS_BINDING_TOOL_ERROR,
                        DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                        "Input argument \"%s\" cannot have return
value annotation in method \"%s\" of interface \"%s\"\\n",
                        arg_info_get_name (arg),
                        method_info_get_name (method),
                        interface_info_get_name (interface));
            return FALSE;
        }

        if (found_out_args != 1)
        {

```

```

        g_set_error (error,
                    DBUS_BINDING_TOOL_ERROR,
                    DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                    "An output <arg> after the first cannot have
the ReturnVal annotation, in argument \"%s\" of method \"%s\" of
interface \"%s\"\\n",
                    arg_info_get_name (arg),
                    method_info_get_name (method),
                    interface_info_get_name (interface));
        return FALSE;
    }

    if (!strcmp ("", returnval_annotation))
        g_string_append_c (object_introspection_data_blob,
'R');
    else if (!strcmp ("error", returnval_annotation))
    {
        gtype = _dbus_gtype_from_signature (arg_info_get_type
(arg), TRUE);
        if (!_dbus_gtype_can_signal_error (gtype))
        {
            g_set_error (error,
                        DBUS_BINDING_TOOL_ERROR,
                        DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                        "Output argument \"%s\" cannot signal
error with type \"%s\" in method \"%s\" of interface \"%s\"\\n",
                        arg_info_get_name (arg),
                        g_type_name (gtype),
                        method_info_get_name (method),
                        interface_info_get_name (interface));
            return FALSE;
        }
        g_string_append_c (object_introspection_data_blob,
'E');
    }
    else
    {
        g_set_error (error,
                    DBUS_BINDING_TOOL_ERROR,
                    DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION,
                    "Invalid ReturnVal annotation for argument
\"%s\" in method \"%s\" of interface \"%s\"\\n",
                    arg_info_get_name (arg),
                    method_info_get_name (method),
                    interface_info_get_name (interface));
        return FALSE;
    }

    g_string_append_c (object_introspection_data_blob, '\\0');
}
else if (arg_info_get_direction (arg) == ARG_OUT)

```

```

        {
            g_string_append_c (object_introspection_data_blob, 'N');
            g_string_append_c (object_introspection_data_blob, '\0');
        }

        g_string_append (object_introspection_data_blob,
arg_info_get_type (arg));
            g_string_append_c (object_introspection_data_blob, '\0');
        }

        g_string_append_c (object_introspection_data_blob, '\0');

        data->count++;
    }

signals = interface_info_get_signals (interface);

for (tmp = signals; tmp != NULL; tmp = g_slist_next (tmp))
    {
        SignalInfo *sig;

        sig = tmp->data;

        g_string_append (data->signal_blob, interface_info_get_name
(interface));
        g_string_append_c (data->signal_blob, '\0');
        g_string_append (data->signal_blob, signal_info_get_name
(sig));
        g_string_append_c (data->signal_blob, '\0');
    }

properties = interface_info_get_properties (interface);

for (tmp = properties; tmp != NULL; tmp = g_slist_next (tmp))
    {
        PropertyInfo *prop;
        PropertyAccessFlags access_flags;
        const char *access_string;
        char *uscored;

        prop = tmp->data;

        access_flags = property_info_get_access (prop);
        if ((access_flags & PROPERTY_READ) && (access_flags &
PROPERTY_WRITE))
            access_string = "readwrite";
        else if (access_flags & PROPERTY_READ)
            access_string = "read";
        else if (access_flags & PROPERTY_WRITE)
            access_string = "write";
        else
            continue;
    }

```

```

        /* We append both in the blob so we have to malloc() less
when processing
        * properties */
        uscored = _dbus_gutils_wincaps_to_uscore
(property_info_get_name (prop));

        g_string_append (data->property_blob,
interface_info_get_name (interface));
        g_string_append_c (data->property_blob, '\\0');
        g_string_append (data->property_blob, property_info_get_name
(prop));
        g_string_append_c (data->property_blob, '\\0');
        g_string_append (data->property_blob, uscored);
        g_string_append_c (data->property_blob, '\\0');
        g_string_append (data->property_blob, access_string);
        g_string_append_c (data->property_blob, '\\0');

        g_free (uscored);
    }
}
return TRUE;
io_lose:
return FALSE;
}

static void
write_marshallor (gpointer key, gpointer value, gpointer user_data)
{
    DBusBindingToolCData *data;
    const char *marshaller;
    gsize bytes_written;

    data = user_data;
    marshaller = key;

    if (data->error && *data->error)
        return;

    if (g_io_channel_write_chars (data->channel, marshaller, -1,
&bytes_written, data->error) == G_IO_STATUS_NORMAL)
        g_io_channel_write_chars (data->channel, "\\n", -1, &bytes_written,
data->error);
}

gboolean
dbus_binding_tool_output_glib_server (BaseInfo *info, GIOChannel
*channel, const char *prefix, GError **error)
{
    gboolean ret;
    GPtrArray *argv;
    gint child_stdout;

```

```

GIOChannel *genmarshal_stdout;
GPid child_pid;
DBusBindingToolCData data;
char *tempfile_name;
gint tempfile_fd;
GIOStatus iostatus;
char buf[4096];
gsize bytes_read, bytes_written;

memset (&data, 0, sizeof (data));

dbus_g_type_specialized_init ();
_dbus_g_type_specialized_builtins_init ();

data.prefix = prefix;
data.generated = g_hash_table_new_full (g_str_hash, g_str_equal,
(GDestroyNotify) g_free, NULL);
data.error = error;
genmarshal_stdout = NULL;
tempfile_name = NULL;

if (!gather_marshallers (info, &data, error))
    goto io_lose;

tempfile_fd = g_file_open_tmp ("dbus-binding-tool-c-
marshallers.XXXXXX",
    &tempfile_name, error);
if (tempfile_fd < 0)
    goto io_lose;

data.channel = g_io_channel_unix_new (tempfile_fd);
if (!g_io_channel_set_encoding (data.channel, NULL, error))
    goto io_lose;
g_hash_table_foreach (data.generated, write_marshaller, &data);
if (error && *error != NULL)
{
    ret = FALSE;
    g_io_channel_shutdown (data.channel, TRUE, error);
    g_io_channel_unref (data.channel);
    goto io_lose;
}

g_io_channel_shutdown (data.channel, TRUE, error);
g_io_channel_unref (data.channel);

/* Now spawn glib-genmarshal to insert all our required marshallers
*/
argv = g_ptr_array_new ();
g_ptr_array_add (argv, "glib-genmarshal");
g_ptr_array_add (argv, "--header");
g_ptr_array_add (argv, "--body");
g_ptr_array_add (argv, "--skip-source");

```

```

    g_ptr_array_add (argv, g_strdup_printf ("--prefix=%s%s",
MARSHAL_PREFIX, prefix));
    g_ptr_array_add (argv, tempfile_name);
    g_ptr_array_add (argv, NULL);
    if (!g_spawn_async_with_pipes (NULL, (char**)argv->pdata, NULL,
        G_SPAWN_SEARCH_PATH,
        NULL, NULL,
        &child_pid,
        NULL,
        &child_stdout, NULL, error))
    {
        g_ptr_array_free (argv, TRUE);
        goto io_lose;
    }
    g_ptr_array_free (argv, TRUE);

    genmarshal_stdout = g_io_channel_unix_new (child_stdout);
    if (!g_io_channel_set_encoding (genmarshal_stdout, NULL, error))
        goto io_lose;

    WRITE_OR_LOSE ("/* Generated by dbus-binding-tool; do not edit!
*/\n\n");

    while ((iostatus = g_io_channel_read_chars (genmarshal_stdout, buf,
sizeof (buf),
        &bytes_read, error)) ==
G_IO_STATUS_NORMAL)
        if (g_io_channel_write_chars (channel, buf, bytes_read,
&bytes_written, error) != G_IO_STATUS_NORMAL)
            goto io_lose;
    if (iostatus != G_IO_STATUS_EOF)
        goto io_lose;

    g_io_channel_shutdown (genmarshal_stdout, TRUE, error);

    WRITE_OR_LOSE ("#include <dbus/dbus-glib.h>\n");

    data.channel = channel;
    g_io_channel_ref (data.channel);
    if (!generate_glue_toplevel (info, &data, error))
        goto io_lose;

    ret = TRUE;
cleanup:
    if (tempfile_name)
        unlink (tempfile_name);
    g_free (tempfile_name);
    if (genmarshal_stdout)
        g_io_channel_unref (genmarshal_stdout);
    if (data.channel)
        g_io_channel_unref (data.channel);
    g_hash_table_destroy (data.generated);

```

```

    return ret;
io_lose:
    ret = FALSE;
    goto cleanup;
}

static char *
iface_to_c_prefix (const char *iface)
{
    char **components;
    char **component;
    GString *ret;
    gboolean first;

    components = g_strsplit (iface, ".", 0);

    first = TRUE;
    ret = g_string_new ("");
    for (component = components; *component; component++)
    {
        if (!first)
            g_string_append_c (ret, '_');
        else
            first = FALSE;
        g_string_append (ret, *component);
    }
    g_strfreev (components);
    return g_string_free (ret, FALSE);
}

static char *
compute_client_method_name (const char *iface_prefix, MethodInfo
*method)
{
    char *method_name_uscored, *ret;

    method_name_uscored = _dbus_gutils_wincaps_to_uscore
(method_info_get_name (method));
    ret = g_strdup_printf ("%s_%s", iface_prefix, method_name_uscored);
    g_free (method_name_uscored);

    return ret;
}

static gboolean
write_formal_parameters (InterfaceInfo *iface, MethodInfo *method,
GIOChannel *channel, GError **error)
{
    GSList *args;

    for (args = method_info_get_args (method); args; args = args->next)

```

```

{
    ArgInfo *arg;
    const char *type_str;
    const char *type_suffix;
    GType gtype;
    int direction;

    arg = args->data;

    WRITE_OR_LOSE (" ", " ");

    direction = arg_info_get_direction (arg);

    gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
    if (gtype == G_TYPE_INVALID)
    {
        g_set_error (error,
                    DBUS_BINDING_TOOL_ERROR,
                    DBUS_BINDING_TOOL_ERROR_UNSUPPORTED_CONVERSION,
                    "Unsupported conversion from D-BUS type signature
\"%s\" to glib C type in method \"%s\" of interface \"%s\"",
                    arg_info_get_type (arg),
                    method_info_get_name (method),
                    interface_info_get_name (iface));
        return FALSE;
    }
    type_str = dbus_g_type_get_c_name (gtype);
    g_assert (type_str);
    /* Variants are special...*/
    if (gtype == G_TYPE_VALUE)
    {
        if (direction == ARG_IN)
            type_suffix = "*";
        else
            type_suffix = "";
    }
    else if ((g_type_is_a (gtype, G_TYPE_BOXED)
             || g_type_is_a (gtype, G_TYPE_OBJECT)
             || g_type_is_a (gtype, G_TYPE_POINTER)))
        type_suffix = "*";
    else
        type_suffix = "";

    switch (direction)
    {
    case ARG_IN:
        if (!write_printf_to_iochannel ("const %s%s IN_%s", channel,
error,
                                     type_str,
                                     type_suffix,

```



```

        arg_info_get_name (arg))
    goto io_lose;
break;
case ARG_OUT:
    if (!write_printf_to_iochannel ("%s%s* OUT_%s", channel, error,
        type_str,
        type_suffix,
        arg_info_get_name (arg)))
        goto io_lose;
    break;
case ARG_INVALID:
    break;
}
}

return TRUE;
io_lose:
return FALSE;
}

#define MAP_FUNDAMENTAL(NAME) \
    case G_TYPE_ ## NAME: \
        return g_strdup ("G_TYPE_" #NAME);
#define MAP_KNOWN(NAME) \
    if (gtype == NAME) \
        return g_strdup (#NAME)
static char *
dbus_g_type_get_lookup_function (GType gtype)
{
    char *type_lookup;
    switch (gtype)
    {
        MAP_FUNDAMENTAL (CHAR);
        MAP_FUNDAMENTAL (UCHAR);
        MAP_FUNDAMENTAL (BOOLEAN);
        MAP_FUNDAMENTAL (LONG);
        MAP_FUNDAMENTAL (ULONG);
        MAP_FUNDAMENTAL (INT);
        MAP_FUNDAMENTAL (UINT);
        MAP_FUNDAMENTAL (INT64);
        MAP_FUNDAMENTAL (UINT64);
        MAP_FUNDAMENTAL (FLOAT);
        MAP_FUNDAMENTAL (DOUBLE);
        MAP_FUNDAMENTAL (STRING);
    }
    if (dbus_g_type_is_collection (gtype))
    {
        GType elt_gtype;
        char *sublookup;

        elt_gtype = dbus_g_type_get_collection_specialization (gtype);
        sublookup = dbus_g_type_get_lookup_function (elt_gtype);
    }
}

```

```

g_assert (sublookup);

if (_dbus_g_type_is_fixed (elt_gtype))
  {
    type_lookup = g_strdup_printf ("dbus_g_type_get_collection "
                                   "\"GArray\"", %s)", sublookup);
  }
else
  {
    type_lookup = g_strdup_printf ("dbus_g_type_get_collection "
                                   "\"GPtrArray\"", %s)", sublookup);
  }

g_free (sublookup);

return type_lookup;
}
else if (dbus_g_type_is_map (gtype))
  {
    GType key_gtype;
    char *key_lookup;
    GType value_gtype;
    char *value_lookup;

    key_gtype = dbus_g_type_get_map_key_specialization (gtype);
    value_gtype = dbus_g_type_get_map_value_specialization (gtype);
    key_lookup = dbus_g_type_get_lookup_function (key_gtype);
    g_assert (key_lookup);
    value_lookup = dbus_g_type_get_lookup_function (value_gtype);
    g_assert (value_lookup);
    type_lookup = g_strdup_printf ("dbus_g_type_get_map
(\\"GHashTable\"", %s, %s)",
                                   key_lookup, value_lookup);
    g_free (key_lookup);
    g_free (value_lookup);
    return type_lookup;
  }
else if (dbus_g_type_is_struct (gtype))
  {
    GType value_gtype;
    GString *string;
    char *value_lookup = NULL;
    guint size, i;

    string = g_string_new ("dbus_g_type_get_struct
(\\"GValueArray\"");

    size = dbus_g_type_get_struct_size (gtype);
    for (i=0; i < size; i++)
      {
        value_gtype = dbus_g_type_get_struct_member_type(gtype, i);

```

```

        value_lookup = dbus_g_type_get_lookup_function
(value_gtype);
        g_assert (value_lookup);
        g_string_append_printf (string, ", %s", value_lookup);
        g_free (value_lookup);
    }
    g_string_append (string, ", G_TYPE_INVALID");
    return g_string_free (string, FALSE);
}

MAP_KNOWN(G_TYPE_VALUE);
MAP_KNOWN(G_TYPE_STRV);
MAP_KNOWN(G_TYPE_VALUE_ARRAY);
MAP_KNOWN(DBUS_TYPE_G_PROXY);
MAP_KNOWN(DBUS_TYPE_G_OBJECT_PATH);
MAP_KNOWN(DBUS_TYPE_G_SIGNATURE);
return NULL;
}
#undef MAP_FUNDAMENTAL
#undef MAP_KNOWN

static gboolean
write_args_for_direction (InterfaceInfo *iface, MethodInfo *method,
GIOChannel *channel, int direction, GError **error)
{
    GSList *args;
    char *type_lookup = NULL;

    for (args = method_info_get_args (method); args; args = args->next)
    {
        ArgInfo *arg;
        GType gtype;

        arg = args->data;

        if (direction != arg_info_get_direction (arg))
            continue;

        gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
        g_assert (gtype != G_TYPE_INVALID);
        type_lookup = dbus_g_type_get_lookup_function (gtype);
        g_assert (type_lookup != NULL);

        switch (direction)
        {
        case ARG_IN:
            if (!write_printf_to_iochannel ("%s, IN_%s, ", channel, error,
type_lookup,
arg_info_get_name (arg)))
                goto io_lose;

```

```

        break;
    case ARG_OUT:
        if (!write_printf_to_iochannel ("%s, OUT_%s, ", channel, error,
                                        type_lookup,
                                        arg_info_get_name (arg)))
            goto io_lose;
        break;
    case ARG_INVALID:
        break;
    }
    g_free (type_lookup);
}

return TRUE;
io_lose:
g_free (type_lookup);
return FALSE;
}

static gboolean
check_supported_parameters (MethodInfo *method)
{
    GSList *args;

    for (args = method_info_get_args (method); args; args = args->next)
        {
            ArgInfo *arg;
            GType gtype;

            arg = args->data;
            gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
            if (gtype == G_TYPE_INVALID)
                return FALSE;
        }
    return TRUE;
}

static gboolean
write_untyped_out_args (InterfaceInfo *iface, MethodInfo *method,
GIOChannel *channel, GError **error)
{
    GSList *args;

    for (args = method_info_get_args (method); args; args = args->next)
        {
            ArgInfo *arg;

            arg = args->data;
            if (arg_info_get_direction (arg) != ARG_OUT)
                continue;

```

```

        if (!write_printf_to_iochannel ("OUT_%s, ", channel, error,
                                       arg_info_get_name (arg)))
            goto io_lose;
    }

    return TRUE;
io_lose:
    return FALSE;
}

static gboolean
write_formal_declarations_for_direction (InterfaceInfo *iface,
MethodInfo *method, GIOChannel *channel, const int direction, GError
**error)
{
    GSList *args;

    for (args = method_info_get_args (method); args; args = args->next)
    {
        ArgInfo *arg;
        GType gtype;
        const char *type_str, *type_suffix, *type_initializer = "";
        int dir;

        arg = args->data;

        dir = arg_info_get_direction (arg);

        gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
        type_str = dbus_g_type_get_c_name (gtype);

        if (!type_str)
        {
            g_set_error (error,
                        DBUS_BINDING_TOOL_ERROR,
                        DBUS_BINDING_TOOL_ERROR_UNSUPPORTED_CONVERSION,
                        "Unsupported conversion from D-BUS type
signature \"%s\" to glib C type in method \"%s\" of interface \"%s\"",
                        arg_info_get_type (arg),
                        method_info_get_name (method),
                        interface_info_get_name (iface));
            return FALSE;
        }

        /* Variants are special...*/
        if (gtype == G_TYPE_VALUE)
        {
            if (direction == ARG_IN)
                type_suffix = "*";
            else
            {

```

```

        type_suffix = "";
        type_initializer = " = { 0, }";
    }
}
else if ((g_type_is_a (gtype, G_TYPE_BOXED)
        || g_type_is_a (gtype, G_TYPE_OBJECT)
        || g_type_is_a (gtype, G_TYPE_POINTER)))
type_suffix = "*";
else
type_suffix = "";

if (direction != dir)
    continue;

    switch (dir)
    {
    case ARG_IN:
        if (!write_printf_to_iochannel (" %s%s IN_%s;\n", channel,
error,
                                     type_str, type_suffix,
                                     arg_info_get_name (arg)))
            goto io_lose;
        break;
    case ARG_OUT:
        if (!write_printf_to_iochannel (" %s%s OUT_%s%s;\n",
channel, error,
                                     type_str, type_suffix,
                                     arg_info_get_name (arg),
                                     type_initializer))
            goto io_lose;
        break;
    case ARG_INVALID:
        break;
    }
    }
return TRUE;
io_lose:
return FALSE;
}

static gboolean
write_formal_parameters_for_direction (InterfaceInfo *iface,
MethodInfo *method, int dir, GIOChannel *channel, GError **error)
{
    GSList *args;

    for (args = method_info_get_args (method); args; args = args->next)
    {
        ArgInfo *arg;
        const char *type_str;
        const char *type_suffix;
        GType gtype;

```

```

int direction;

arg = args->data;

direction = arg_info_get_direction (arg);
if (dir != direction) continue;

WRITE_OR_LOSE (" ", ");

gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
type_str = dbus_g_type_get_c_name (gtype);
/* Variants are special...*/
if (gtype == G_TYPE_VALUE)
{
    if (direction == ARG_IN)
        type_suffix = "*";
    else
        type_suffix = "";
}
else if ((g_type_is_a (gtype, G_TYPE_BOXED)
        || g_type_is_a (gtype, G_TYPE_OBJECT)
        || g_type_is_a (gtype, G_TYPE_POINTER)))
    type_suffix = "*";
else
    type_suffix = "";

if (!type_str)
{
    g_set_error (error,
                DBUS_BINDING_TOOL_ERROR,
                DBUS_BINDING_TOOL_ERROR_UNSUPPORTED_CONVERSION,
                "Unsupported conversion from D-BUS type signature
\"%s\" to glib C type in method \"%s\" of interface \"%s\"",
                arg_info_get_type (arg),
                method_info_get_name (method),
                interface_info_get_name (iface));
    return FALSE;
}

switch (direction)
{
case ARG_IN:
    if (!write_printf_to_iochannel ("const %s%s IN_%s", channel,
error,
                                type_str,
                                type_suffix,
                                arg_info_get_name (arg)))
        goto io_lose;
    break;
case ARG_OUT:
    if (!write_printf_to_iochannel ("%s%s* OUT_%s", channel, error,

```

```

        type_str,
        type_suffix,
        arg_info_get_name (arg))
        goto io_lose;
    break;
case ARG_INVALID:
    break;
}
}
return TRUE;
io_lose:
return FALSE;
}

static gboolean
write_typed_args_for_direction (InterfaceInfo *iface, MethodInfo
*method, GIOChannel *channel, const int direction, GError **error)
{
    GSList *args;
    char *type_lookup = NULL;

    for (args = method_info_get_args (method); args; args = args->next)
    {
        ArgInfo *arg;
        int dir;
        GType gtype;

        arg = args->data;

        dir = arg_info_get_direction (arg);

        if (dir != direction)
            continue;

        gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
        type_lookup = dbus_g_type_get_lookup_function (gtype);

        if (!write_printf_to_iochannel ("%s, &%s %s, ", channel, error,
type_lookup, direction == ARG_IN ? "IN" : "OUT", arg_info_get_name
(arg)))
            goto io_lose;
        g_free (type_lookup);
    }
    return TRUE;
io_lose:
    g_free (type_lookup);
    return FALSE;
}

static gboolean

```



```

write_async_method_client (GIOChannel *channel, InterfaceInfo
*interface, MethodInfo *method, GError **error)
{
    char *method_name, *iface_prefix;
    const char *interface_c_name;

    iface_prefix = iface_to_c_prefix (interface_info_get_name
(interface));
    interface_c_name = interface_info_get_annotation (interface,
DBUS_GLIB_ANNOTATION_CLIENT_C_SYMBOL);
    if (interface_c_name == NULL)
    {
        interface_c_name = (const char *) iface_prefix;
    }

    method_name = g_strdup (method_info_get_annotation (method,
DBUS_GLIB_ANNOTATION_CLIENT_C_SYMBOL));
    if (method_name == NULL)
    {
        method_name = compute_client_method_name (interface_c_name,
method);
    }
    g_free(iface_prefix);

    /* Write the typedef for the client callback */
    if (!write_printf_to_iochannel ("typedef void (*%s_reply)
(DBusGProxy *proxy, ", channel, error, method_name))
        goto io_lose;
    {
        GSList *args;
        for (args = method_info_get_args (method); args; args = args-
>next)
        {
            ArgInfo *arg;
            const char *type_suffix, *type_str;
            GType gtype;

            arg = args->data;

            if (arg_info_get_direction (arg) != ARG_OUT)
                continue;
            gtype = _dbus_gtype_from_signature (arg_info_get_type (arg),
TRUE);
            if (gtype != G_TYPE_VALUE && (g_type_is_a (gtype, G_TYPE_BOXED)
|| g_type_is_a (gtype, G_TYPE_OBJECT)
|| g_type_is_a (gtype, G_TYPE_POINTER)))
                type_suffix = "*";
            else
                type_suffix = "";
            type_str = dbus_g_type_get_c_name (_dbus_gtype_from_signature
(arg_info_get_type (arg), TRUE));

```

```

        if (!write_printf_to_iochannel ("%s %sOUT_%s, ", channel, error,
type_str, type_suffix, arg_info_get_name (arg)))
            goto io_lose;
    }
}
WRITE_OR_LOSE ("GError *error, gpointer userdata);\n\n");

/* Write the callback when the call returns */
WRITE_OR_LOSE ("static void\n");
if (!write_printf_to_iochannel ("%s_async_callback (DBusGProxy
*proxy, DBusGProxyCall *call, void *user_data)\n", channel, error,
method_name))
    goto io_lose;
WRITE_OR_LOSE ("{\n");
WRITE_OR_LOSE ("    DBusGAsyncData *data = (DBusGAsyncData*)
user_data;\n    GError *error = NULL;\n");
if (!write_formal_declarations_for_direction (interface, method,
channel, ARG_OUT, error))
    goto io_lose;
/* TODO: handle return boolean of end_call */
WRITE_OR_LOSE ("    dbus_g_proxy_end_call (proxy, call, &error, ");
if (!write_typed_args_for_direction (interface, method, channel,
ARG_OUT, error))
    goto io_lose;
WRITE_OR_LOSE ("G_TYPE_INVALID);\n");
if (!write_printf_to_iochannel ("    (*(%s_reply)data->cb) (proxy, ",
channel, error, method_name))
    goto io_lose;
if (!write_untyped_out_args (interface, method, channel, error))
    goto io_lose;
WRITE_OR_LOSE ("error, data->userdata);\n");
WRITE_OR_LOSE ("    return;\n}\n\n");

/* Write the main wrapper function */
WRITE_OR_LOSE ("static\n#ifdef
G_HAVE_INLINE\ninline\n#endif\nDBusGProxyCall*\n");
if (!write_printf_to_iochannel ("%s_async (DBusGProxy *proxy",
channel, error,
                                method_name))
    goto io_lose;
if (!write_formal_parameters_for_direction (interface, method,
ARG_IN, channel, error))
    goto io_lose;

if (!write_printf_to_iochannel ("    , %s_reply callback, gpointer
userdata)\n\n", channel, error, method_name))
    goto io_lose;

WRITE_OR_LOSE ("{\n");

```

```

WRITE_OR_LOSE (" DbusGAsyncData *stuff;\n stuff = g_slice_new
(DBusGAsyncData);\n stuff->cb = G_CALLBACK (callback);\n stuff-
>userdata = userdata;\n");
if (!write_printf_to_iochannel (" return dbus_g_proxy_begin_call
(proxy, \"%s\", %s_async_callback, stuff, _dbus_glib_async_data_free,
", channel, error, method_info_get_name (method), method_name))
goto io_lose;
if (!write_args_for_direction (interface, method, channel, ARG_IN,
error))
goto io_lose;
WRITE_OR_LOSE ("G_TYPE_INVALID);\n}\n");

g_free (method_name);
return TRUE;
io_lose:
g_free (method_name);
return FALSE;
}

```

```

static gboolean
generate_client_glue_list (GSList *list, DbusBindingToolCData *data,
GError **error)
{
GSList *tmp;

tmp = list;
while (tmp != NULL)
{
if (!generate_client_glue (tmp->data, data, error))
return FALSE;
tmp = tmp->next;
}
return TRUE;
}

```

```

static gboolean
generate_client_glue (BaseInfo *base, DbusBindingToolCData *data,
GError **error)
{
char *iface_prefix;
char *method_c_name;
iface_prefix = NULL;
method_c_name = NULL;

if (base_info_get_type (base) == INFO_TYPE_NODE)
{
if (!generate_client_glue_list (node_info_get_nodes ((NodeInfo
*) base),
data, error))
return FALSE;
if (!generate_client_glue_list (node_info_get_interfaces
((NodeInfo *) base),

```

```

        data, error))
    return FALSE;
}
else
{
    GIOChannel *channel;
    InterfaceInfo *interface;
    GSList *methods;
    GSList *tmp;
    const char *interface_c_name;

    channel = data->channel;

    interface = (InterfaceInfo *) base;

    methods = interface_info_get_methods (interface);

    iface_prefix = iface_to_c_prefix (interface_info_get_name
(interface));
    interface_c_name = interface_info_get_annotation (interface,
DBUS_GLIB_ANNOTATION_CLIENT_C_SYMBOL);
    if (interface_c_name == NULL)
    {
        interface_c_name = (const char *) iface_prefix;
    }

    if (!write_printf_to_iochannel ("#ifndef
DBUS_GLIB_CLIENT_WRAPPERS_%s\n"
        "#define DBUS_GLIB_CLIENT_WRAPPERS_%s\n\n",
        channel, error,
        iface_prefix, iface_prefix))
        goto io_lose;

    for (tmp = methods; tmp != NULL; tmp = g_slist_next (tmp))
    {
        MethodInfo *method;
        gboolean is_noreply;

        method = (MethodInfo *) tmp->data;
        method_c_name = g_strdup (method_info_get_annotation
(method, DBUS_GLIB_ANNOTATION_CLIENT_C_SYMBOL));
        if (method_c_name == NULL)
        {
            method_c_name = compute_client_method_name
(interface_c_name, method);
        }

        is_noreply = method_info_get_annotation (method,
DBUS_GLIB_ANNOTATION_NOREPLY) != NULL;

        if (data->ignore_unsupported && !check_supported_parameters
(method))

```

```

        {
            g_warning ("Ignoring unsupported signature in method
\"%s\" of interface \"%s\"\\n",
                method_info_get_name (method),
                interface_info_get_name (interface));
            g_free (method_c_name);
            method_c_name = NULL;
            continue;
        }

        WRITE_OR_LOSE ("static\\n#ifdef
G_HAVE_INLINE\\ninline\\n#endif\\ngboolean\\n");
        if (!write_printf_to_iochannel ("%s (DBusGProxy *proxy",
channel, error,
                method_c_name))
            goto io_lose;
        g_free (method_c_name);
        method_c_name = NULL;

        if (!write_formal_parameters (interface, method, channel,
error))
            goto io_lose;

        WRITE_OR_LOSE ("", GError **error)\\n\\n");

        WRITE_OR_LOSE ("{\\n");

        if (is_noreply) {
            if (!write_printf_to_iochannel ("
dbus_g_proxy_call_no_reply (proxy, \"%s\", \"\", channel, error,
                method_info_get_name (method)))
                goto io_lose;

            if (!write_args_for_direction (interface, method, channel,
ARG_IN, error))
                goto io_lose;

            WRITE_OR_LOSE ("G_TYPE_INVALID, ");

            if (!write_args_for_direction (interface, method, channel,
ARG_OUT, error))
                goto io_lose;

            WRITE_OR_LOSE ("G_TYPE_INVALID);\\n");

            WRITE_OR_LOSE (" return TRUE;\\n}\\n\\n");
        } else {
            if (!write_printf_to_iochannel (" return
dbus_g_proxy_call (proxy, \"%s\", \"\", channel, error,
                method_info_get_name (method)))
                goto io_lose;

```

```

        WRITE_OR_LOSE ("error, ");

        if (!write_args_for_direction (interface, method, channel,
ARG_IN, error))
            goto io_lose;

        WRITE_OR_LOSE ("G_TYPE_INVALID, ");

        if (!write_args_for_direction (interface, method, channel,
ARG_OUT, error))
            goto io_lose;

        WRITE_OR_LOSE ("G_TYPE_INVALID);\n}\n\n");
    }

    write_async_method_client (channel, interface, method,
error);
}

    if (!write_printf_to_iochannel ("#endif /* defined
DBUS_GLIB_CLIENT_WRAPPERS_%s */\n\n", channel, error, iface_prefix))
        goto io_lose;

    g_free (iface_prefix);
}
return TRUE;
io_lose:
    g_free (method_c_name);
    g_free (iface_prefix);
    return FALSE;
}

gboolean
dbus_binding_tool_output_glib_client (BaseInfo *info, GIOChannel
*channel, gboolean ignore_unsupported, GError **error)
{
    DbusBindingToolCData data;
    gboolean ret;

    memset (&data, 0, sizeof (data));

    data.channel = channel;
    data.ignore_unsupported = ignore_unsupported;

    dbus_g_type_specialized_init ();
    _dbus_g_type_specialized_builtins_init ();

    WRITE_OR_LOSE ("/* Generated by dbus-binding-tool; do not edit!
*/\n\n");
    WRITE_OR_LOSE ("#include <glib.h>\n");

```

```

WRITE_OR_LOSE ("#include <dbus/dbus-glib.h>\n\n");
WRITE_OR_LOSE ("G_BEGIN_DECLS\n\n");

WRITE_OR_LOSE ("#ifndef _DBUS_GLIB_ASYNC_DATA_FREE\n");
WRITE_OR_LOSE ("#define _DBUS_GLIB_ASYNC_DATA_FREE\n");
WRITE_OR_LOSE ("static\n#ifdef
G_HAVE_INLINE\ninline\n#endif\nvoid\n");
WRITE_OR_LOSE ("_dbus_glib_async_data_free (gpointer
stuff)\n{\n\tg_slice_free (DBusGAsyncData, stuff);\n}\n");
WRITE_OR_LOSE ("#endif\n\n");

ret = generate_client_glue (info, &data, error);
if (!ret)
    goto io_lose;

WRITE_OR_LOSE ("G_END_DECLS\n");

return ret;
io_lose:
return FALSE;
}

```

File = dbus-binding-tool-glib.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-binding-tool-output-glib.h prototypes for glib output
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#endif DBUS_BINDING_TOOL_OUTPUT_GLIB_H

```

```

#define DBUS_BINDING_TOOL_OUTPUT_GLIB_H

G_BEGIN_DECLS

#define DBUS_GLIB_ANNOTATION_C_SYMBOL
"org.freedesktop.DBus.GLib.CSymbol"
#define DBUS_GLIB_ANNOTATION_CLIENT_C_SYMBOL
"org.freedesktop.DBus.GLib.ClientCSymbol"
#define DBUS_GLIB_ANNOTATION_ASYNC "org.freedesktop.DBus.GLib.Async"
#define DBUS_GLIB_ANNOTATION_CONST "org.freedesktop.DBus.GLib.Const"
#define DBUS_GLIB_ANNOTATION_RETURNVAL
"org.freedesktop.DBus.GLib.ReturnVal"
#define DBUS_GLIB_ANNOTATION_NOREPLY
"org.freedesktop.DBus.Method.NoReply"

gboolean dbus_binding_tool_output_glib_client (BaseInfo *info,
GIOChannel *channel, gboolean ignore_unsupported, GError **error);
gboolean dbus_binding_tool_output_glib_server (BaseInfo *info,
GIOChannel *channel, const char *prefix, GError **error);

G_END_DECLS

#endif

File = dbus-binding-tool.1

\" te
.TH dbus-binding-tool 1 "26 Feb 2009" "SunOS 5.11" "User Commands"
.SH "NAME"
dbus-binding-tool \- C language GLib bindings generation utility\&.
.SH "SYNOPSIS"
.PP
\fbdbus-binding-tool\fr [-\fb-force\fr] [-\fb-help\fr] [-\fb-ignore-
unsupported\fr] [-\fb-mode=\fIpretty|glib-client|glib-server\fr\fr] [-
\fb-output=\fIfile\fr\fr] [-\fb-prefix=\fIsymbol-prefix\fr\fr] [-\fb-
version\fr] [\fb\fiifile\fr\fr\&...]
.SH "DESCRIPTION"
.PP
\fbdbus-binding-tool\fr is used to expose a GObject via D\-\Bus\&. As
input,
\fbdbus-binding-tool\fr uses a D\-\Bus Introspection XML file\&. As
output,
the client-side or server-side bindings is generated\&. This output
is a header
file which eases the use of a remote D\-\Bus object\&. Output is sent
to standard
out or to the filename specified with the -\fb-output\fr argument\&.
.SH "EXTENDED DESCRIPTION"
.SS ""
.PP

```


The following is a sample D\Bus Introspection XML file which describes an

object that exposes one method, named ManyArgs:

.PP

.nf

```
<?xml version="1.0" encoding="UTF-8" ?>
<node name="/com/example/MyObject">
  <interface name="com\example\MyObject">
    <method name="ManyArgs">
      <arg type="u" name="x" direction="in" />
      <arg type="s" name="str" direction="in" />
      <arg type="d" name="trouble" direction="in" />
      <arg type="d" name="d_ret" direction="out" />
      <arg type="s" name="str_ret" direction="out" />
    </method>
  </interface>
</node>
```

.fi

.PP

\fBdbus-binding-tool\fR supports annotations in the XML format to further

control how the bindings are generated\&.

.SS "client-side bindings"

.PP

When building client-side bindings, the `-fb-mode=glib-client` argument is used\&. The client-side bindings support the `"org\freedesktop\DBus\Glib\NoReply"` annotation\&. This is specified

within the `<method>` tag to indicate that the client is not expecting a reply to the method call, so a reply should not be sent\&. This is often used to speed up rapid method calls where there are no "out" arguments, and not knowing if the method succeeded is an acceptable compromise

to halve the traffic on the bus\&. For example:

.PP

.nf

```
<method name "FooMethod">
  [\&.\&.\&.]
  <annotation name="org\freedesktop\DBus\GLib\NoReply"
value="yes"/>
  [\&.\&.\&.]
</method>
```

.fi

.SS "server-side bindings"

.PP

When building server-side bindings, the `-fb-mode=glib-server` argument is used\&. Also the `-fb-prefix` argument must be used when

building server-side bindings so that functions in the generated output are

prefixed with the specified value\&. The server-side bindings support the

following annotations:

```
.PP
"org\&.freedesktop\&.DBus\&.GLib\&.CSymbol"
.PP
```

This annotation is used to specify the C symbol names for the various types (interface, method, etc\&.), if it differs from the name D\&-Bus generates\&.

```
.PP
.PP
.nf
<interface name="com\&.example\&.MyObject">
  <annotation name="org\&.freedesktop\&.DBus\&.GLib\&.CSymbol"
value="my_object"/>
  [\&.\&.\&.]
  <method name "ManyArgs">
    <annotation name="org\&.freedesktop\&.DBus\&.GLib\&.CSymbol"
value="my_object_many_args"/>
    [\&.\&.\&.]
  </method>
</interface>
```

```
.fi
.PP
"org\&.freedesktop\&.DBus\&.GLib\&.Async"
.PP
```

This annotation marks the method implementation as an asynchronous function, which does not return a response straight away but will send the response at some later point to complete the call\&. This is used to implement non-blocking services where method calls can take time\&.

```
.PP
When a method is asynchronous, the function prototype is different\&. It is required that the function conform to the following rules:
```

```
.sp
.in +2
\ (bu
.mk
.in +3
.rt
The function must return a value of type gboolean; TRUE on success, and FALSE otherwise\&.
```

```
.in -3
\ (bu
.mk
.in +3
.rt
The first parameter is a pointer to an instance of the object\&.
```

```
.in -3
\ (bu
```

```

.mk
.in +3
.rt
Following the object instance pointer are the method input values\&.
.in -3
\ (bu
.mk
.in +3
.rt
The final parameter must be a (DBusGMethodInvocation *)\&. This is
used
when sending the response message back to the client, by calling
dbus_g_method_return or dbus_g_method_return_error\&.
.in -3
.in -2
.PP
For example:
.PP
.nf
<method name "FooMethod">
  [\&.\&.\&.]
  <annotation name="org\&.freedesktop\&.DBus\&.GLib\&.Async"
value="yes"/>
  [\&.\&.\&.]
</method>
.fi
.PP
"org\&.freedesktop\&.DBus\&.GLib\&.Const"
.PP
This attribute can only be applied to "out" <arg> nodes, and
specifies that the parameter is not being copied when returned\&. For
example,
this turns a \&'s\&' argument from a (char **) to a (const char **),
and results in
the argument not being freed by D\&-Bus after the message is sent\&.
For example:
.PP
.nf
  <arg type="u" name="x" direction="out">
    <annotation name="org\&.freedesktop\&.DBus\&.GLib\&.Const"
value=""/>
  </arg>
.fi
.PP
"org\&.freedesktop\&.DBus\&.GLib\&.RetVal"
.PP
This attribute can only be applied to "out" <arg> nodes, and
alters the expected function signature\&. It currently can be set to
two values:
"" or "error"\&. The argument marked with this attribute is
not returned via a pointer argument, but by the function\&'s return
value\&. If the

```

attribute\&'s value is the empty string, the (GError *) argument is also omitted

so there is no standard way to return an error value\&. This is very useful for

interfacing with existing code, as it is possible to match existing APIs\&. If

the attribute\&'s value is "error", then the final argument is a (GError *) as usual\&. For example:

```
.PP
.nf
  <arg type="u" name="x" direction="out">
    <annotation name="org\&.freedesktop\&.DBus\&.GLib\&.RetVal"
value=""/>
  </arg>
```

```
.fi
.SH "OPTIONS"
```

```
.PP
The following options are supported:
```

```
.sp
.ne 2
.mk
\FB-\fB-force\fR\fR
.sp .6
.in +4
```

Overwrite the output file if it already exists with a newer timestamp than the source files\&.

```
.sp
.sp 1
.in -4
.sp
.ne 2
.mk
\FB-\fB-help\fR\fR
.sp .6
.in +4
```

Display usage information\&.

```
.sp
.sp 1
.in -4
.sp
.ne 2
.mk
\FB-\fB-ignore-unsupported\fR\fR
.sp .6
.in +4
```

If set, then unsupported signatures for <method> parameters are ignored\&.

```
.sp
.sp 1
.in -4
.sp
```

```

.ne 2
.mk
\fb-\fb-mode=\fIpretty|glib-client|glib-server\fR\fR\fR
.sp .6
.in +4
If the value is "glib-client", then client bindings are generated\&.
If the value is "glib-server", then server bindings are generated\&.
If the value is "pretty", then the output is in a more human readable
format\&.
.sp
.sp 1
.in -4
.sp
.ne 2
.mk
\fb-\fb-output=\fIfile\fR\fR\fR
.sp .6
.in +4
Specify the output \fIfile\fR\&.
.sp
.sp 1
.in -4
.sp
.ne 2
.mk
\fb-\fb-prefix=\fIsymbol-prefix\fR\fR\fR
.sp .6
.in +4
Functions in the generated output are prefixed with the
\fIsymbol-prefix\fR value\&.
.sp
.sp 1
.in -4
.sp
.ne 2
.mk
\fb-\fb-version\fR\fR
.sp .6
.in +4
Display the version number of the \fbdbus-binding-tool\fR command\&.
.sp
.sp 1
.in -4
.SH "OPERANDS"
.PP
The following operands are supported:
.sp
.ne 2
.mk
\fb\fb\fIfile\fR\fR\fR
.in +16n
.rt

```

A list of one or more input D\ -Bus Introspection XML files to include in the generated output\&.

```
.sp
.sp 1
.in -16n
.SH "FILES"
.PP
```

The following files are used by this application:

```
.sp
.ne 2
.mk
\fb\fb/usr/bin/dbus-binding-tool\fr \fr
.in +32n
.rtf
```

Executable for the D\ -Bus Binding Tool application\&.

```
.sp
.sp 1
.in -32n
.SH "ATTRIBUTES"
.PP
```

See \fbattributes\fr(5)

for descriptions of the following attributes:

```
.sp
.TS
tab( ) allbox;
cw(2.750000i) | cw(2.750000i)
lw(2.750000i) | lw(2.750000i).
ATTRIBUTE TYPE ATTRIBUTE VALUE
Availability SUNWdbus-bindings
Interface stability Volatile
```

```
.TE
.sp
.SH "SEE ALSO"
```

```
.PP
\fbdbus-cleanup-sockets\fr(1),
\fbdbus-daemon\fr(1),
\fbdbus-monitor\fr(1),
\fbdbus-send\fr(1),
\fbdbus-uuidgen\fr(1),
\fbldbdbus-glib-1\fr(3),
\fbattributes\fr(5)
```

```
.SH "NOTES"
.PP
```

Written by Brian Cameron, Sun Microsystems Inc\&., 2009\&.

...\ " created by instant / solbook-to-man, Thu 26 Feb 2009, 19:15

...\ " LSARC 2006/368 D-BUS Message Bus System

File = dbus-binding-tool.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>dbus-binding-tool</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch03.html" title="Tools Reference">
<link rel="prev" href="ch03.html" title="Tools Reference">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2"><tr valign="middle">
<td><a accesskey="p" href="ch03.html"></a></td>
<td><a accesskey="u" href="ch03.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td>Â </td>
</tr></table>
<div class="refentry">
<a name="dbus-binding-tool"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle">dbus-binding-tool</span></h2>
<p>dbus-binding-tool â€” C language GLib bindings generation
utility</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsynopsisdiv">
<h2>Synopsis</h2>
<div class="cmdsynopsis"><p>[options...] [files...]</p></div>
</div>
<div class="refsect1">
<a name="idp3759776"></a><h2>Description</h2>
<p><span class="command"><strong>dbus-binding-tool</strong></span> bla
bla bla
</p>
</div>
<div class="refsect1">
<a name="idp4771024"></a><h2>Invocation</h2>
<p><span class="command"><strong>dbus-binding-tool</strong></span> bla
bla bla
</p>

```

```

<div class="refsect2">
<a name="idp4772416"></a><h3>Options</h3>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><code class="option">--mode</code><em
class="replaceable"><code>pretty|glib-server|glib-
client</code></em></span></p></td>
<td><p>
bla bla
</p></td>
</tr>
<tr>
<td><p><span class="term"><code class="option">--
help</code></span></p></td>
<td><p>
Print brief help and exit.
</p></td>
</tr>
<tr>
<td><p><span class="term"><code class="option">--
version</code></span></p></td>
<td><p>
Print version and exit.
</p></td>
</tr>
</tbody>
</table></div>
</div>
</div>
<div class="refsect1">
<a name="idp4778048"></a><h2>See also</h2>
<p><span class="command"><strong>glib-genmarshal</strong></span> (1)
</p>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-binding-tool.xml

```

<refentry id="dbus-binding-tool">
<refmeta>
<refentrytitle>dbus-binding-tool</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>

```



```
<refnamediv>
<refname>dbus-binding-tool</refname>
<refpurpose>C language GLib bindings generation utility</refpurpose>
</refnamediv>
```

```
<refsynopsisdiv>
<cmdsynopsis>
<arg choice="opt" rep="repeat">options</arg>
<arg choice="opt" rep="repeat">files</arg>
</cmdsynopsis>
</refsynopsisdiv>
```

```
<refsect1><title>Description</title>
<para><command>dbus-binding-tool</command> bla bla bla
</para>
</refsect1>
```

```
<refsect1><title>Invocation</title>
<para><command>dbus-binding-tool</command> bla bla bla
</para>
```

```
<refsect2><title>Options</title>
<variablelist>
```

```
<varlistentry>
<term><option>--mode</option><replaceable>pretty|glib-server|glib-
client</replaceable></term>
<listitem><para>
bla bla
</para></listitem>
</varlistentry>
```

```
<varlistentry>
<term><option>--help</option></term>
<listitem><para>
Print brief help and exit.
</para></listitem>
</varlistentry>
```

```
<varlistentry>
<term><option>--version</option></term>
<listitem><para>
Print version and exit.
</para></listitem>
</varlistentry>
```

```
</variablelist>
</refsect2>
```

```
</refsect1>
```

```
<refsect1><title>See also</title>
<para><command>glib-genmarshal</command>(1)
</para>
</refsect1>
</refentry>
```

File = dbus-bus-introspect.xml

```
<!DOCTYPE node PUBLIC "-//freedesktop//DTD D-BUS Object
Introspection 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/introspect.dtd">
<node>
  <interface name="org.freedesktop.DBus.Introspectable">
    <method name="Introspect">
      <arg name="data" direction="out" type="s"/>
    </method>
  </interface>
  <interface name="org.freedesktop.DBus">
    <method name="RequestName">
      <arg direction="in" type="s"/>
      <arg direction="in" type="u"/>
      <arg direction="out" type="u"/>
    </method>
    <method name="ReleaseName">
      <arg direction="in" type="s"/>
      <arg direction="out" type="u"/>
    </method>
    <method name="StartServiceByName">
      <arg direction="in" type="s"/>
      <arg direction="in" type="u"/>
      <arg direction="out" type="u"/>
    </method>
    <method name="Hello">
      <arg direction="out" type="s"/>
    </method>
    <method name="NameHasOwner">
      <arg direction="in" type="s"/>
      <arg direction="out" type="b"/>
    </method>
    <method name="ListNames">
      <arg direction="out" type="as"/>
    </method>
    <method name="ListActivatableNames">
      <arg direction="out" type="as"/>
    </method>
    <method name="AddMatch">
      <arg direction="in" type="s"/>
    </method>
    <method name="RemoveMatch">
      <arg direction="in" type="s"/>
    </method>
```

```

</method>
<method name="GetNameOwner">
  <arg direction="in" type="s"/>
  <arg direction="out" type="s"/>
</method>
<method name="ListQueuedOwners">
  <arg direction="in" type="s"/>
  <arg direction="out" type="as"/>
</method>
<method name="GetConnectionUnixUser">
  <arg direction="in" type="s"/>
  <arg direction="out" type="u"/>
</method>
<method name="GetConnectionUnixProcessID">
  <arg direction="in" type="s"/>
  <arg direction="out" type="u"/>
</method>
<method name="GetConnectionSELinuxSecurityContext">
  <arg direction="in" type="s"/>
  <arg direction="out" type="ay"/>
</method>
<method name="ReloadConfig">
</method>
<signal name="NameOwnerChanged">
  <arg type="s"/>
  <arg type="s"/>
  <arg type="s"/>
</signal>
<signal name="NameLost">
  <arg type="s"/>
</signal>
<signal name="NameAcquired">
  <arg type="s"/>
</signal>
</interface>
</node>

```

File = dbus-bus.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-bus.c Convenience functions for communicating with the bus.
 *
 * Copyright (C) 2003 CodeFactory AB
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify

```

* it under the terms of the GNU General Public License as published
by

* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.

*

* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

*

* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA

02110-1301 USA

*

*/

```
#include <config.h>
#include "dbus-bus.h"
#include "dbus-protocol.h"
#include "dbus-internals.h"
#include "dbus-message.h"
#include "dbus-marshall-validate.h"
#include "dbus-threads-internal.h"
#include "dbus-connection-internal.h"
#include "dbus-string.h"
```

/**

* @defgroup DBusBus Message bus APIs

* @ingroup DBus

* @brief Functions for communicating with the message bus

*

* dbus_bus_get() allows all modules and libraries in a given
* process to share the same connection to the bus daemon by storing
* the connection globally.

*

* All other functions in this module are just convenience functions;
* most of them invoke methods on the bus daemon, by sending method
* call messages to #DBUS_SERVICE_DBUS. These convenience functions
* often make blocking method calls. If you don't want to block,
* you can send the method call messages manually in the same way
* you would any other method call message.

*

* This module is the only one in libdbus that's specific to
* communicating with the message bus daemon. The rest of the API can
* also be used for connecting to another application directly.

*

* @todo right now the default address of the system bus is hardcoded,
* so if you change it in the global config file suddenly you have to
* set DBUS_SYSTEM_BUS_ADDRESS env variable. Might be nice if the
* client lib somehow read the config file, or if the bus on startup
* somehow wrote out its address to a well-known spot, but might also

```

* not be worth it.
*/

/**
 * @defgroup DBusBusInternals Message bus APIs internals
 * @ingroup DBusInternals
 * @brief Internals of functions for communicating with the message
bus
 *
 * @{
 */

/**
 * Block of message-bus-related data we attach to each
 * #DBusConnection used with these convenience functions.
 *
 */
typedef struct
{
    DBusConnection *connection; /**< Connection we're associated with */
    char *unique_name; /**< Unique name of this connection */

    unsigned int is_well_known : 1; /**< Is one of the well-known
connections in our global array */
} BusData;

/** The slot we have reserved to store BusData.
 */
static dbus_int32_t bus_data_slot = -1;

/** Number of bus types */
#define N_BUS_TYPES 3

static DBusConnection *bus_connections[N_BUS_TYPES];
static char *bus_connection_addresses[N_BUS_TYPES] = { NULL, NULL,
NULL };

static DBusBusType activation_bus_type = DBUS_BUS_STARTER;

static dbus_bool_t initialized = FALSE;

/**
 * Lock for globals in this file
 */
_DBUS_DEFINE_GLOBAL_LOCK (bus);

/**
 * Global lock covering all BusData on any connection. The bet is
 * that some lock contention is better than more memory
 * for a per-connection lock, but it's tough to imagine it mattering
 * either way.
 */

```

```

_DBUS_DEFINE_GLOBAL_LOCK (bus_datas);

static void
addresses_shutdown_func (void *data)
{
    int i;

    i = 0;
    while (i < N_BUS_TYPES)
    {
        if (bus_connections[i] != NULL)
            _dbus_warn_check_failed ("dbus_shutdown() called but
connections were still live. This probably means the application did
not drop all its references to bus connections.\n");

        dbus_free (bus_connection_addresses[i]);
        bus_connection_addresses[i] = NULL;
        ++i;
    }

    activation_bus_type = DBUS_BUS_STARTER;

    initialized = FALSE;
}

static dbus_bool_t
get_from_env (char          **connection_p,
              const char    *env_var)
{
    const char *s;

    _dbus_assert (*connection_p == NULL);

    s = _dbus_getenv (env_var);
    if (s == NULL || *s == '\0')
        return TRUE; /* successfully didn't use the env var */
    else
    {
        *connection_p = _dbus_strdup (s);
        return *connection_p != NULL;
    }
}

static dbus_bool_t
init_session_address (void)
{
    dbus_bool_t retval;

    retval = FALSE;

    /* First, look in the environment. This is the normal case on
    * freedesktop.org/Unix systems. */

```

```

get_from_env (&bus_connection_addresses[DBUS_BUS_SESSION],
             "DBUS_SESSION_BUS_ADDRESS");
if (bus_connection_addresses[DBUS_BUS_SESSION] == NULL)
{
    dbus_bool_t supported;
    DBusString addr;
    DBusError error = DBUS_ERROR_INIT;

    if (!_dbus_string_init (&addr))
        return FALSE;

    supported = FALSE;
    /* So it's not in the environment - let's try a platform-
specific method.
    * On MacOS, this involves asking launchd. On Windows (not
specified yet)
    * we might do a COM lookup.
    * Ignore errors - if we failed, fall back to autolaunch. */
    retval = _dbus_lookup_session_address (&supported, &addr,
&error);
    if (supported && retval)
    {
        retval = _dbus_string_steal_data (&addr,
&bus_connection_addresses[DBUS_BUS_SESSION]);
    }
    else if (supported && !retval)
    {
        if (dbus_error_is_set(&error))
            _dbus_warn ("Dynamic session lookup supported but failed:
%s\n", error.message);
        else
            _dbus_warn ("Dynamic session lookup supported but failed
silently\n");
    }
    _dbus_string_free (&addr);
}
else
    retval = TRUE;

if (!retval)
    return FALSE;

/* The DBUS_SESSION_BUS_DEFAULT_ADDRESS should have really been
named
* DBUS_SESSION_BUS_FALLBACK_ADDRESS.
*/
if (bus_connection_addresses[DBUS_BUS_SESSION] == NULL)
    bus_connection_addresses[DBUS_BUS_SESSION] =
        _dbus_strdup (DBUS_SESSION_BUS_DEFAULT_ADDRESS);
if (bus_connection_addresses[DBUS_BUS_SESSION] == NULL)
    return FALSE;

```

```

    return TRUE;
}

static dbus_bool_t
init_connections_unlocked (void)
{
    if (!initialized)
    {
        const char *s;
        int i;

        i = 0;
        while (i < N_BUS_TYPES)
        {
            bus_connections[i] = NULL;
            ++i;
        }

        /* Don't init these twice, we may run this code twice if
         * init_connections_unlocked() fails midway through.
         * In practice, each block below should contain only one
         * "return FALSE" or running through twice may not
         * work right.
         */

        if (bus_connection_addresses[DBUS_BUS_SYSTEM] == NULL)
        {
            _dbus_verbose ("Filling in system bus address...\n");

            if (!get_from_env
                (&bus_connection_addresses[DBUS_BUS_SYSTEM],
                 "DBUS_SYSTEM_BUS_ADDRESS"))
                return FALSE;
        }

        if (bus_connection_addresses[DBUS_BUS_SYSTEM] == NULL)
        {
            /* Use default system bus address if none set in
             environment */
            bus_connection_addresses[DBUS_BUS_SYSTEM] =
                _dbus_strdup (DBUS_SYSTEM_BUS_DEFAULT_ADDRESS);

            if (bus_connection_addresses[DBUS_BUS_SYSTEM] == NULL)
                return FALSE;

            _dbus_verbose ("  used default system bus \"%s\"\n",
                bus_connection_addresses[DBUS_BUS_SYSTEM]);
        }
        else
            _dbus_verbose ("  used env var system bus \"%s\"\n",
                bus_connection_addresses[DBUS_BUS_SYSTEM]);
    }
}

```



```

if (bus_connection_addresses[DBUS_BUS_SESSION] == NULL)
{
    _dbus_verbose ("Filling in session bus address...\n");

    if (!init_session_address ())
        return FALSE;

    _dbus_verbose (" \\\n",
bus_connection_addresses[DBUS_BUS_SESSION] ?
        bus_connection_addresses[DBUS_BUS_SESSION] :
"none set");
}

if (bus_connection_addresses[DBUS_BUS_STARTER] == NULL)
{
    _dbus_verbose ("Filling in activation bus address...\n");

    if (!get_from_env
(&bus_connection_addresses[DBUS_BUS_STARTER],
        "DBUS_STARTER_ADDRESS"))
        return FALSE;

    _dbus_verbose (" \\\n",
bus_connection_addresses[DBUS_BUS_STARTER] ?
        bus_connection_addresses[DBUS_BUS_STARTER] :
"none set");
}

if (bus_connection_addresses[DBUS_BUS_STARTER] != NULL)
{
    s = _dbus_getenv ("DBUS_STARTER_BUS_TYPE");

    if (s != NULL)
    {
        _dbus_verbose ("Bus activation type was set to
\\n", s);

        if (strcmp (s, "system") == 0)
            activation_bus_type = DBUS_BUS_SYSTEM;
        else if (strcmp (s, "session") == 0)
            activation_bus_type = DBUS_BUS_SESSION;
    }
}
else
{
    /* Default to the session bus instead if available */
    if (bus_connection_addresses[DBUS_BUS_SESSION] != NULL)
    {
        bus_connection_addresses[DBUS_BUS_STARTER] =

```

```

        _dbus_strdup
(bus_connection_addresses[DBUS_BUS_SESSION]);
        if (bus_connection_addresses[DBUS_BUS_STARTER] == NULL)
            return FALSE;
    }
}

/* If we return FALSE we have to be sure that restarting
 * the above code will work right
 */

if (!_dbus_setenv ("DBUS_ACTIVATION_ADDRESS", NULL))
    return FALSE;

if (!_dbus_setenv ("DBUS_ACTIVATION_BUS_TYPE", NULL))
    return FALSE;

if (!_dbus_register_shutdown_func (addresses_shutdown_func,
                                   NULL))
    return FALSE;

    initialized = TRUE;
}

return initialized;
}

static void
bus_data_free (void *data)
{
    BusData *bd = data;

    if (bd->is_well_known)
    {
        int i;
        _DBUS_LOCK (bus);
        /* We may be stored in more than one slot */
        /* This should now be impossible - these slots are supposed to
         * be cleared on disconnect, so should not need to be cleared on
         * finalize
         */
        i = 0;
        while (i < N_BUS_TYPES)
        {
            if (bus_connections[i] == bd->connection)
                bus_connections[i] = NULL;

            ++i;
        }
        _DBUS_UNLOCK (bus);
    }
}

```

```

    dbus_free (bd->unique_name);
    dbus_free (bd);

    dbus_connection_free_data_slot (&bus_data_slot);
}

static BusData*
ensure_bus_data (DBusConnection *connection)
{
    BusData *bd;

    if (!dbus_connection_allocate_data_slot (&bus_data_slot))
        return NULL;

    bd = dbus_connection_get_data (connection, bus_data_slot);
    if (bd == NULL)
    {
        bd = dbus_new0 (BusData, 1);
        if (bd == NULL)
        {
            dbus_connection_free_data_slot (&bus_data_slot);
            return NULL;
        }

        bd->connection = connection;

        if (!dbus_connection_set_data (connection, bus_data_slot, bd,
                                       bus_data_free))
        {
            dbus_free (bd);
            dbus_connection_free_data_slot (&bus_data_slot);
            return NULL;
        }

        /* Data slot refcount now held by the BusData */
    }
    else
    {
        dbus_connection_free_data_slot (&bus_data_slot);
    }

    return bd;
}

/**
 * Internal function that checks to see if this
 * is a shared connection owned by the bus and if it is unref it.
 *
 * @param connection a connection that has been disconnected.
 */
void

```

```

_dbus_bus_notify_shared_connection_disconnected_unlocked
(DBusConnection *connection)
{
    int i;

    _DBUS_LOCK (bus);

    /* We are expecting to have the connection saved in only one of
these
* slots, but someone could in a pathological case set system and
session
* bus to the same bus or something. Or set one of them to the
starter
* bus without setting the starter bus type in the env variable.
* So we don't break the loop as soon as we find a match.
*/
    for (i = 0; i < N_BUS_TYPES; ++i)
    {
        if (bus_connections[i] == connection)
        {
            bus_connections[i] = NULL;
        }
    }

    _DBUS_UNLOCK (bus);
}

static DBusConnection *
internal_bus_get (DBusBusType type,
                 dbus_bool_t private,
                 DBusError *error)
{
    const char *address;
    DBusConnection *connection;
    BusData *bd;
    DBusBusType address_type;

    _dbus_return_val_if_fail (type >= 0 && type < N_BUS_TYPES, NULL);
    _dbus_return_val_if_error_is_set (error, NULL);

    connection = NULL;

    _DBUS_LOCK (bus);

    if (!init_connections_unlocked ())
    {
        _DBUS_SET_OOM (error);
        goto out;
    }

    /* We want to use the activation address even if the
* activating bus is the session or system bus,

```

```

    * per the spec.
    */
address_type = type;

/* Use the real type of the activation bus for getting its
 * connection, but only if the real type's address is available. (If
 * the activating bus isn't a well-known bus then
 * activation_bus_type == DBUS_BUS_STARTER)
 */
if (type == DBUS_BUS_STARTER &&
    bus_connection_addresses[activation_bus_type] != NULL)
    type = activation_bus_type;

if (!private && bus_connections[type] != NULL)
{
    connection = bus_connections[type];
    dbus_connection_ref (connection);
    goto out;
}

address = bus_connection_addresses[address_type];
if (address == NULL)
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
        "Unable to determine the address of the message
bus (try 'man dbus-launch' and 'man dbus-daemon' for help)");
    goto out;
}

if (private)
    connection = dbus_connection_open_private (address, error);
else
    connection = dbus_connection_open (address, error);

if (!connection)
{
    goto out;
}

if (!dbus_bus_register (connection, error))
{
    _dbus_connection_close_possibly_shared (connection);
    dbus_connection_unref (connection);
    connection = NULL;
    goto out;
}

if (!private)
{
    /* store a weak ref to the connection (dbus-connection.c is
     * supposed to have a strong ref that it drops on disconnect,
     * since this is a shared connection)

```

```

        */
        bus_connections[type] = connection;
    }

    /* By default we're bound to the lifecycle of
     * the message bus.
     */
    dbus_connection_set_exit_on_disconnect (connection,
                                           TRUE);

    _DBUS_LOCK (bus_datas);
    bd = ensure_bus_data (connection);
    _dbus_assert (bd != NULL); /* it should have been created on
                               register, so OOM not possible */
    bd->is_well_known = TRUE;
    _DBUS_UNLOCK (bus_datas);

out:
    /* Return a reference to the caller, or NULL with error set. */
    if (connection == NULL)
        _DBUS_ASSERT_ERROR_IS_SET (error);

    _DBUS_UNLOCK (bus);
    return connection;
}

/** @} */ /* end of implementation details docs */

/**
 * @addtogroup DDBus
 * @{
 */

/**
 * Connects to a bus daemon and registers the client with it. If a
 * connection to the bus already exists, then that connection is
 * returned. The caller of this function owns a reference to the bus.
 *
 * The caller may NOT call dbus_connection_close() on this connection;
 * see dbus_connection_open() and dbus_connection_close() for details
 * on that.
 *
 * If this function obtains a new connection object never before
 * returned from dbus_bus_get(), it will call
 * dbus_connection_set_exit_on_disconnect(), so the application
 * will exit if the connection closes. You can undo this
 * by calling dbus_connection_set_exit_on_disconnect() yourself
 * after you get the connection.
 *
 * dbus_bus_get() calls dbus_bus_register() for you.
 */

```

```

* If returning a newly-created connection, this function will block
* until authentication and bus registration are complete.
*
* @param type bus type
* @param error address where an error can be returned.
* @returns a #DBusConnection with new ref
*/
DBusConnection *
dbus_bus_get (DBusBusType type,
             DBusError *error)
{
    return internal_bus_get (type, FALSE, error);
}

/**
 * Connects to a bus daemon and registers the client with it as with
 * dbus_bus_register(). Unlike dbus_bus_get(), always creates a new
 * connection. This connection will not be saved or recycled by
 * libdbus. Caller owns a reference to the bus and must either close
 * it or know it to be closed prior to releasing this reference.
 *
 * See dbus_connection_open_private() for more details on when to
 * close and unref this connection.
 *
 * This function calls
 * dbus_connection_set_exit_on_disconnect() on the new connection, so
the application
 * will exit if the connection closes. You can undo this
 * by calling dbus_connection_set_exit_on_disconnect() yourself
 * after you get the connection.
 *
 * dbus_bus_get_private() calls dbus_bus_register() for you.
 *
 * This function will block until authentication and bus registration
 * are complete.
 *
 * @param type bus type
 * @param error address where an error can be returned.
 * @returns a DBusConnection with new ref
 */
DBusConnection *
dbus_bus_get_private (DBusBusType type,
                    DBusError *error)
{
    return internal_bus_get (type, TRUE, error);
}

/**
 * Registers a connection with the bus. This must be the first
 * thing an application does when connecting to the message bus.
 * If registration succeeds, the unique name will be set,
 * and can be obtained using dbus_bus_get_unique_name().

```

```

*
* This function will block until registration is complete.
*
* If the connection has already registered with the bus
* (determined by checking whether dbus_bus_get_unique_name()
* returns a non-#NULL value), then this function does nothing.
*
* If you use dbus_bus_get() or dbus_bus_get_private() this
* function will be called for you.
*
* @note Just use dbus_bus_get() or dbus_bus_get_private() instead of
* dbus_bus_register() and save yourself some pain. Using
* dbus_bus_register() manually is only useful if you have your
* own custom message bus not found in #DBusBusType.
*
* If you open a bus connection with dbus_connection_open() or
* dbus_connection_open_private() you will have to dbus_bus_register()
* yourself, or make the appropriate registration method calls
* yourself. If you send the method calls yourself, call
* dbus_bus_set_unique_name() with the unique bus name you get from
* the bus.
*
* For shared connections (created with dbus_connection_open()) in a
* multithreaded application, you can't really make the registration
* calls yourself, because you don't know whether some other thread is
* also registering, and the bus will kick you off if you send two
* registration messages.
*
* If you use dbus_bus_register() however, there is a lock that
* keeps both apps from registering at the same time.
*
* The rule in a multithreaded app, then, is that dbus_bus_register()
* must be used to register, or you need to have your own locks that
* all threads in the app will respect.
*
* In a single-threaded application you can register by hand instead
* of using dbus_bus_register(), as long as you check
* dbus_bus_get_unique_name() to see if a unique name has already been
* stored by another thread before you send the registration messages.
*
* @param connection the connection
* @param error place to store errors
* @returns #TRUE on success
*/
dbus_bool_t
dbus_bus_register (DBusConnection *connection,
                  DBusError      *error)
{
    DBusMessage *message, *reply;
    char *name;
    BusData *bd;
    dbus_bool_t retval;

```



```

_dbus_return_val_if_fail (connection != NULL, FALSE);
_dbus_return_val_if_error_is_set (error, FALSE);

retval = FALSE;
message = NULL;
reply = NULL;

_DBUS_LOCK (bus_datas);

bd = ensure_bus_data (connection);
if (bd == NULL)
{
    _DBUS_SET_OOM (error);
    goto out;
}

if (bd->unique_name != NULL)
{
    _dbus_verbose ("Ignoring attempt to register the same
DBusConnection %s with the message bus a second time.\n",
                  bd->unique_name);
    /* Success! */
    retval = TRUE;
    goto out;
}

message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                        DBUS_PATH_DBUS,
                                        DBUS_INTERFACE_DBUS,
                                        "Hello");

if (!message)
{
    _DBUS_SET_OOM (error);
    goto out;
}

reply = dbus_connection_send_with_reply_and_block (connection,
message, -1, error);

if (reply == NULL)
    goto out;
else if (dbus_set_error_from_message (error, reply))
    goto out;
else if (!dbus_message_get_args (reply, error,
                                DBUS_TYPE_STRING, &name,
                                DBUS_TYPE_INVALID))

    goto out;

bd->unique_name = _dbus_strdup (name);
if (bd->unique_name == NULL)

```

```

    {
        _DBUS_SET_OOM (error);
        goto out;
    }

    retval = TRUE;

out:
    _DBUS_UNLOCK (bus_datas);

    if (message)
        dbus_message_unref (message);

    if (reply)
        dbus_message_unref (reply);

    if (!retval)
        _DBUS_ASSERT_ERROR_IS_SET (error);

    return retval;
}

/**
 * Sets the unique name of the connection, as assigned by the message
 * bus. Can only be used if you registered with the bus manually
 * (i.e. if you did not call dbus_bus_register()). Can only be called
 * once per connection. After the unique name is set, you can get it
 * with dbus_bus_get_unique_name().
 *
 * The only reason to use this function is to re-implement the
 * equivalent of dbus_bus_register() yourself. One (probably unusual)
 * reason to do that might be to do the bus registration call
 * asynchronously instead of synchronously.
 *
 * @note Just use dbus_bus_get() or dbus_bus_get_private(), or worst
 * case dbus_bus_register(), instead of messing with this
 * function. There's really no point creating pain for yourself by
 * doing things manually.
 *
 * It's hard to use this function safely on shared connections
 * (created by dbus_connection_open()) in a multithreaded application,
 * because only one registration attempt can be sent to the bus. If
 * two threads are both sending the registration message, there is no
 * mechanism in libdbus itself to avoid sending it twice.
 *
 * Thus, you need a way to coordinate which thread sends the
 * registration attempt; which also means you know which thread
 * will call dbus_bus_set_unique_name(). If you don't know
 * about all threads in the app (for example, if some libraries
 * you're using might start libdbus-using threads), then you
 * need to avoid using this function on shared connections.

```

```

*
* @param connection the connection
* @param unique_name the unique name
* @returns #FALSE if not enough memory
*/
dbus_bool_t
dbus_bus_set_unique_name (DBusConnection *connection,
                        const char      *unique_name)
{
    BusData *bd;
    dbus_bool_t success = FALSE;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (unique_name != NULL, FALSE);

    _DBUS_LOCK (bus_datas);

    bd = ensure_bus_data (connection);
    if (bd == NULL)
        goto out;

    _dbus_assert (bd->unique_name == NULL);

    bd->unique_name = _dbus_strdup (unique_name);
    success = bd->unique_name != NULL;

out:
    _DBUS_UNLOCK (bus_datas);

    return success;
}

/**
 * Gets the unique name of the connection as assigned by the message
 * bus. Only possible after the connection has been registered with
 * the message bus. All connections returned by dbus_bus_get() or
 * dbus_bus_get_private() have been successfully registered.
 *
 * The name remains valid until the connection is freed, and
 * should not be freed by the caller.
 *
 * Other than dbus_bus_get(), there are two ways to set the unique
 * name; one is dbus_bus_register(), the other is
 * dbus_bus_set_unique_name(). You are responsible for calling
 * dbus_bus_set_unique_name() if you register by hand instead of using
 * dbus_bus_register().
 *
 * @param connection the connection
 * @returns the unique name or #NULL on error
 */
const char*
dbus_bus_get_unique_name (DBusConnection *connection)

```

```

{
    BusData *bd;
    const char *unique_name = NULL;

    _dbus_return_val_if_fail (connection != NULL, NULL);

    _DBUS_LOCK (bus_datas);

    bd = ensure_bus_data (connection);
    if (bd == NULL)
        goto out;

    unique_name = bd->unique_name;

out:
    _DBUS_UNLOCK (bus_datas);

    return unique_name;
}

/**
 * Asks the bus to return the UID the named connection authenticated
 * as, if any. Only works on UNIX; only works for connections on the
 * same machine as the bus. If you are not on the same machine as the
 * bus, then calling this is probably a bad idea, since the UID will
 * mean little to your application.
 *
 * For the system message bus you're guaranteed to be on the same
 * machine since it only listens on a UNIX domain socket (at least,
 * as shipped by default).
 *
 * This function only works for connections that authenticated as
 * a UNIX user, right now that includes all bus connections, but
 * it's very possible to have connections with no associated UID.
 * So check for errors and do something sensible if they happen.
 *
 * This function will always return an error on Windows.
 *
 * @param connection the connection
 * @param name a name owned by the connection
 * @param error location to store the error
 * @returns the unix user id, or ((unsigned)-1) if error is set
 */
unsigned long
dbus_bus_get_unix_user (DBusConnection *connection,
                       const char      *name,
                       DBusError       *error)
{
    DBusMessage *message, *reply;
    dbus_uint32_t uid;

    _dbus_return_val_if_fail (connection != NULL, DBUS_UID_UNSET);

```

```

    _dbus_return_val_if_fail (name != NULL, DBUS_UID_UNSET);
    _dbus_return_val_if_fail (_dbus_check_is_valid_bus_name (name),
DBUS_UID_UNSET);
    _dbus_return_val_if_error_is_set (error, DBUS_UID_UNSET);

message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                        DBUS_PATH_DBUS,
                                        DBUS_INTERFACE_DBUS,
                                        "GetConnectionUnixUser");

if (message == NULL)
{
    _DBUS_SET_OOM (error);
    return DBUS_UID_UNSET;
}

if (!dbus_message_append_args (message,
                               DBUS_TYPE_STRING, &name,
                               DBUS_TYPE_INVALID))
{
    dbus_message_unref (message);
    _DBUS_SET_OOM (error);
    return DBUS_UID_UNSET;
}

reply = dbus_connection_send_with_reply_and_block (connection,
message, -1,
                                                    error);

dbus_message_unref (message);

if (reply == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    return DBUS_UID_UNSET;
}

if (dbus_set_error_from_message (error, reply))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return DBUS_UID_UNSET;
}

if (!dbus_message_get_args (reply, error,
                            DBUS_TYPE_UINT32, &uid,
                            DBUS_TYPE_INVALID))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return DBUS_UID_UNSET;
}

```

```

    dbus_message_unref (reply);

    return (unsigned long) uid;
}

/**
 * Asks the bus to return its globally unique ID, as described in the
 * D-Bus specification. For the session bus, this is useful as a way
 * to uniquely identify each user session. For the system bus,
 * probably the bus ID is not useful; instead, use the machine ID
 * since it's accessible without necessarily connecting to the bus and
 * may be persistent beyond a single bus instance (across reboots for
 * example). See dbus_get_local_machine_id().
 *
 * In addition to an ID for each bus and an ID for each machine, there
is
 * an ID for each address that the bus is listening on; that can
 * be retrieved with dbus_connection_get_server_id(), though it is
 * probably not very useful.
 *
 * @param connection the connection
 * @param error location to store the error
 * @returns the bus ID or #NULL if error is set
 */
char*
dbus_bus_get_id (DBusConnection *connection,
                DBusError      *error)
{
    DBusMessage *message, *reply;
    char *id;
    const char *v_STRING;

    _dbus_return_val_if_fail (connection != NULL, NULL);
    _dbus_return_val_if_error_is_set (error, NULL);

    message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                           DBUS_PATH_DBUS,
                                           DBUS_INTERFACE_DBUS,
                                           "GetId");

    if (message == NULL)
    {
        _DBUS_SET_OOM (error);
        return NULL;
    }

    reply = dbus_connection_send_with_reply_and_block (connection,
message, -1,
                                                    error);

    dbus_message_unref (message);

```

```

if (reply == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    return NULL;
}

if (dbus_set_error_from_message (error, reply))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return NULL;
}

v_STRING = NULL;
if (!dbus_message_get_args (reply, error,
                            DBUS_TYPE_STRING, &v_STRING,
                            DBUS_TYPE_INVALID))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return NULL;
}

id = _dbus_strdup (v_STRING); /* may be NULL */

dbus_message_unref (reply);

if (id == NULL)
    _DBUS_SET_OOM (error);

/* FIXME it might be nice to cache the ID locally */

return id;
}

/**
 * Asks the bus to assign the given name to this connection by
invoking
 * the RequestName method on the bus. This method is fully documented
 * in the D-Bus specification. For quick reference, the flags and
 * result codes are discussed here, but the specification is the
 * canonical version of this information.
 *
 * First you should know that for each bus name, the bus stores
 * a queue of connections that would like to own it. Only
 * one owns it at a time - called the primary owner. If the primary
 * owner releases the name or disconnects, then the next owner in the
 * queue atomically takes over.
 *
 * So for example if you have an application
org.freedesktop.TextEditor

```

* and multiple instances of it can be run, you can have all of them
* sitting in the queue. The first one to start up will receive
messages
* sent to org.freedesktop.TextEditor, but if that one exits another
* will become the primary owner and receive messages.
*
* The queue means you don't need to manually watch for the current
owner to
* disappear and then request the name again.
*
* When requesting a name, you can specify several flags.
*
* #DBUS_NAME_FLAG_ALLOW_REPLACEMENT and #DBUS_NAME_FLAG_DO_NOT_QUEUE
* are properties stored by the bus for this connection with respect
to
* each requested bus name. These properties are stored even if the
* connection is queued and does not become the primary owner.
* You can update these flags by calling RequestName again (even if
* you already own the name).
*
* #DBUS_NAME_FLAG_ALLOW_REPLACEMENT means that another requestor of
the
* name can take it away from you by specifying
#DBUS_NAME_FLAG_REPLACE_EXISTING.
*
* #DBUS_NAME_FLAG_DO_NOT_QUEUE means that if you aren't the primary
owner,
* you don't want to be queued up - you only care about being the
* primary owner.
*
* Unlike the other two flags, #DBUS_NAME_FLAG_REPLACE_EXISTING is a
property
* of the individual RequestName call, i.e. the bus does not
persistently
* associate it with the connection-name pair. If a RequestName call
includes
* the #DBUS_NAME_FLAG_REPLACE_EXISTING flag, and the current primary
* owner has #DBUS_NAME_FLAG_ALLOW_REPLACEMENT set, then the current
primary
* owner will be kicked off.
*
* If no flags are given, an application will receive the requested
* name only if the name is currently unowned; and it will NOT give
* up the name if another application asks to take it over using
* #DBUS_NAME_FLAG_REPLACE_EXISTING.
*
* This function returns a result code. The possible result codes
* are as follows.
*
* #DBUS_REQUEST_NAME_REPLY_PRIMARY_OWNER means that the name had no
* existing owner, and the caller is now the primary owner; or that
* the name had an owner, and the caller specified


```

* #DBUS_NAME_FLAG_REPLACE_EXISTING, and the current owner
* specified #DBUS_NAME_FLAG_ALLOW_REPLACEMENT.
*
* #DBUS_REQUEST_NAME_REPLY_IN_QUEUE happens only if the caller does
NOT
* specify #DBUS_NAME_FLAG_DO_NOT_QUEUE and either the current owner
* did NOT specify #DBUS_NAME_FLAG_ALLOW_REPLACEMENT or the caller did
NOT
* specify #DBUS_NAME_FLAG_REPLACE_EXISTING. In this case the caller
ends up
* in a queue to own the name after the current owner gives it up.
*
* #DBUS_REQUEST_NAME_REPLY_EXISTS happens if the name has an owner
* already and the caller specifies #DBUS_NAME_FLAG_DO_NOT_QUEUE
* and either the current owner has NOT specified
* #DBUS_NAME_FLAG_ALLOW_REPLACEMENT or the caller did NOT specify
* #DBUS_NAME_FLAG_REPLACE_EXISTING.
*
* #DBUS_REQUEST_NAME_REPLY_ALREADY_OWNER happens if an application
* requests a name it already owns. (Re-requesting a name is useful if
* you want to change the #DBUS_NAME_FLAG_ALLOW_REPLACEMENT or
* #DBUS_NAME_FLAG_DO_NOT_QUEUE settings.)
*
* When a service represents an application, say "text editor," then
* it should specify #DBUS_NAME_FLAG_ALLOW_REPLACEMENT if it wants
* the last editor started to be the user's editor vs. the first one
* started. Then any editor that can be the user's editor should
* specify #DBUS_NAME_FLAG_REPLACE_EXISTING to either take over
* (last-started-wins) or be queued up (first-started-wins) according
* to whether #DBUS_NAME_FLAG_ALLOW_REPLACEMENT was given.
*
* Conventionally, single-instance applications often offer a command
* line option called --replace which means to replace the current
* instance. To implement this, always set
* #DBUS_NAME_FLAG_ALLOW_REPLACEMENT when you request your
* application's bus name. When you lose ownership of your bus name,
* you need to exit. Look for the signal "NameLost" from
* #DBUS_SERVICE_DBUS and #DBUS_INTERFACE_DBUS (the signal's first
* argument is the bus name that was lost). If starting up without
* --replace, do not specify #DBUS_NAME_FLAG_REPLACE_EXISTING, and
* exit if you fail to become the bus name owner. If --replace is
* given, ask to replace the old owner.
*
* @param connection the connection
* @param name the name to request
* @param flags flags
* @param error location to store the error
* @returns a result code, -1 if error is set
*/
int
dbus_bus_request_name (DBusConnection *connection,
                      const char      *name,

```

```

                unsigned int    flags,
                DBusError      *error)
{
    DBusMessage *message, *reply;
    dbus_uint32_t result;

    _dbus_return_val_if_fail (connection != NULL, 0);
    _dbus_return_val_if_fail (name != NULL, 0);
    _dbus_return_val_if_fail (_dbus_check_is_valid_bus_name (name), 0);
    _dbus_return_val_if_error_is_set (error, 0);

    message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                           DBUS_PATH_DBUS,
                                           DBUS_INTERFACE_DBUS,
                                           "RequestName");

    if (message == NULL)
    {
        _DBUS_SET_OOM (error);
        return -1;
    }

    if (!dbus_message_append_args (message,
                                   DBUS_TYPE_STRING, &name,
                                   DBUS_TYPE_UINT32, &flags,
                                   DBUS_TYPE_INVALID))
    {
        dbus_message_unref (message);
        _DBUS_SET_OOM (error);
        return -1;
    }

    reply = dbus_connection_send_with_reply_and_block (connection,
                                                       message, -1,
                                                       error);

    dbus_message_unref (message);

    if (reply == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        return -1;
    }

    if (dbus_set_error_from_message (error, reply))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        dbus_message_unref (reply);
        return -1;
    }

    if (!dbus_message_get_args (reply, error,

```

```

        DBUS_TYPE_UINT32, &result,
        DBUS_TYPE_INVALID))
    {
        _dbus_assert_error_is_set (error);
        dbus_message_unref (reply);
        return -1;
    }

    dbus_message_unref (reply);

    return result;
}

/**
 * Asks the bus to unassign the given name from this connection by
 * invoking the ReleaseName method on the bus. The "ReleaseName"
 * method is canonically documented in the D-Bus specification.
 *
 * Possible results are: #DBUS_RELEASE_NAME_REPLY_RELEASED
 * which means you owned the name or were in the queue to own it,
 * and and now you don't own it and aren't in the queue.
 * #DBUS_RELEASE_NAME_REPLY_NOT_OWNER which means someone else
 * owns the name so you can't release it.
 * #DBUS_RELEASE_NAME_REPLY_NON_EXISTENT
 * which means nobody owned the name.
 *
 * @param connection the connection
 * @param name the name to remove
 * @param error location to store the error
 * @returns a result code, -1 if error is set
 */
int
dbus_bus_release_name (DBusConnection *connection,
                      const char      *name,
                      DBusError       *error)
{
    DBusMessage *message, *reply;
    dbus_uint32_t result;

    _dbus_return_val_if_fail (connection != NULL, 0);
    _dbus_return_val_if_fail (name != NULL, 0);
    _dbus_return_val_if_fail (_dbus_check_is_valid_bus_name (name), 0);
    _dbus_return_val_if_error_is_set (error, 0);

    message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                           DBUS_PATH_DBUS,
                                           DBUS_INTERFACE_DBUS,
                                           "ReleaseName");

    if (message == NULL)
    {

```

```

    _DBUS_SET_OOM (error);
    return -1;
}

if (!dbus_message_append_args (message,
                               DBUS_TYPE_STRING, &name,
                               DBUS_TYPE_INVALID))
{
    dbus_message_unref (message);
    _DBUS_SET_OOM (error);
    return -1;
}

reply = dbus_connection_send_with_reply_and_block (connection,
message, -1,
                                                    error);

dbus_message_unref (message);

if (reply == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    return -1;
}

if (dbus_set_error_from_message (error, reply))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return -1;
}

if (!dbus_message_get_args (reply, error,
                            DBUS_TYPE_UINT32, &result,
                            DBUS_TYPE_INVALID))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return -1;
}

dbus_message_unref (reply);

return result;
}

/**
 * Asks the bus whether a certain name has an owner.
 *
 * Using this can easily result in a race condition,
 * since an owner can appear or disappear after you
 * call this.

```

```

*
* If you want to request a name, just request it;
* if you want to avoid replacing a current owner,
* don't specify #DBUS_NAME_FLAG_REPLACE_EXISTING and
* you will get an error if there's already an owner.
*
* @param connection the connection
* @param name the name
* @param error location to store any errors
* @returns #TRUE if the name exists, #FALSE if not or on error
*/
dbus_bool_t
dbus_bus_name_has_owner (DBusConnection *connection,
                        const char      *name,
                        DBusError       *error)
{
    DBusMessage *message, *reply;
    dbus_bool_t exists;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (name != NULL, FALSE);
    _dbus_return_val_if_fail (_dbus_check_is_valid_bus_name (name),
FALSE);
    _dbus_return_val_if_error_is_set (error, FALSE);

    message = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                           DBUS_PATH_DBUS,
                                           DBUS_INTERFACE_DBUS,
                                           "NameHasOwner");

    if (message == NULL)
    {
        _DBUS_SET_OOM (error);
        return FALSE;
    }

    if (!dbus_message_append_args (message,
                                  DBUS_TYPE_STRING, &name,
                                  DBUS_TYPE_INVALID))
    {
        dbus_message_unref (message);
        _DBUS_SET_OOM (error);
        return FALSE;
    }

    reply = dbus_connection_send_with_reply_and_block (connection,
message, -1, error);
    dbus_message_unref (message);

    if (reply == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        return FALSE;
    }
}

```

```

    }

    if (!dbus_message_get_args (reply, error,
                                DBUS_TYPE_BOOLEAN, &exists,
                                DBUS_TYPE_INVALID))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        dbus_message_unref (reply);
        return FALSE;
    }

    dbus_message_unref (reply);
    return exists;
}

/**
 * Starts a service that will request ownership of the given name.
 * The returned result will be one of be one of
 * #DBUS_START_REPLY_SUCCESS or #DBUS_START_REPLY_ALREADY_RUNNING if
 * successful. Pass #NULL if you don't care about the result.
 *
 * The flags parameter is for future expansion, currently you should
 * specify 0.
 *
 * It's often easier to avoid explicitly starting services, and
 * just send a method call to the service's bus name instead.
 * Method calls start a service to handle them by default
 * unless you call dbus_message_set_auto_start() to disable this
 * behavior.
 *
 * @param connection the connection
 * @param name the name we want the new service to request
 * @param flags the flags (should always be 0 for now)
 * @param result a place to store the result or #NULL
 * @param error location to store any errors
 * @returns #TRUE if the activation succeeded, #FALSE if not
 */
dbus_bool_t
dbus_bus_start_service_by_name (DBusConnection *connection,
                                const char *name,
                                dbus_uint32_t flags,
                                dbus_uint32_t *result,
                                DBusError *error)
{
    DBusMessage *msg;
    DBusMessage *reply;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (_dbus_check_is_valid_bus_name (name),
    FALSE);

    msg = dbus_message_new_method_call (DBUS_SERVICE_DBUS,

```

```

        DBUS_PATH_DBUS,
        DBUS_INTERFACE_DBUS,
        "StartServiceByName");

if (!dbus_message_append_args (msg, DBUS_TYPE_STRING, &name,
                               DBUS_TYPE_UINT32, &flags, DBUS_TYPE_INVALID))
{
    dbus_message_unref (msg);
    _DBUS_SET_OOM (error);
    return FALSE;
}

reply = dbus_connection_send_with_reply_and_block (connection, msg,
                                                  -1, error);
dbus_message_unref (msg);

if (reply == NULL)
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    return FALSE;
}

if (dbus_set_error_from_message (error, reply))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return FALSE;
}

if (result != NULL &&
    !dbus_message_get_args (reply, error, DBUS_TYPE_UINT32,
                           result, DBUS_TYPE_INVALID))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    dbus_message_unref (reply);
    return FALSE;
}

dbus_message_unref (reply);
return TRUE;
}

static void
send_no_return_values (DBusConnection *connection,
                      DBusMessage *msg,
                      DBusError *error)
{
    if (error)
    {
        /* Block to check success codepath */
        DBusMessage *reply;
    }
}

```

```

        reply = dbus_connection_send_with_reply_and_block (connection,
msg,
                                                    -1, error);

        if (reply == NULL)
            _DBUS_ASSERT_ERROR_IS_SET (error);
        else
            dbus_message_unref (reply);
    }
    else
    {
        /* Silently-fail nonblocking codepath */
        dbus_message_set_no_reply (msg, TRUE);
        dbus_connection_send (connection, msg, NULL);
    }
}

/**
 * Adds a match rule to match messages going through the message bus.
 * The "rule" argument is the string form of a match rule.
 *
 * If you pass #NULL for the error, this function will not
 * block; the match thus won't be added until you flush the
 * connection, and if there's an error adding the match
 * you won't find out about it. This is generally acceptable, since
the
 * possible errors (including a lack of resources in the bus, the
connection
 * having exceeded its quota of active match rules, or the match rule
being
 * unparseable) are generally unrecoverable.
 *
 * If you pass non-#NULL for the error this function will
 * block until it gets a reply. This may be useful when using match
rule keys
 * introduced in recent versions of D-Bus, like 'arg0namespace', to
allow the
 * application to fall back to less efficient match rules supported by
older
 * versions of the daemon if the running version is not new enough; or
when
 * using user-supplied rules rather than rules hard-coded at compile
time.
 *
 * Normal API conventions would have the function return
 * a boolean value indicating whether the error was set,
 * but that would require blocking always to determine
 * the return value.
 *
 * The AddMatch method is fully documented in the D-Bus
 * specification. For quick reference, the format of the
 * match rules is discussed here, but the specification

```


* is the canonical version of this information.
*
* Rules are specified as a string of comma separated
* key/value pairs. An example is
* "type='signal',sender='org.freedesktop.DBus',
* interface='org.freedesktop.DBus',member='Foo',
* path='/bar/foo',destination=':452345.34'"
*
* Possible keys you can match on are type, sender,
* interface, member, path, destination and numbered
* keys to match message args (keys are 'arg0', 'arg1', etc.).
* Omitting a key from the rule indicates
* a wildcard match. For instance omitting
* the member from a match rule but adding a sender would
* let all messages from that sender through regardless of
* the member.
*
* Matches are inclusive not exclusive so as long as one
* rule matches the message will get through. It is important
* to note this because every time a message is received the
* application will be paged into memory to process it. This
* can cause performance problems such as draining batteries
* on embedded platforms.
*
* If you match message args ('arg0', 'arg1', and so forth)
* only string arguments will match. That is, arg0='5' means
* match the string "5" not the integer 5.
*
* Currently there is no way to match against non-string arguments.
*
* A specialised form of wildcard matching on arguments is
* supported for path-like namespaces. If your argument match has
* a 'path' suffix (eg: "arg0path='/some/path/'") then it is
* considered a match if the argument exactly matches the given
* string or if one of them ends in a '/' and is a prefix of the
* other.
*
* Matching on interface is tricky because method call
* messages only optionally specify the interface.
* If a message omits the interface, then it will NOT match
* if the rule specifies an interface name. This means match
* rules on method calls should not usually give an interface.
*
* However, signal messages are required to include the interface
* so when matching signals usually you should specify the interface
* in the match rule.
*
* For security reasons, you can match arguments only up to
* #DBUS_MAXIMUM_MATCH_RULE_ARG_NUMBER.
*
* Match rules have a maximum length of
#DBUS_MAXIMUM_MATCH_RULE_LENGTH

```

* bytes.
*
* Both of these maximums are much higher than you're likely to need,
* they only exist because the D-Bus bus daemon has fixed limits on
* all resource usage.
*
* @param connection connection to the message bus
* @param rule textual form of match rule
* @param error location to store any errors
*/
void
dbus_bus_add_match (DBusConnection *connection,
                   const char      *rule,
                   DBusError        *error)
{
    DBusMessage *msg;

    _dbus_return_if_fail (rule != NULL);

    msg = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                       DBUS_PATH_DBUS,
                                       DBUS_INTERFACE_DBUS,
                                       "AddMatch");

    if (msg == NULL)
    {
        _DBUS_SET_OOM (error);
        return;
    }

    if (!dbus_message_append_args (msg, DBUS_TYPE_STRING, &rule,
                                   DBUS_TYPE_INVALID))
    {
        dbus_message_unref (msg);
        _DBUS_SET_OOM (error);
        return;
    }

    send_no_return_values (connection, msg, error);

    dbus_message_unref (msg);
}

/**
 * Removes a previously-added match rule "by value" (the most
 * recently-added identical rule gets removed).  The "rule" argument
 * is the string form of a match rule.
 *
 * The bus compares match rules semantically, not textually, so
 * whitespace and ordering don't have to be identical to
 * the rule you passed to dbus_bus_add_match().
 */

```

```

* If you pass #NULL for the error, this function will not
* block; otherwise it will. See detailed explanation in
* docs for dbus_bus_add_match().
*
* @param connection connection to the message bus
* @param rule textual form of match rule
* @param error location to store any errors
*/
void
dbus_bus_remove_match (DBusConnection *connection,
                      const char      *rule,
                      DBusError       *error)
{
    DBusMessage *msg;

    _dbus_return_if_fail (rule != NULL);

    msg = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                        DBUS_PATH_DBUS,
                                        DBUS_INTERFACE_DBUS,
                                        "RemoveMatch");

    if (!dbus_message_append_args (msg, DBUS_TYPE_STRING, &rule,
                                   DBUS_TYPE_INVALID))
    {
        dbus_message_unref (msg);
        _DBUS_SET_OOM (error);
        return;
    }

    send_no_return_values (connection, msg, error);

    dbus_message_unref (msg);
}

/** @} */

```

File = dbus-bus.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-bus.h Convenience functions for communicating with the bus.
 *
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by

```

```

* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_BUS_H
#define DBUS_BUS_H

#include <dbus/dbus-connection.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusBus
 * @{
 */

DBUS_EXPORT
DBusConnection *dbus_bus_get                (DBusBusType      type,
                                             DBusError        *error);

DBUS_EXPORT
DBusConnection *dbus_bus_get_private       (DBusBusType      type,
                                             DBusError        *error);

DBUS_EXPORT
dbus_bool_t     dbus_bus_register          (DBusConnection *connection,
                                             DBusError        *error);

DBUS_EXPORT
dbus_bool_t     dbus_bus_set_unique_name   (DBusConnection *connection,
                                             const char        *unique_name);

DBUS_EXPORT
const char*     dbus_bus_get_unique_name   (DBusConnection
*connection);

DBUS_EXPORT
unsigned long   dbus_bus_get_unix_user     (DBusConnection *connection,
                                             const char        *name,
                                             DBusError        *error);

DBUS_EXPORT

```

```

char*          dbus_bus_get_id          (DBusConnection *connection,
                                         DBusError      *error);

DBUS_EXPORT
int            dbus_bus_request_name    (DBusConnection *connection,
                                         const char      *name,
                                         unsigned int    flags,
                                         DBusError      *error);

DBUS_EXPORT
int            dbus_bus_release_name    (DBusConnection *connection,
                                         const char      *name,
                                         DBusError      *error);

DBUS_EXPORT
dbus_bool_t    dbus_bus_name_has_owner (DBusConnection *connection,
                                         const char      *name,
                                         DBusError      *error);

DBUS_EXPORT
dbus_bool_t    dbus_bus_start_service_by_name (DBusConnection
                                                const char      *name,
                                                dbus_uint32_t  flags,
                                                dbus_uint32_t  *reply,
                                                DBusError
*error);

DBUS_EXPORT
void           dbus_bus_add_match       (DBusConnection *connection,
                                         const char      *rule,
                                         DBusError      *error);

DBUS_EXPORT
void           dbus_bus_remove_match    (DBusConnection *connection,
                                         const char      *rule,
                                         DBusError      *error);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_BUS_H */

```

File = dbus-cleanup-sockets.1

```

.\"
.\" dbus\ -cleanup\ -sockets manual page.
.\" Copyright (C) 2003 Red Hat, Inc.
.\"
.TH dbus\ -cleanup\ -sockets 1
.SH NAME
dbus\ -cleanup\ -sockets \- clean up leftover sockets in a directory
.SH SYNOPSIS

```

```
.PP
.B dbus\cleanup\sockets [DIRECTORY]
```

```
.SH DESCRIPTION
```

The `\fIdbus\cleanup\sockets\fp` command cleans up unused D\Bus connection sockets. See <http://www.freedesktop.org/software/dbus/> for more information about the big picture.

```
.PP
If given no arguments, \fIdbus\cleanup\sockets\fp cleans up sockets in the standard default socket directory for the per\user\login\session message bus; this is usually /tmp. Optionally, you can pass a different directory on the command line.
```

```
.PP
On Linux, this program is essentially useless, because D\Bus defaults to using "abstract sockets" that exist only in memory and don't have a corresponding file in /tmp.
```

```
.PP
On most other flavors of UNIX, it's possible for the socket files to leak when programs using D\Bus exit abnormally or without closing their D\Bus connections. Thus, it might be interesting to run dbus\cleanup\sockets in a cron job to mop up any leaked sockets. Or you can just ignore the leaked sockets, they aren't really hurting anything, other than cluttering the output of "ls /tmp"
```

```
.SH AUTHOR
```

`dbus\cleanup\sockets` was adapted by Havoc Pennington from `linc\cleanup\sockets` written by Michael Meeks.

```
.SH BUGS
```

Please send bug reports to the D\Bus mailing list or bug tracker, see <http://www.freedesktop.org/software/dbus/>

```
File = dbus-connection-internal.h
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-connection-internal.h DBusConnection internal interfaces
 *
 * Copyright (C) 2002 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
```

```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#endif DBUS_CONNECTION_INTERNAL_H
#define DBUS_CONNECTION_INTERNAL_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-connection.h>
#include <dbus/dbus-message.h>
#include <dbus/dbus-transport.h>
#include <dbus/dbus-resources.h>
#include <dbus/dbus-list.h>
#include <dbus/dbus-timeout.h>
#include <dbus/dbus-dataslot.h>

DBUS_BEGIN_DECLS

typedef enum
{
    DBUS_ITERATION_DO_WRITING = 1 << 0, /**< Write messages out. */
    DBUS_ITERATION_DO_READING = 1 << 1, /**< Read messages in. */
    DBUS_ITERATION_BLOCK      = 1 << 2  /**< Block if nothing to do. */
} DBusIterationFlags;

/** default timeout value when waiting for a message reply, 25 seconds
*/
#define _DBUS_DEFAULT_TIMEOUT_VALUE (25 * 1000)

void          _dbus_connection_lock
(DBusConnection *connection);
void          _dbus_connection_unlock
(DBusConnection *connection);
DBusConnection * _dbus_connection_ref_unlocked
(DBusConnection *connection);
void          _dbus_connection_unref_unlocked
(DBusConnection *connection);
void          _dbus_connection_queue_received_message_link
(DBusConnection *connection,

DBusList     *link);
dbus_bool_t  _dbus_connection_has_messages_to_send_unlocked
(DBusConnection *connection);

```

```

DBusMessage*      _dbus_connection_get_message_to_send
(DBusConnection  *connection);
void              _dbus_connection_message_sent_unlocked
(DBusConnection  *connection,

DBusMessage      *message);
dbus_bool_t      _dbus_connection_add_watch_unlocked
(DBusConnection  *connection,

DBusWatch        *watch);
void              _dbus_connection_remove_watch_unlocked
(DBusConnection  *connection,

DBusWatch        *watch);
void              _dbus_connection_toggle_watch_unlocked
(DBusConnection  *connection,

DBusWatch        *watch,

dbus_bool_t      enabled);
dbus_bool_t      _dbus_connection_handle_watch
(DBusWatch       *watch,

unsigned int     condition,

void

*data);
dbus_bool_t      _dbus_connection_add_timeout_unlocked
(DBusConnection  *connection,

DBusTimeout      *timeout);
void              _dbus_connection_remove_timeout_unlocked
(DBusConnection  *connection,

DBusTimeout      *timeout);
void              _dbus_connection_toggle_timeout_unlocked
(DBusConnection  *connection,

DBusTimeout      *timeout,

dbus_bool_t      enabled);
DBusConnection*  _dbus_connection_new_for_transport
(DBusTransport    *transport);
void              _dbus_connection_do_iteration_unlocked
(DBusConnection  *connection,

DBusPendingCall  *pending,

unsigned int     flags,

int

timeout_milliseconds);
void              _dbus_connection_close_possibly_shared
(DBusConnection  *connection);

```



```

void      _dbus_connection_close_if_only_one_ref
(DBusConnection *connection);

DBusPendingCall* _dbus_pending_call_new
(DBusConnection *connection,

int
timeout_milliseconds,

DBusTimeoutHandler timeout_handler);
void      _dbus_pending_call_notify
(DBusPendingCall *pending);
void      _dbus_connection_remove_pending_call
(DBusConnection *connection,

DBusPendingCall *pending);
void      _dbus_connection_block_pending_call
(DBusPendingCall *pending);
void      _dbus_pending_call_complete_and_unlock
(DBusPendingCall *pending,

DBusMessage *message);
dbus_bool_t _dbus_connection_send_and_unlock
(DBusConnection *connection,

DBusMessage *message,

dbus_uint32_t *client_serial);

void      _dbus_connection_queue_synthesized_message_link
(DBusConnection *connection,

DBusList *link);
void      _dbus_connection_test_get_locks
(DBusConnection *conn,

DBusMutex **mutex_loc,

DBusMutex **dispatch_mutex_loc,

DBusMutex **io_path_mutex_loc,

DBusCondVar **dispatch_cond_loc,

DBusCondVar **io_path_cond_loc);

/* if DBUS_ENABLE_STATS */
void _dbus_connection_get_stats (DBusConnection *connection,
dbus_uint32_t *in_messages,
dbus_uint32_t *in_bytes,
dbus_uint32_t *in_fds,
dbus_uint32_t *in_peak_bytes,
dbus_uint32_t *in_peak_fds,
dbus_uint32_t *out_messages,

```

```

                                dbus_uint32_t  *out_bytes,
                                dbus_uint32_t  *out_fds,
                                dbus_uint32_t  *out_peak_bytes,
                                dbus_uint32_t  *out_peak_fds);

/* if DBUS_BUILD_TESTS */
const char* _dbus_connection_get_address (DBusConnection *connection);

/* This _dbus_bus_* stuff doesn't really belong here, but dbus-bus-
internal.h seems
 * silly for one function
 */
/**
 * @addtogroup DBusBusInternals
 * @{
 */

void
_dbus_bus_notify_shared_connection_disconnected_unlocked
(DBusConnection *connection);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_CONNECTION_INTERNAL_H */

```

File = dbus-connection.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-connection.c DBusConnection object
 *
 * Copyright (C) 2002-2006 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 */

```

```
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-shared.h"
#include "dbus-connection.h"
#include "dbus-list.h"
#include "dbus-timeout.h"
#include "dbus-transport.h"
#include "dbus-watch.h"
#include "dbus-connection-internal.h"
#include "dbus-pending-call-internal.h"
#include "dbus-list.h"
#include "dbus-hash.h"
#include "dbus-message-internal.h"
#include "dbus-message-private.h"
#include "dbus-threads.h"
#include "dbus-protocol.h"
#include "dbus-dataslot.h"
#include "dbus-string.h"
#include "dbus-signature.h"
#include "dbus-pending-call.h"
#include "dbus-object-tree.h"
#include "dbus-threads-internal.h"
#include "dbus-bus.h"
#include "dbus-marshal-basic.h"

#ifdef DBUS_DISABLE_CHECKS
#define TOOK_LOCK_CHECK(connection)
#define RELEASING_LOCK_CHECK(connection)
#define HAVE_LOCK_CHECK(connection)
#else
#define TOOK_LOCK_CHECK(connection) do { \
    _dbus_assert (!(connection)->have_connection_lock); \
    (connection)->have_connection_lock = TRUE; \
} while (0)
#define RELEASING_LOCK_CHECK(connection) do { \
    _dbus_assert ((connection)->have_connection_lock); \
    (connection)->have_connection_lock = FALSE; \
} while (0)
#define HAVE_LOCK_CHECK(connection) _dbus_assert ((connection)->have_connection_lock)
/* A "DO_NOT_HAVE_LOCK_CHECK" is impossible since we need the lock to
check the flag */
#endif

#define TRACE_LOCKS 1
```

```

#define CONNECTION_LOCK(connection) do {
\
    if (TRACE_LOCKS) { _dbus_verbose ("LOCK\n"); } \
    _dbus_rmutex_lock ((connection)->mutex);
\
    TOOK_LOCK_CHECK (connection);
\
} while (0)

#define CONNECTION_UNLOCK(connection) _dbus_connection_unlock
(connection)

#define SLOTS_LOCK(connection) do { \
    _dbus_rmutex_lock ((connection)->slot_mutex); \
} while (0)

#define SLOTS_UNLOCK(connection) do { \
    _dbus_rmutex_unlock ((connection)->slot_mutex); \
} while (0)

#define DISPATCH_STATUS_NAME(s)
\
    ((s) == DBUS_DISPATCH_COMPLETE ? "complete" :
\
    (s) == DBUS_DISPATCH_DATA_REMAINS ? "data
remains" : \
    (s) == DBUS_DISPATCH_NEED_MEMORY ? "need memory"
: \
    "???" )

/**
 * @defgroup DBusConnection DBusConnection
 * @ingroup DBus
 * @brief Connection to another application
 *
 * A DBusConnection represents a connection to another
 * application. Messages can be sent and received via this connection.
 * The other application may be a message bus; for convenience, the
 * function dbus_bus_get() is provided to automatically open a
 * connection to the well-known message buses.
 *
 * In brief a DBusConnection is a message queue associated with some
 * message transport mechanism such as a socket. The connection
 * maintains a queue of incoming messages and a queue of outgoing
 * messages.
 *
 * Several functions use the following terms:
 * <ul>
 * <li><b>read</b> means to fill the incoming message queue by reading
from the socket</li>
 * <li><b>write</b> means to drain the outgoing queue by writing to
the socket</li>

```

* `dispatch` means to drain the incoming queue by invoking application-provided message handlers

- *
- * The function `dbus_connection_read_write_dispatch()` for example does all
- * three of these things, offering a simple alternative to a main loop.
- * In an application with a main loop, the read/write/dispatch operations are usually separate.
- * The connection provides `#DBusWatch` and `#DBusTimeout` objects to the main loop. These are used to know when reading, writing, or dispatching should be performed.
- * Incoming messages are processed by calling `dbus_connection_dispatch()`. `dbus_connection_dispatch()` runs any handlers registered for the topmost message in the message queue, then discards the message, then returns.
- * `dbus_connection_get_dispatch_status()` indicates whether messages are currently in the queue that need dispatching.
- * `dbus_connection_set_dispatch_status_function()` allows you to set a function to be used to monitor the dispatch status.
- * If you're using GLib or Qt add-on libraries for D-Bus, there are special convenience APIs in those libraries that hide all the details of dispatch and watch/timeout monitoring. For example, `dbus_connection_setup_with_g_main()`.
- * If you aren't using these add-on libraries, but want to process messages asynchronously, you must manually call `dbus_connection_set_dispatch_status_function()`, `dbus_connection_set_watch_functions()`, `dbus_connection_set_timeout_functions()` providing appropriate functions to integrate the connection with your application's main loop. This can be tricky to get right; main loops are not simple.
- * If you don't need to be asynchronous, you can ignore `#DBusWatch`, `#DBusTimeout`, and `dbus_connection_dispatch()`. Instead, `dbus_connection_read_write_dispatch()` can be used.
- * Or, in *very* simple applications, `dbus_connection_pop_message()` may be all you need, allowing you to avoid setting up any handler functions (see `dbus_connection_add_filter()`, `dbus_connection_register_object_path()` for more on handlers).
- * When you use `dbus_connection_send()` or one of its variants to send a message, the message is added to the outgoing queue. It's actually written to the network later; either in

```

* dbus_watch_handle() invoked by your main loop, or in
* dbus_connection_flush() which blocks until it can write out the
* entire outgoing queue. The GLib/Qt add-on libraries again
* handle the details here for you by setting up watch functions.
*
* When a connection is disconnected, you are guaranteed to get a
* signal "Disconnected" from the interface
* #DBUS_INTERFACE_LOCAL, path
* #DBUS_PATH_LOCAL.
*
* You may not drop the last reference to a #DBusConnection
* until that connection has been disconnected.
*
* You may dispatch the unprocessed incoming message queue even if the
* connection is disconnected. However, "Disconnected" will always be
* the last message in the queue (obviously no messages are received
* after disconnection).
*
* After calling dbus_threads_init(), #DBusConnection has thread
* locks and drops them when invoking user callbacks, so in general is
* transparently threadsafe. However, #DBusMessage does NOT have
* thread locks; you must not send the same message to multiple
* #DBusConnection if those connections will be used from different
threads,
* for example.
*
* Also, if you dispatch or pop messages from multiple threads, it
* may work in the sense that it won't crash, but it's tough to
imagine
* sane results; it will be completely unpredictable which messages
* go to which threads.
*
* It's recommended to dispatch from a single thread.
*
* The most useful function to call from multiple threads at once
* is dbus_connection_send_with_reply_and_block(). That is,
* multiple threads can make method calls at the same time.
*
* If you aren't using threads, you can use a main loop and
* dbus_pending_call_set_notify() to achieve a similar result.
*/

/**
 * @defgroup DBusConnectionInternals DBusConnection implementation
details
 * @ingroup DBusInternals
 * @brief Implementation details of DBusConnection
 *
 * @{
 */

#ifdef DBUS_ENABLE_VERBOSE_MODE

```

```

static void
_dbus_connection_trace_ref (DBusConnection *connection,
    int old_refcount,
    int new_refcount,
    const char *why)
{
    static int enabled = -1;

    _dbus_trace_ref ("DBusConnection", connection, old_refcount,
new_refcount,
        why, "DBUS_CONNECTION_TRACE", &enabled);
}
#else
#define _dbus_connection_trace_ref(c,o,n,w) \
    do \
    {\
        (void) (o); \
        (void) (n); \
    } while (0)
#endif

/**
 * Internal struct representing a message filter function
 */
typedef struct DBusMessageFilter DBusMessageFilter;

/**
 * Internal struct representing a message filter function
 */
struct DBusMessageFilter
{
    DBusAtomic refcount; /**< Reference count */
    DBusHandleMessageFunction function; /**< Function to call to filter
*/
    void *user_data; /**< User data for the function */
    DBusFreeFunction free_user_data_function; /**< Function to free the
user data */
};

/**
 * Internals of DBusPreallocatedSend
 */
struct DBusPreallocatedSend
{
    DBusConnection *connection; /**< Connection we'd send the message to
*/
    DBusList *queue_link; /**< Preallocated link in the queue */
    DBusList *counter_link; /**< Preallocated link in the resource
counter */
};

```

```

#if HAVE_DECL_MSG_NOSIGNAL
static dbus_bool_t _dbus_modify_sigpipe = FALSE;
#else
static dbus_bool_t _dbus_modify_sigpipe = TRUE;
#endif

/**
 * Implementation details of DBusConnection. All fields are private.
 */
struct DBusConnection
{
    DBusAtomic refcount; /**< Reference count. */

    DBusRMutex *mutex; /**< Lock on the entire DBusConnection */

    DBusCMutex *dispatch_mutex; /**< Protects dispatch_acquired */
    DBusCondVar *dispatch_cond; /**< Notify when dispatch_acquired is
available */
    DBusCMutex *io_path_mutex; /**< Protects io_path_acquired */
    DBusCondVar *io_path_cond; /**< Notify when io_path_acquired is
available */

    DBusList *outgoing_messages; /**< Queue of messages we need to send,
send the end of the list first. */
    DBusList *incoming_messages; /**< Queue of messages we have
received, end of the list received most recently. */
    DBusList *expired_messages; /**< Messages that will be released
when we next unlock. */

    DBusMessage *message_borrowed; /**< Filled in if the first incoming
message has been borrowed;
                                *   dispatch_acquired will be set by
the borrower
                                */

    int n_outgoing; /**< Length of outgoing queue. */
    int n_incoming; /**< Length of incoming queue. */

    DBusCounter *outgoing_counter; /**< Counts size of outgoing
messages. */

    DBusTransport *transport; /**< Object that sends/receives
messages over network. */
    DBusWatchList *watches; /**< Stores active watches. */
    DBusTimeoutList *timeouts; /**< Stores active timeouts. */

    DBusList *filter_list; /**< List of filters. */

    DBusRMutex *slot_mutex; /**< Lock on slot_list so overall
connection lock need not be taken */
    DBusDataSlotList slot_list; /**< Data stored by allocated integer
ID */

```



```

    DBusHashTable *pending_replies; /**< Hash of message serials to
#DBusPendingCall. */

    dbus_uint32_t client_serial; /**< Client serial. Increments
each time a message is sent */
    DBusList *disconnect_message_link; /**< Preallocated list node for
queueing the disconnection message */

    DBusWakeupMainFunction wakeup_main_function; /**< Function to wake
up the mainloop */
    void *wakeup_main_data; /**< Application data for
wakeup_main_function */
    DBusFreeFunction free_wakeup_main_data; /**< free wakeup_main_data
*/

    DBusDispatchStatusFunction dispatch_status_function; /**< Function
on dispatch status changes */
    void *dispatch_status_data; /**< Application data for
dispatch_status_function */
    DBusFreeFunction free_dispatch_status_data; /**< free
dispatch_status_data */

    DBusDispatchStatus last_dispatch_status; /**< The last dispatch
status we reported to the application. */

    DBusObjectTree *objects; /**< Object path handlers registered with
this connection */

    char *server_guid; /**< GUID of server if we are in
shared_connections, #NULL if server GUID is unknown or connection is
private */

    /* These two MUST be bools and not bitfields, because they are
protected by a separate lock
    * from connection->mutex and all bitfields in a word have to be
read/written together.
    * So you can't have a different lock for different bitfields in the
same word.
    */
    dbus_bool_t dispatch_acquired; /**< Someone has dispatch path (can
drain incoming queue) */
    dbus_bool_t io_path_acquired; /**< Someone has transport io path
(can use the transport to read/write messages) */

    unsigned int shareable : 1; /**< #TRUE if libdbus owns a reference
to the connection and can return it from dbus_connection_open() more
than once */

    unsigned int exit_on_disconnect : 1; /**< If #TRUE, exit after
handling disconnect signal */

```

```

    unsigned int route_peer_messages : 1; /**< If #TRUE, if
org.freedesktop.DBus.Peer messages have a bus name, don't handle them
automatically */

    unsigned int disconnected_message_arrived : 1; /**< We popped or
are dispatching the disconnected message.

                                * if the
disconnect_message_link is NULL then we queued it, but
                                * this flag is
whether it got to the head of the queue.
                                */

    unsigned int disconnected_message_processed : 1; /**< We did our
default handling of the disconnected message,
                                * such as closing
the connection.
                                */

#ifdef DBUS_DISABLE_CHECKS
    unsigned int have_connection_lock : 1; /**< Used to check locking */
#endif

#ifdef DBUS_DISABLE_CHECKS
    int generation; /**< _dbus_current_generation that should correspond
to this connection */
#endif
};

static DBusDispatchStatus
_dbus_connection_get_dispatch_status_unlocked (DBusConnection
*connection);
static void
_dbus_connection_update_dispatch_status_and_unlock (DBusConnection
*connection,
DBusDispatchStatus new_status);
static void
_dbus_connection_last_unref
(DBusConnection *connection);
static void
_dbus_connection_acquire_dispatch
(DBusConnection *connection);
static void
_dbus_connection_release_dispatch
(DBusConnection *connection);
static DBusDispatchStatus _dbus_connection_flush_unlocked
(DBusConnection *connection);
static void
_dbus_connection_close_possibly_shared_and_unlock (DBusConnection
*connection);
static dbus_bool_t
_dbus_connection_get_is_connected_unlocked
(DBusConnection *connection);
static dbus_bool_t
_dbus_connection_peek_for_reply_unlocked
(DBusConnection *connection,
dbus_uint32_t client_serial);

```

```

static DBusMessageFilter *
_dbus_message_filter_ref (DBusMessageFilter *filter)
{
#ifdef DBUS_DISABLE_ASSERT
    _dbus_atomic_inc (&filter->refcount);
#else
    dbus_int32_t old_value;

    old_value = _dbus_atomic_inc (&filter->refcount);
    _dbus_assert (old_value > 0);
#endif

    return filter;
}

static void
_dbus_message_filter_unref (DBusMessageFilter *filter)
{
    dbus_int32_t old_value;

    old_value = _dbus_atomic_dec (&filter->refcount);
    _dbus_assert (old_value > 0);

    if (old_value == 1)
    {
        if (filter->free_user_data_function)
            (* filter->free_user_data_function) (filter->user_data);

        dbus_free (filter);
    }
}

/**
 * Acquires the connection lock.
 *
 * @param connection the connection.
 */
void
_dbus_connection_lock (DBusConnection *connection)
{
    CONNECTION_LOCK (connection);
}

/**
 * Releases the connection lock.
 *
 * @param connection the connection.
 */
void
_dbus_connection_unlock (DBusConnection *connection)
{

```

```

DBusList *expired_messages;
DBusList *iter;

if (TRACE_LOCKS)
{
    _dbus_verbose ("UNLOCK\n");
}

/* If we had messages that expired (fell off the incoming or
outgoing
 * queues) while we were locked, actually release them now */
expired_messages = connection->expired_messages;
connection->expired_messages = NULL;

RELEASING_LOCK_CHECK (connection);
_dbus_rmutex_unlock (connection->mutex);

for (iter = _dbus_list_pop_first_link (&expired_messages);
     iter != NULL;
     iter = _dbus_list_pop_first_link (&expired_messages))
{
    DBusMessage *message = iter->data;

    dbus_message_unref (message);
    _dbus_list_free_link (iter);
}
}

/**
 * Wakes up the main loop if it is sleeping
 * Needed if we're e.g. queueing outgoing messages
 * on a thread while the mainloop sleeps.
 *
 * @param connection the connection.
 */
static void
_dbus_connection_wakeup_mainloop (DBusConnection *connection)
{
    if (connection->wakeup_main_function)
        (*connection->wakeup_main_function) (connection->wakeup_main_data);
}

#ifdef DBUS_BUILD_TESTS
/**
 * Gets the locks so we can examine them
 *
 * @param connection the connection.
 * @param mutex_loc return for the location of the main mutex pointer
 * @param dispatch_mutex_loc return location of the dispatch mutex
 * pointer
 */

```

```

    * @param io_path_mutex_loc return location of the io_path mutex
    pointer
    * @param dispatch_cond_loc return location of the dispatch
    conditional
    *     variable pointer
    * @param io_path_cond_loc return location of the io_path conditional
    *     variable pointer
    */
void
_dbus_connection_test_get_locks (DBusConnection *connection,
                                DBusMutex      **mutex_loc,
                                DBusMutex      **dispatch_mutex_loc,
                                DBusMutex      **io_path_mutex_loc,
                                DBusCondVar    **dispatch_cond_loc,
                                DBusCondVar    **io_path_cond_loc)
{
    *mutex_loc = (DBusMutex *) connection->mutex;
    *dispatch_mutex_loc = (DBusMutex *) connection->dispatch_mutex;
    *io_path_mutex_loc = (DBusMutex *) connection->io_path_mutex;
    *dispatch_cond_loc = connection->dispatch_cond;
    *io_path_cond_loc = connection->io_path_cond;
}
#endif

/**
 * Adds a message-containing list link to the incoming message queue,
 * taking ownership of the link and the message's current refcount.
 * Cannot fail due to lack of memory.
 *
 * @param connection the connection.
 * @param link the message link to queue.
 */
void
_dbus_connection_queue_received_message_link (DBusConnection
*connection,
                                              DBusList      *link)
{
    DBusPendingCall *pending;
    dbus_uint32_t reply_serial;
    DBusMessage *message;

    _dbus_assert (_dbus_transport_get_is_authenticated (connection-
>transport));

    _dbus_list_append_link (&connection->incoming_messages,
                            link);
    message = link->data;

    /* If this is a reply we're waiting on, remove timeout for it */
    reply_serial = dbus_message_get_reply_serial (message);
    if (reply_serial != 0)
    {

```

```

    pending = _dbus_hash_table_lookup_int (connection-
>pending_replies,
                                           reply_serial);
    if (pending != NULL)
    {
        if (_dbus_pending_call_is_timeout_added_unlocked (pending))
            _dbus_connection_remove_timeout_unlocked (connection,
_dbus_pending_call_get_timeout_unlocked (pending));
        _dbus_pending_call_set_timeout_added_unlocked (pending, FALSE);
    }
}

connection->n_incoming += 1;

_dbus_connection_wakeup_mainloop (connection);

_dbus_verbose ("Message %p (%s %s %s %s '%s' reply to %u) added to
incoming queue %p, %d incoming\n",
               message,
               dbus_message_type_to_string (dbus_message_get_type
(message)),
               dbus_message_get_path (message) ?
               dbus_message_get_path (message) :
               "no path",
               dbus_message_get_interface (message) ?
               dbus_message_get_interface (message) :
               "no interface",
               dbus_message_get_member (message) ?
               dbus_message_get_member (message) :
               "no member",
               dbus_message_get_signature (message),
               dbus_message_get_reply_serial (message),
               connection,
               connection->n_incoming);

_dbus_message_trace_ref (message, -1, -1,
                        "_dbus_connection_queue_received_message_link");
}

/**
 * Adds a link + message to the incoming message queue.
 * Can't fail. Takes ownership of both link and message.
 *
 * @param connection the connection.
 * @param link the list node and message to queue.
 */
void

```

```

_dbus_connection_queue_synthesized_message_link (DBusConnection
*connection,
                                         DBusList *link)
{
    HAVE_LOCK_CHECK (connection);

    _dbus_list_append_link (&connection->incoming_messages, link);

    connection->n_incoming += 1;

    _dbus_connection_wakeup_mainloop (connection);

    _dbus_message_trace_ref (link->data, -1, -1,
        "_dbus_connection_queue_synthesized_message_link");

    _dbus_verbose ("Synthesized message %p added to incoming queue %p,
%d incoming\n",
        link->data, connection, connection->n_incoming);
}

/**
 * Checks whether there are messages in the outgoing message queue.
 * Called with connection lock held.
 *
 * @param connection the connection.
 * @returns #TRUE if the outgoing queue is non-empty.
 */
dbus_bool_t
_dbus_connection_has_messages_to_send_unlocked (DBusConnection
*connection)
{
    HAVE_LOCK_CHECK (connection);
    return connection->outgoing_messages != NULL;
}

/**
 * Checks whether there are messages in the outgoing message queue.
 * Use dbus_connection_flush() to block until all outgoing
 * messages have been written to the underlying transport
 * (such as a socket).
 *
 * @param connection the connection.
 * @returns #TRUE if the outgoing queue is non-empty.
 */
dbus_bool_t
dbus_connection_has_messages_to_send (DBusConnection *connection)
{
    dbus_bool_t v;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
}

```

```

CONNECTION_LOCK (connection);
v = _dbus_connection_has_messages_to_send_unlocked (connection);
CONNECTION_UNLOCK (connection);

return v;
}

/**
 * Gets the next outgoing message. The message remains in the
 * queue, and the caller does not own a reference to it.
 *
 * @param connection the connection.
 * @returns the message to be sent.
 */
DBusMessage*
_dbus_connection_get_message_to_send (DBusConnection *connection)
{
    HAVE_LOCK_CHECK (connection);

    return _dbus_list_get_last (&connection->outgoing_messages);
}

/**
 * Notifies the connection that a message has been sent, so the
 * message can be removed from the outgoing queue.
 * Called with the connection lock held.
 *
 * @param connection the connection.
 * @param message the message that was sent.
 */
void
_dbus_connection_message_sent_unlocked (DBusConnection *connection,
                                       DBusMessage *message)
{
    DBusList *link;

    HAVE_LOCK_CHECK (connection);

    /* This can be called before we even complete authentication, since
     * it's called on disconnect to clean up the outgoing queue.
     * It's also called as we successfully send each message.
     */

    link = _dbus_list_get_last_link (&connection->outgoing_messages);
    _dbus_assert (link != NULL);
    _dbus_assert (link->data == message);

    _dbus_list_unlink (&connection->outgoing_messages,
                      link);
    _dbus_list_prepend_link (&connection->expired_messages, link);

    connection->n_outgoing -= 1;
}

```



```

    _dbus_verbose ("Message %p (%s %s %s %s '%s') removed from outgoing
queue %p, %d left to send\n",
                message,
                dbus_message_type_to_string (dbus_message_get_type
(message)),
                dbus_message_get_path (message) ?
dbus_message_get_path (message) :
                "no path",
                dbus_message_get_interface (message) ?
dbus_message_get_interface (message) :
                "no interface",
                dbus_message_get_member (message) ?
dbus_message_get_member (message) :
                "no member",
                dbus_message_get_signature (message),
                connection, connection->n_outgoing);

    /* It's OK that in principle we call the notify function, because
for the
    * outgoing limit, there isn't one */
    _dbus_message_remove_counter (message, connection-
>outgoing_counter);

    /* The message will actually be unreffed when we unlock */
}

/** Function to be called in protected_change_watch() with refcount
held */
typedef dbus_bool_t (* DBusWatchAddFunction)      (DBusWatchList *list,
                                                    DBusWatch
*watch);
/** Function to be called in protected_change_watch() with refcount
held */
typedef void        (* DBusWatchRemoveFunction)   (DBusWatchList *list,
                                                    DBusWatch
*watch);
/** Function to be called in protected_change_watch() with refcount
held */
typedef void        (* DBusWatchToggleFunction)  (DBusWatchList *list,
                                                    DBusWatch
*watch,
                                                    dbus_bool_t
enabled);

static dbus_bool_t
protected_change_watch (DBusConnection      *connection,
                       DBusWatch          *watch,
                       DBusWatchAddFunction add_function,
                       DBusWatchRemoveFunction remove_function,
                       DBusWatchToggleFunction toggle_function,
                       dbus_bool_t        enabled)

```

```

{
    dbus_bool_t retval;

    HAVE_LOCK_CHECK (connection);

    /* The original purpose of protected_change_watch() was to hold a
     * ref on the connection while dropping the connection lock, then
     * calling out to the app. This was a broken hack that did not
     * work, since the connection was in a hosed state (no WatchList
     * field) while calling out.
     *
     * So for now we'll just keep the lock while calling out. This means
     * apps are not allowed to call DBusConnection methods inside a
     * watch function or they will deadlock.
     *
     * The "real fix" is to use the _and_unlock() pattern found
     * elsewhere in the code, to defer calling out to the app until
     * we're about to drop locks and return flow of control to the app
     * anyway.
     *
     * See http://lists.freedesktop.org/archives/dbus/2007-
    July/thread.html#8144
     */

    if (connection->watches)
    {
        if (add_function)
            retval = (* add_function) (connection->watches, watch);
        else if (remove_function)
        {
            retval = TRUE;
            (* remove_function) (connection->watches, watch);
        }
        else
        {
            retval = TRUE;
            (* toggle_function) (connection->watches, watch, enabled);
        }
        return retval;
    }
    else
        return FALSE;
}

/**
 * Adds a watch using the connection's DBusAddWatchFunction if
 * available. Otherwise records the watch to be added when said
 * function is available. Also re-adds the watch if the
 * DBusAddWatchFunction changes. May fail due to lack of memory.
 * Connection lock should be held when calling this.
 */

```



```

        enabled);
}

/** Function to be called in protected_change_timeout() with refcount
held */
typedef dbus_bool_t (* DBusTimeoutAddFunction) (DBusTimeoutList
*list,
                                                DBusTimeout
*timeout);
/** Function to be called in protected_change_timeout() with refcount
held */
typedef void (* DBusTimeoutRemoveFunction) (DBusTimeoutList
*list,
                                                DBusTimeout
*timeout);
/** Function to be called in protected_change_timeout() with refcount
held */
typedef void (* DBusTimeoutToggleFunction) (DBusTimeoutList
*list,
                                                DBusTimeout
*timeout,
                                                dbus_bool_t
enabled);

static dbus_bool_t
protected_change_timeout (DBusConnection *connection,
                          DBusTimeout *timeout,
                          DBusTimeoutAddFunction add_function,
                          DBusTimeoutRemoveFunction remove_function,
                          DBusTimeoutToggleFunction toggle_function,
                          dbus_bool_t enabled)
{
    dbus_bool_t retval;

    HAVE_LOCK_CHECK (connection);

    /* The original purpose of protected_change_timeout() was to hold a
    * ref on the connection while dropping the connection lock, then
    * calling out to the app. This was a broken hack that did not
    * work, since the connection was in a hosed state (no TimeoutList
    * field) while calling out.
    *
    * So for now we'll just keep the lock while calling out. This means
    * apps are not allowed to call DBusConnection methods inside a
    * timeout function or they will deadlock.
    *
    * The "real fix" is to use the _and_unlock() pattern found
    * elsewhere in the code, to defer calling out to the app until
    * we're about to drop locks and return flow of control to the app
    * anyway.
    *
    */
}

```

```

    * See http://lists.freedesktop.org/archives/dbus/2007-
    July/thread.html#8144
    */

    if (connection->timeouts)
    {
        if (add_function)
            retval = (* add_function) (connection->timeouts, timeout);
        else if (remove_function)
        {
            retval = TRUE;
            (* remove_function) (connection->timeouts, timeout);
        }
        else
        {
            retval = TRUE;
            (* toggle_function) (connection->timeouts, timeout,
enabled);
        }
        return retval;
    }
    else
        return FALSE;
}

/**
 * Adds a timeout using the connection's DBusAddTimeoutFunction if
 * available. Otherwise records the timeout to be added when said
 * function is available. Also re-adds the timeout if the
 * DBusAddTimeoutFunction changes. May fail due to lack of memory.
 * The timeout will fire repeatedly until removed.
 * Connection lock should be held when calling this.
 *
 * @param connection the connection.
 * @param timeout the timeout to add.
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_connection_add_timeout_unlocked (DBusConnection *connection,
                                      DBusTimeout      *timeout)
{
    return protected_change_timeout (connection, timeout,
                                     _dbus_timeout_list_add_timeout,
                                     NULL, NULL, FALSE);
}

/**
 * Removes a timeout using the connection's DBusRemoveTimeoutFunction
 * if available. It's an error to call this function on a timeout
 * that was not previously added.
 * Connection lock should be held when calling this.
 */

```

```

* @param connection the connection.
* @param timeout the timeout to remove.
*/
void
_dbus_connection_remove_timeout_unlocked (DBusConnection *connection,
                                         DBusTimeout      *timeout)
{
    protected_change_timeout (connection, timeout,
                              NULL,
                              _dbus_timeout_list_remove_timeout,
                              NULL, FALSE);
}

/**
* Toggles a timeout and notifies app via connection's
* DBusTimeoutToggledFunction if available. It's an error to call this
* function on a timeout that was not previously added.
* Connection lock should be held when calling this.
*
* @param connection the connection.
* @param timeout the timeout to toggle.
* @param enabled whether to enable or disable
*/
void
_dbus_connection_toggle_timeout_unlocked (DBusConnection
*connection,
                                         DBusTimeout      *timeout,
                                         dbus_bool_t      enabled)
{
    protected_change_timeout (connection, timeout,
                              NULL, NULL,
                              _dbus_timeout_list_toggle_timeout,
                              enabled);
}

static dbus_bool_t
_dbus_connection_attach_pending_call_unlocked (DBusConnection
*connection,
                                               DBusPendingCall
*pending)
{
    dbus_uint32_t reply_serial;
    DBusTimeout *timeout;

    HAVE_LOCK_CHECK (connection);

    reply_serial = _dbus_pending_call_get_reply_serial_unlocked
(pending);

    _dbus_assert (reply_serial != 0);

    timeout = _dbus_pending_call_get_timeout_unlocked (pending);

```

```

    if (timeout)
    {
        if (!_dbus_connection_add_timeout_unlocked (connection,
timeout))
            return FALSE;

        if (!_dbus_hash_table_insert_int (connection->pending_replies,
reply_serial,
pending))
        {
            _dbus_connection_remove_timeout_unlocked (connection,
timeout);

            _dbus_pending_call_set_timeout_added_unlocked (pending,
FALSE);
            HAVE_LOCK_CHECK (connection);
            return FALSE;
        }

        _dbus_pending_call_set_timeout_added_unlocked (pending, TRUE);
    }
    else
    {
        if (!_dbus_hash_table_insert_int (connection->pending_replies,
reply_serial,
pending))
        {
            HAVE_LOCK_CHECK (connection);
            return FALSE;
        }
    }

    _dbus_pending_call_ref_unlocked (pending);

    HAVE_LOCK_CHECK (connection);

    return TRUE;
}

static void
free_pending_call_on_hash_removal (void *data)
{
    DBusPendingCall *pending;
    DBusConnection *connection;

    if (data == NULL)
        return;

    pending = data;

    connection = _dbus_pending_call_get_connection_unlocked (pending);

```

```

HAVE_LOCK_CHECK (connection);

if (_dbus_pending_call_is_timeout_added_unlocked (pending))
{
    _dbus_connection_remove_timeout_unlocked (connection,
_dbus_pending_call_get_timeout_unlocked (pending));

    _dbus_pending_call_set_timeout_added_unlocked (pending, FALSE);
}

/* FIXME 1.0? this is sort of dangerous and undesirable to drop the
lock
* here, but the pending call finalizer could in principle call out
to
* application code so we pretty much have to... some larger code
reorg
* might be needed.
*/
_dbus_connection_ref_unlocked (connection);
_dbus_pending_call_unref_and_unlock (pending);
CONNECTION_LOCK (connection);
_dbus_connection_unref_unlocked (connection);
}

static void
_dbus_connection_detach_pending_call_unlocked (DBusConnection
*connection,
                                             DBusPendingCall
*pending)
{
    /* This ends up unlocking to call the pending call finalizer, which
is unexpected to
* say the least.
*/
    _dbus_hash_table_remove_int (connection->pending_replies,
_dbus_pending_call_get_reply_serial_unlocked (pending));
}

static void
_dbus_connection_detach_pending_call_and_unlock (DBusConnection
*connection,
                                             DBusPendingCall
*pending)
{
    /* The idea here is to avoid finalizing the pending call
* with the lock held, since there's a destroy notifier
* in pending call that goes out to application code.
*
* There's an extra unlock inside the hash table

```



```

    * "free pending call" function FIXME...
    */
    _dbus_pending_call_ref_unlocked (pending);
    _dbus_hash_table_remove_int (connection->pending_replies,
_dbus_pending_call_get_reply_serial_unlocked (pending));

    if (_dbus_pending_call_is_timeout_added_unlocked (pending))
        _dbus_connection_remove_timeout_unlocked (connection,
            _dbus_pending_call_get_timeout_unlocked (pending));

    _dbus_pending_call_set_timeout_added_unlocked (pending, FALSE);

    _dbus_pending_call_unref_and_unlock (pending);
}

/**
 * Removes a pending call from the connection, such that
 * the pending reply will be ignored. May drop the last
 * reference to the pending call.
 *
 * @param connection the connection
 * @param pending the pending call
 */
void
_dbus_connection_remove_pending_call (DBusConnection *connection,
                                     DBusPendingCall *pending)
{
    CONNECTION_LOCK (connection);
    _dbus_connection_detach_pending_call_and_unlock (connection,
pending);
}

/**
 * Acquire the transporter I/O path. This must be done before
 * doing any I/O in the transporter. May sleep and drop the
 * IO path mutex while waiting for the I/O path.
 *
 * @param connection the connection.
 * @param timeout_milliseconds maximum blocking time, or -1 for no
limit.
 * @returns TRUE if the I/O path was acquired.
 */
static dbus_bool_t
_dbus_connection_acquire_io_path (DBusConnection *connection,
                                 int
                                 timeout_milliseconds)
{
    dbus_bool_t we_acquired;

    HAVE_LOCK_CHECK (connection);

    /* We don't want the connection to vanish */

```

```

    _dbus_connection_ref_unlocked (connection);

    /* We will only touch io_path_acquired which is protected by our
    mutex */
    CONNECTION_UNLOCK (connection);

    _dbus_verbose ("locking io_path_mutex\n");
    _dbus_cmutex_lock (connection->io_path_mutex);

    _dbus_verbose ("start connection->io_path_acquired = %d timeout =
    %d\n",
                  connection->io_path_acquired, timeout_milliseconds);

    we_acquired = FALSE;

    if (connection->io_path_acquired)
    {
        if (timeout_milliseconds != -1)
        {
            _dbus_verbose ("waiting %d for IO path to be acquirable\n",
                          timeout_milliseconds);

            if (!_dbus_condvar_wait_timeout (connection->io_path_cond,
                                           connection->io_path_mutex,
                                           timeout_milliseconds))
            {
                /* We timed out before anyone signaled. */
                /* (writing the loop to handle the !timedout case by
                * waiting longer if needed is a pain since dbus
                * wraps pthread_cond_timedwait to take a relative
                * time instead of absolute, something kind of stupid
                * on our part. for now it doesn't matter, we will just
                * end up back here eventually.)
                */
            }
        }
        else
        {
            while (connection->io_path_acquired)
            {
                _dbus_verbose ("waiting for IO path to be
                acquirable\n");
                _dbus_condvar_wait (connection->io_path_cond,
                                   connection->io_path_mutex);
            }
        }
    }

    if (!connection->io_path_acquired)
    {
        we_acquired = TRUE;
        connection->io_path_acquired = TRUE;
    }

```

```

    }

    _dbus_verbose ("end connection->io_path_acquired = %d we_acquired =
%d\n",
                  connection->io_path_acquired, we_acquired);

    _dbus_verbose ("unlocking io_path_mutex\n");
    _dbus_cmutex_unlock (connection->io_path_mutex);

    CONNECTION_LOCK (connection);

    HAVE_LOCK_CHECK (connection);

    _dbus_connection_unref_unlocked (connection);

    return we_acquired;
}

/**
 * Release the I/O path when you're done with it. Only call
 * after you've acquired the I/O. Wakes up at most one thread
 * currently waiting to acquire the I/O path.
 *
 * @param connection the connection.
 */
static void
_dbus_connection_release_io_path (DBusConnection *connection)
{
    HAVE_LOCK_CHECK (connection);

    _dbus_verbose ("locking io_path_mutex\n");
    _dbus_cmutex_lock (connection->io_path_mutex);

    _dbus_assert (connection->io_path_acquired);

    _dbus_verbose ("start connection->io_path_acquired = %d\n",
                  connection->io_path_acquired);

    connection->io_path_acquired = FALSE;
    _dbus_condvar_wake_one (connection->io_path_cond);

    _dbus_verbose ("unlocking io_path_mutex\n");
    _dbus_cmutex_unlock (connection->io_path_mutex);
}

/**
 * Queues incoming messages and sends outgoing messages for this
 * connection, optionally blocking in the process. Each call to
 * _dbus_connection_do_iteration_unlocked() will call select() or
 * poll() one
 * time and then read or write data if possible.
 */

```

```

* The purpose of this function is to be able to flush outgoing
* messages or queue up incoming messages without returning
* control to the application and causing reentrancy weirdness.
*
* The flags parameter allows you to specify whether to
* read incoming messages, write outgoing messages, or both,
* and whether to block if no immediate action is possible.
*
* The timeout_milliseconds parameter does nothing unless the
* iteration is blocking.
*
* If there are no outgoing messages and DBUS_ITERATION_DO_READING
* wasn't specified, then it's impossible to block, even if
* you specify DBUS_ITERATION_BLOCK; in that case the function
* returns immediately.
*
* If pending is not NULL then a check is made if the pending call
* is completed after the io path has been required. If the call
* has been completed nothing is done. This must be done since
* the _dbus_connection_acquire_io_path releases the connection
* lock for a while.
*
* Called with connection lock held.
*
* @param connection the connection.
* @param pending the pending call that should be checked or NULL
* @param flags iteration flags.
* @param timeout_milliseconds maximum blocking time, or -1 for no
limit.
*/
void
_dbus_connection_do_iteration_unlocked (DBusConnection *connection,
                                       DBusPendingCall *pending,
                                       unsigned int flags,
                                       int
timeout_milliseconds)
{
    _dbus_verbose ("start\n");

    HAVE_LOCK_CHECK (connection);

    if (connection->n_outgoing == 0)
        flags &= ~DBUS_ITERATION_DO_WRITING;

    if (_dbus_connection_acquire_io_path (connection,
                                         (flags & DBUS_ITERATION_BLOCK) ?
timeout_milliseconds : 0))
    {
        HAVE_LOCK_CHECK (connection);

        if ( (pending != NULL) &&
_dbus_pending_call_get_completed_unlocked(pending))

```

```

        {
            _dbus_verbose ("pending call completed while acquiring I/O
path");
        }
        else if ( (pending != NULL) &&
                 _dbus_connection_peek_for_reply_unlocked (connection,
_dbus_pending_call_get_reply_serial_unlocked (pending)))
        {
            _dbus_verbose ("pending call completed while acquiring I/O
path (reply found in queue)");
        }
        else
        {
            _dbus_transport_do_iteration (connection->transport,
                                         flags, timeout_milliseconds);
        }

        _dbus_connection_release_io_path (connection);
    }

    HAVE_LOCK_CHECK (connection);

    _dbus_verbose ("end\n");
}

/**
 * Creates a new connection for the given transport.  A transport
 * represents a message stream that uses some concrete mechanism, such
 * as UNIX domain sockets.  May return #NULL if insufficient
 * memory exists to create the connection.
 *
 * @param transport the transport.
 * @returns the new connection, or #NULL on failure.
 */
DBusConnection*
_dbus_connection_new_for_transport (DBusTransport *transport)
{
    DBusConnection *connection;
    DBusWatchList *watch_list;
    DBusTimeoutList *timeout_list;
    DBusHashTable *pending_replies;
    DBusList *disconnect_link;
    DBusMessage *disconnect_message;
    DBusCounter *outgoing_counter;
    DBusObjectTree *objects;

    watch_list = NULL;
    connection = NULL;
    pending_replies = NULL;
    timeout_list = NULL;
    disconnect_link = NULL;

```



```

if (disconnect_message == NULL)
    goto error;

disconnect_link = _dbus_list_alloc_link (disconnect_message);
if (disconnect_link == NULL)
    goto error;

outgoing_counter = _dbus_counter_new ();
if (outgoing_counter == NULL)
    goto error;

objects = _dbus_object_tree_new (connection);
if (objects == NULL)
    goto error;

if (_dbus_modify_sigpipe)
    _dbus_disable_sigpipe ();

/* initialized to 0: use atomic op to avoid mixing atomic and non-
atomic */
_dbus_atomic_inc (&connection->refcount);
connection->transport = transport;
connection->watches = watch_list;
connection->timeouts = timeout_list;
connection->pending_replies = pending_replies;
connection->outgoing_counter = outgoing_counter;
connection->filter_list = NULL;
connection->last_dispatch_status = DBUS_DISPATCH_COMPLETE; /* so
we're notified first time there's data */
connection->objects = objects;
connection->exit_on_disconnect = FALSE;
connection->shareable = FALSE;
connection->route_peer_messages = FALSE;
connection->disconnected_message_arrived = FALSE;
connection->disconnected_message_processed = FALSE;

#ifdef DBUS_DISABLE_CHECKS
    connection->generation = _dbus_current_generation;
#endif

_dbus_data_slot_list_init (&connection->slot_list);

connection->client_serial = 1;

connection->disconnect_message_link = disconnect_link;

CONNECTION_LOCK (connection);

if (!_dbus_transport_set_connection (transport, connection))
{
    CONNECTION_UNLOCK (connection);

```

```

        goto error;
    }

    _dbus_transport_ref (transport);

    CONNECTION_UNLOCK (connection);

    _dbus_connection_trace_ref (connection, 0, 1, "new_for_transport");
    return connection;

error:
    if (disconnect_message != NULL)
        dbus_message_unref (disconnect_message);

    if (disconnect_link != NULL)
        _dbus_list_free_link (disconnect_link);

    if (connection != NULL)
    {
        _dbus_condvar_free_at_location (&connection->io_path_cond);
        _dbus_condvar_free_at_location (&connection->dispatch_cond);
        _dbus_rmutex_free_at_location (&connection->mutex);
        _dbus_cmutex_free_at_location (&connection->io_path_mutex);
        _dbus_cmutex_free_at_location (&connection->dispatch_mutex);
        _dbus_rmutex_free_at_location (&connection->slot_mutex);
        dbus_free (connection);
    }
    if (pending_replies)
        _dbus_hash_table_unref (pending_replies);

    if (watch_list)
        _dbus_watch_list_free (watch_list);

    if (timeout_list)
        _dbus_timeout_list_free (timeout_list);

    if (outgoing_counter)
        _dbus_counter_unref (outgoing_counter);

    if (objects)
        _dbus_object_tree_unref (objects);

    return NULL;
}

/**
 * Increments the reference count of a DbusConnection.
 * Requires that the caller already holds the connection lock.
 *
 * @param connection the connection.
 * @returns the connection.
 */

```



```

DBusConnection *
_dbus_connection_ref_unlocked (DBusConnection *connection)
{
    dbus_int32_t old_refcount;

    _dbus_assert (connection != NULL);
    _dbus_assert (connection->generation == _dbus_current_generation);

    HAVE_LOCK_CHECK (connection);

    old_refcount = _dbus_atomic_inc (&connection->refcount);
    _dbus_connection_trace_ref (connection, old_refcount, old_refcount +
1,
        "ref_unlocked");

    return connection;
}

/**
 * Decrements the reference count of a DBusConnection.
 * Requires that the caller already holds the connection lock.
 *
 * @param connection the connection.
 */
void
_dbus_connection_unref_unlocked (DBusConnection *connection)
{
    dbus_int32_t old_refcount;

    HAVE_LOCK_CHECK (connection);

    _dbus_assert (connection != NULL);

    old_refcount = _dbus_atomic_dec (&connection->refcount);

    _dbus_connection_trace_ref (connection, old_refcount, old_refcount -
1,
        "unref_unlocked");

    if (old_refcount == 1)
        _dbus_connection_last_unref (connection);
}

static dbus_uint32_t
_dbus_connection_get_next_client_serial (DBusConnection *connection)
{
    dbus_uint32_t serial;

    serial = connection->client_serial++;

    if (connection->client_serial == 0)
        connection->client_serial = 1;
}

```

```

    return serial;
}

/**
 * A callback for use with dbus_watch_new() to create a DBusWatch.
 *
 * @todo This is basically a hack - we could delete
_dbus_transport_handle_watch()
* and the virtual handle_watch in DBusTransport if we got rid of it.
* The reason this is some work is threading, see the
_dbus_connection_handle_watch()
* implementation.
 *
 * @param watch the watch.
 * @param condition the current condition of the file descriptors
being watched.
 * @param data must be a pointer to a #DBusConnection
 * @returns #FALSE if the IO condition may not have been fully handled
due to lack of memory
 */
dbus_bool_t
_dbus_connection_handle_watch (DBusWatch                *watch,
                               unsigned int             condition,
                               void                    *data)
{
    DBusConnection *connection;
    dbus_bool_t retval;
    DBusDispatchStatus status;

    connection = data;

    _dbus_verbose ("start\n");

    CONNECTION_LOCK (connection);

    if (!_dbus_connection_acquire_io_path (connection, 1))
    {
        /* another thread is handling the message */
        CONNECTION_UNLOCK (connection);
        return TRUE;
    }

    HAVE_LOCK_CHECK (connection);
    retval = _dbus_transport_handle_watch (connection->transport,
                                          watch, condition);

    _dbus_connection_release_io_path (connection);

    HAVE_LOCK_CHECK (connection);

    _dbus_verbose ("middle\n");

```

```

    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* this calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    _dbus_verbose ("end\n");

    return retval;
}

_DBUS_DEFINE_GLOBAL_LOCK (shared_connections);
static DBusHashTable *shared_connections = NULL;
static DBusList *shared_connections_no_guid = NULL;

static void
close_connection_on_shutdown (DBusConnection *connection)
{
    DBusMessage *message;

    dbus_connection_ref (connection);
    _dbus_connection_close_possibly_shared (connection);

    /* Churn through to the Disconnected message */
    while ((message = dbus_connection_pop_message (connection)))
    {
        dbus_message_unref (message);
    }
    dbus_connection_unref (connection);
}

static void
shared_connections_shutdown (void *data)
{
    int n_entries;

    _DBUS_LOCK (shared_connections);

    /* This is a little bit unpleasant... better ideas? */
    while ((n_entries = _dbus_hash_table_get_n_entries
(shared_connections)) > 0)
    {
        DBusConnection *connection;
        DBusHashIter iter;

        _dbus_hash_iter_init (shared_connections, &iter);
        _dbus_hash_iter_next (&iter);

        connection = _dbus_hash_iter_get_value (&iter);

        _DBUS_UNLOCK (shared_connections);

```

```

    close_connection_on_shutdown (connection);
    _DBUS_LOCK (shared_connections);

    /* The connection should now be dead and not in our hash ... */
    _dbus_assert (_dbus_hash_table_get_n_entries
(shared_connections) < n_entries);
    }

    _dbus_assert (_dbus_hash_table_get_n_entries (shared_connections) ==
0);

    _dbus_hash_table_unref (shared_connections);
    shared_connections = NULL;

    if (shared_connections_no_guid != NULL)
    {
        DBusConnection *connection;
        connection = _dbus_list_pop_first (&shared_connections_no_guid);
        while (connection != NULL)
        {
            _DBUS_UNLOCK (shared_connections);
            close_connection_on_shutdown (connection);
            _DBUS_LOCK (shared_connections);
            connection = _dbus_list_pop_first
(&shared_connections_no_guid);
        }
    }

    shared_connections_no_guid = NULL;

    _DBUS_UNLOCK (shared_connections);
}

static dbus_bool_t
connection_lookup_shared (DBusAddressEntry *entry,
                        DBusConnection **result)
{
    _dbus_verbose ("checking for existing connection\n");

    *result = NULL;

    _DBUS_LOCK (shared_connections);

    if (shared_connections == NULL)
    {
        _dbus_verbose ("creating shared_connections hash table\n");

        shared_connections = _dbus_hash_table_new (DBUS_HASH_STRING,
                                                    dbus_free,
                                                    NULL);

        if (shared_connections == NULL)
        {

```

```

        _DBUS_UNLOCK (shared_connections);
        return FALSE;
    }

    if (!_dbus_register_shutdown_func (shared_connections_shutdown,
NULL))
    {
        _dbus_hash_table_unref (shared_connections);
        shared_connections = NULL;
        _DBUS_UNLOCK (shared_connections);
        return FALSE;
    }

    _dbus_verbose (" successfully created shared_connections\n");

    _DBUS_UNLOCK (shared_connections);
    return TRUE; /* no point looking up in the hash we just made */
}
else
{
    const char *guid;

    guid = dbus_address_entry_get_value (entry, "guid");

    if (guid != NULL)
    {
        DBusConnection *connection;

        connection = _dbus_hash_table_lookup_string
(shared_connections,
                                                                    guid);

        if (connection)
        {
            /* The DBusConnection can't be finalized without taking
            * the shared_connections lock to remove it from the
            * hash. So it's safe to ref the connection here.
            * However, it may be disconnected if the Disconnected
            * message hasn't been processed yet, in which case we
            * want to pretend it isn't in the hash and avoid
            * returning it.
            *
            * The idea is to avoid ever returning a disconnected
connection
            * from dbus_connection_open(). We could just
synchronously
            * drop our shared ref to the connection on connection
disconnect,
            * and then assert here that the connection is
connected, but
            * that causes reentrancy headaches.
            */

```

```

        CONNECTION_LOCK (connection);
        if (_dbus_connection_get_is_connected_unlocked
(connection))
        {
            _dbus_connection_ref_unlocked (connection);
            *result = connection;
            _dbus_verbose ("looked up existing connection to
server guid %s\n",
                        guid);
        }
        else
        {
            _dbus_verbose ("looked up existing connection to
server guid %s but it was disconnected so ignoring it\n",
                        guid);
        }
        CONNECTION_UNLOCK (connection);
    }
}

_DBUS_UNLOCK (shared_connections);
return TRUE;
}

static dbus_bool_t
connection_record_shared_unlocked (DBusConnection *connection,
                                const char *guid)
{
    char *guid_key;
    char *guid_in_connection;

    HAVE_LOCK_CHECK (connection);
    _dbus_assert (connection->server_guid == NULL);
    _dbus_assert (connection->shareable);

    /* get a hard ref on this connection, even if
     * we won't in fact store it in the hash, we still
     * need to hold a ref on it until it's disconnected.
     */
    _dbus_connection_ref_unlocked (connection);

    if (guid == NULL)
    {
        _DBUS_LOCK (shared_connections);

        if (!_dbus_list_prepend (&shared_connections_no_guid,
connection))
        {
            _DBUS_UNLOCK (shared_connections);
            return FALSE;
        }
    }
}

```

```

        _DBUS_UNLOCK (shared_connections);
        return TRUE; /* don't store in the hash */
    }

/* A separate copy of the key is required in the hash table, because
 * we don't have a lock on the connection when we are doing a hash
 * lookup.
 */

guid_key = _dbus_strdup (guid);
if (guid_key == NULL)
    return FALSE;

guid_in_connection = _dbus_strdup (guid);
if (guid_in_connection == NULL)
{
    dbus_free (guid_key);
    return FALSE;
}

_DBUS_LOCK (shared_connections);
_dbus_assert (shared_connections != NULL);

if (!_dbus_hash_table_insert_string (shared_connections,
                                    guid_key, connection))
{
    dbus_free (guid_key);
    dbus_free (guid_in_connection);
    _DBUS_UNLOCK (shared_connections);
    return FALSE;
}

connection->server_guid = guid_in_connection;

_dbus_verbose ("stored connection to %s to be shared\n",
              connection->server_guid);

_DBUS_UNLOCK (shared_connections);

_dbus_assert (connection->server_guid != NULL);

return TRUE;
}

static void
connection_forget_shared_unlocked (DBusConnection *connection)
{
    HAVE_LOCK_CHECK (connection);

    if (!connection->shareable)
        return;
}

```

```

    _DBUS_LOCK (shared_connections);

    if (connection->server_guid != NULL)
    {
        _dbus_verbose ("dropping connection to %s out of the shared
table\n",
                      connection->server_guid);

        if (!_dbus_hash_table_remove_string (shared_connections,
                                             connection->server_guid))
            _dbus_assert_not_reached ("connection was not in the shared
table");

        dbus_free (connection->server_guid);
        connection->server_guid = NULL;
    }
    else
    {
        _dbus_list_remove (&shared_connections_no_guid, connection);
    }

    _DBUS_UNLOCK (shared_connections);

    /* remove our reference held on all shareable connections */
    _dbus_connection_unref_unlocked (connection);
}

static DBusConnection*
connection_try_from_address_entry (DBusAddressEntry *entry,
                                   DBusError         *error)
{
    DBusTransport *transport;
    DBusConnection *connection;

    transport = _dbus_transport_open (entry, error);

    if (transport == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        return NULL;
    }

    connection = _dbus_connection_new_for_transport (transport);

    _dbus_transport_unref (transport);

    if (connection == NULL)
    {
        _DBUS_SET_OOM (error);
        return NULL;
    }
}

```



```

#ifdef DBUS_DISABLE_CHECKS
    _dbus_assert (!connection->have_connection_lock);
#endif
    return connection;
}

/*
 * If the shared parameter is true, then any existing connection will
 * be used (and if a new connection is created, it will be available
 * for use by others). If the shared parameter is false, a new
 * connection will always be created, and the new connection will
 * never be returned to other callers.
 *
 * @param address the address
 * @param shared whether the connection is shared or private
 * @param error error return
 * @returns the connection or #NULL on error
 */
static DBusConnection*
_dbus_connection_open_internal (const char      *address,
                               dbus_bool_t     shared,
                               DBusError       *error)
{
    DBusConnection *connection;
    DBusAddressEntry **entries;
    DBusError tmp_error = DBUS_ERROR_INIT;
    DBusError first_error = DBUS_ERROR_INIT;
    int len, i;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_verbose ("opening %s connection to: %s\n",
                  shared ? "shared" : "private", address);

    if (!dbus_parse_address (address, &entries, &len, error))
        return NULL;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    connection = NULL;

    for (i = 0; i < len; i++)
    {
        if (shared)
        {
            if (!connection_lookup_shared (entries[i], &connection))
                _DBUS_SET_OOM (&tmp_error);
        }

        if (connection == NULL)
        {

```

```

        connection = connection_try_from_address_entry (entries[i],
                                                         &tmp_error);

        if (connection != NULL && shared)
        {
            const char *guid;

            connection->shareable = TRUE;

            /* guid may be NULL */
            guid = dbus_address_entry_get_value (entries[i],
"guid");

            CONNECTION_LOCK (connection);

            if (!connection_record_shared_unlocked (connection,
guid))
            {
                _DBUS_SET_OOM (&tmp_error);
                _dbus_connection_close_possibly_shared_and_unlock
(connection);
                dbus_connection_unref (connection);
                connection = NULL;
            }
            else
                CONNECTION_UNLOCK (connection);
        }

        if (connection)
            break;

        _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);

        if (i == 0)
            dbus_move_error (&tmp_error, &first_error);
        else
            dbus_error_free (&tmp_error);
    }

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    _DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);

    if (connection == NULL)
    {
        _DBUS_ASSERT_ERROR_IS_SET (&first_error);
        dbus_move_error (&first_error, error);
    }
    else
        dbus_error_free (&first_error);

    dbus_address_entries_free (entries);

```

```

    return connection;
}

/**
 * Closes a shared OR private connection, while
dbus_connection_close() can
 * only be used on private connections. Should only be called by the
 * dbus code that owns the connection - an owner must be known,
 * the open/close state is like malloc/free, not like ref/unref.
 *
 * @param connection the connection
 */
void
dbus_connection_close_possibly_shared (DBusConnection *connection)
{
    _dbus_assert (connection != NULL);
    _dbus_assert (connection->generation == _dbus_current_generation);

    CONNECTION_LOCK (connection);
    _dbus_connection_close_possibly_shared_and_unlock (connection);
}

static DBusPreallocatedSend*
dbus_connection_preallocate_send_unlocked (DBusConnection
*connection)
{
    DBusPreallocatedSend *preallocated;

    HAVE_LOCK_CHECK (connection);

    _dbus_assert (connection != NULL);

    preallocated = dbus_new (DBusPreallocatedSend, 1);
    if (preallocated == NULL)
        return NULL;

    preallocated->queue_link = _dbus_list_alloc_link (NULL);
    if (preallocated->queue_link == NULL)
        goto failed_0;

    preallocated->counter_link = _dbus_list_alloc_link (connection-
>outgoing_counter);
    if (preallocated->counter_link == NULL)
        goto failed_1;

    _dbus_counter_ref (preallocated->counter_link->data);

    preallocated->connection = connection;

    return preallocated;

failed_1:

```

```

    _dbus_list_free_link (preallocated->queue_link);
failed_0:
    dbus_free (preallocated);

    return NULL;
}

/* Called with lock held, does not update dispatch status */
static void
_dbus_connection_send_preallocated_unlocked_no_update (DBusConnection
*connection,

DBusPreallocatedSend *preallocated,

DBusMessage
*message,

dbus_uint32_t
*client_serial)
{
    dbus_uint32_t serial;

    preallocated->queue_link->data = message;
    _dbus_list_prepend_link (&connection->outgoing_messages,
        preallocated->queue_link);

    /* It's OK that we'll never call the notify function, because for
the
    * outgoing limit, there isn't one */
    _dbus_message_add_counter_link (message,
        preallocated->counter_link);

    dbus_free (preallocated);
    preallocated = NULL;

    dbus_message_ref (message);

    connection->n_outgoing += 1;

    _dbus_verbose ("Message %p (%s %s %s %s '%s') for %s added to
outgoing queue %p, %d pending to send\n",
        message,
        dbus_message_type_to_string (dbus_message_get_type
(message)),
        dbus_message_get_path (message) ?
        dbus_message_get_path (message) :
        "no path",
        dbus_message_get_interface (message) ?
        dbus_message_get_interface (message) :
        "no interface",
        dbus_message_get_member (message) ?
        dbus_message_get_member (message) :
        "no member",
        dbus_message_get_signature (message),

```



```

                                                                    message,
client_serial);

    _dbus_verbose ("middle\n");
    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* this calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
}

/**
 * Like dbus_connection_send(), but assumes the connection
 * is already locked on function entry, and unlocks before returning.
 *
 * @param connection the connection
 * @param message the message to send
 * @param client_serial return location for client serial of sent
message
 * @returns #FALSE on out-of-memory
 */
dbus_bool_t
_dbus_connection_send_and_unlock (DBusConnection *connection,
                                DBusMessage      *message,
                                dbus_uint32_t     *client_serial)
{
    DBusPreallocatedSend *preallocated;

    _dbus_assert (connection != NULL);
    _dbus_assert (message != NULL);

    preallocated = _dbus_connection_preallocate_send_unlocked
(connection);
    if (preallocated == NULL)
    {
        CONNECTION_UNLOCK (connection);
        return FALSE;
    }

    _dbus_connection_send_preallocated_and_unlock (connection,
                                                    preallocated,
                                                    message,
                                                    client_serial);

    return TRUE;
}

/**
 * Used internally to handle the semantics of
dbus_server_set_new_connection_function().
 * If the new connection function does not ref the connection, we want
to close it.
 */

```

```

* A bit of a hack, probably the new connection function should have
returned a value
* for whether to close, or should have had to close the connection
itself if it
* didn't want it.
*
* But, this works OK as long as the new connection function doesn't
do anything
* crazy like keep the connection around without ref'ing it.
*
* We have to lock the connection across refcount check and close in
case
* the new connection function spawns a thread that closes and unrefs.
* In that case, if the app thread
* closes and unrefs first, we'll harmlessly close again; if the app
thread
* still has the ref, we'll close and then the app will close
harmlessly.
* If the app unrefs without closing, the app is broken since if the
* app refs from the new connection function it is supposed to also
close.
*
* If we didn't atomically check the refcount and close with the lock
held
* though, we could screw this up.
*
* @param connection the connection
*/
void
_dbus_connection_close_if_only_one_ref (DBusConnection *connection)
{
    dbus_int32_t refcount;

    CONNECTION_LOCK (connection);

    refcount = _dbus_atomic_get (&connection->refcount);
    /* The caller should have at least one ref */
    _dbus_assert (refcount >= 1);

    if (refcount == 1)
        _dbus_connection_close_possibly_shared_and_unlock (connection);
    else
        CONNECTION_UNLOCK (connection);
}

/**
* When a function that blocks has been called with a timeout, and we
* run out of memory, the time to wait for memory is based on the
* timeout. If the caller was willing to block a long time we wait a
* relatively long time for memory, if they were only willing to block
* briefly then we retry for memory at a rapid rate.

```



```

        dbus_message_unref (message);
        message = NULL;
        goto out;
    }
}

out:
    return message;
}

/*
 * Peek the incoming queue to see if we got reply for a specific
 * serial
 */
static dbus_bool_t
_dbus_connection_peek_for_reply_unlocked (DBusConnection *connection,
                                         dbus_uint32_t
client_serial)
{
    DBusList *link;
    HAVE_LOCK_CHECK (connection);

    link = _dbus_list_get_first_link (&connection->incoming_messages);

    while (link != NULL)
    {
        DBusMessage *reply = link->data;

        if (dbus_message_get_reply_serial (reply) == client_serial)
        {
            _dbus_verbose ("%s reply to %d found in queue\n",
                _DBUS_FUNCTION_NAME, client_serial);
            return TRUE;
        }
        link = _dbus_list_get_next_link (&connection->incoming_messages,
link);
    }

    return FALSE;
}

/* This is slightly strange since we can pop a message here without
 * the dispatch lock.
 */
static DBusMessage*
check_for_reply_unlocked (DBusConnection *connection,
                        dbus_uint32_t client_serial)
{
    DBusList *link;

    HAVE_LOCK_CHECK (connection);

```

```

link = _dbus_list_get_first_link (&connection->incoming_messages);

while (link != NULL)
{
    DBusMessage *reply = link->data;

    if (dbus_message_get_reply_serial (reply) == client_serial)
    {
        _dbus_list_remove_link (&connection->incoming_messages, link);
        connection->n_incoming -= 1;
        return reply;
    }
    link = _dbus_list_get_next_link (&connection->incoming_messages,
link);
}

return NULL;
}

static void
connection_timeout_and_complete_all_pending_calls_unlocked
(DBusConnection *connection)
{
    /* We can't iterate over the hash in the normal way since we'll be
    * dropping the lock for each item. So we restart the
    * iter each time as we drain the hash table.
    */

    while (_dbus_hash_table_get_n_entries (connection->pending_replies)
> 0)
    {
        DBusPendingCall *pending;
        DBusHashIter iter;

        _dbus_hash_iter_init (connection->pending_replies, &iter);
        _dbus_hash_iter_next (&iter);

        pending = _dbus_hash_iter_get_value (&iter);
        _dbus_pending_call_ref_unlocked (pending);

        _dbus_pending_call_queue_timeout_error_unlocked (pending,
                                                         connection);

        if (_dbus_pending_call_is_timeout_added_unlocked (pending))
            _dbus_connection_remove_timeout_unlocked (connection,

_dbus_pending_call_get_timeout_unlocked (pending));
        _dbus_pending_call_set_timeout_added_unlocked (pending, FALSE);
        _dbus_hash_iter_remove_entry (&iter);

        _dbus_pending_call_unref_and_unlock (pending);
        CONNECTION_LOCK (connection);
    }
}

```

```

    }
    HAVE_LOCK_CHECK (connection);
}

static void
complete_pending_call_and_unlock (DBusConnection *connection,
                                  DbusPendingCall *pending,
                                  DbusMessage *message)
{
    _dbus_pending_call_set_reply_unlocked (pending, message);
    _dbus_pending_call_ref_unlocked (pending); /* in case there's no app
with a ref held */
    _dbus_connection_detach_pending_call_and_unlock (connection,
pending);

    /* Must be called unlocked since it invokes app callback */
    _dbus_pending_call_complete (pending);
    dbus_pending_call_unref (pending);
}

static dbus_bool_t
check_for_reply_and_update_dispatch_unlocked (DBusConnection
*connection,
                                             DbusPendingCall
*pending)
{
    DbusMessage *reply;
    DbusDispatchStatus status;

    reply = check_for_reply_unlocked (connection,
_dbus_pending_call_get_reply_serial_unlocked (pending));
    if (reply != NULL)
    {
        _dbus_verbose ("checked for reply\n");

        _dbus_verbose ("dbus_connection_send_with_reply_and_block(): got
reply\n");

        complete_pending_call_and_unlock (connection, pending, reply);
        dbus_message_unref (reply);

        CONNECTION_LOCK (connection);
        status = _dbus_connection_get_dispatch_status_unlocked
(connection);
        _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
        dbus_pending_call_unref (pending);

        return TRUE;
    }
}

```

```

    return FALSE;
}

/**
 * Blocks until a pending call times out or gets a reply.
 *
 * Does not re-enter the main loop or run filter/path-registered
 * callbacks. The reply to the message will not be seen by
 * filter callbacks.
 *
 * Returns immediately if pending call already got a reply.
 *
 * @todo could use performance improvements (it keeps scanning
 * the whole message queue for example)
 *
 * @param pending the pending call we block for a reply on
 */
void
_dbus_connection_block_pending_call (DBusPendingCall *pending)
{
    long start_tv_sec, start_tv_usec;
    long tv_sec, tv_usec;
    DBusDispatchStatus status;
    DBusConnection *connection;
    dbus_uint32_t client_serial;
    DBusTimeout *timeout;
    int timeout_milliseconds, elapsed_milliseconds;

    _dbus_assert (pending != NULL);

    if (dbus_pending_call_get_completed (pending))
        return;

    dbus_pending_call_ref (pending); /* necessary because the call could
be canceled */

    connection = _dbus_pending_call_get_connection_and_lock (pending);

    /* Flush message queue - note, can affect dispatch status */
    _dbus_connection_flush_unlocked (connection);

    client_serial = _dbus_pending_call_get_reply_serial_unlocked
(pending);

    /* note that timeout_milliseconds is limited to a smallish value
 * in _dbus_pending_call_new() so overflows aren't possible
 * below
 */
    timeout = _dbus_pending_call_get_timeout_unlocked (pending);
    _dbus_get_monotonic_time (&start_tv_sec, &start_tv_usec);
    if (timeout)
    {

```

```

        timeout_milliseconds = dbus_timeout_get_interval (timeout);

        _dbus_verbose ("dbus_connection_send_with_reply_and_block():
will block %d milliseconds for reply serial %u from %ld sec %ld
usec\n",
                    timeout_milliseconds,
                    client_serial,
                    start_tv_sec, start_tv_usec);
    }
    else
    {
        timeout_milliseconds = -1;

        _dbus_verbose ("dbus_connection_send_with_reply_and_block():
will block for reply serial %u\n", client_serial);
    }

    /* check to see if we already got the data off the socket */
    /* from another blocked pending call */
    if (check_for_reply_and_update_dispatch_unlocked (connection,
pending))
        return;

    /* Now we wait... */
    /* always block at least once as we know we don't have the reply yet
*/
    _dbus_connection_do_iteration_unlocked (connection,
                                        pending,
                                        DBUS_ITERATION_DO_READING |
                                        DBUS_ITERATION_BLOCK,
                                        timeout_milliseconds);

recheck_status:

    _dbus_verbose ("top of recheck\n");

    HAVE_LOCK_CHECK (connection);

    /* queue messages and get status */

    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* the get_completed() is in case a dispatch() while we were
blocking
* got the reply instead of us.
*/
    if (_dbus_pending_call_get_completed_unlocked (pending))
    {
        _dbus_verbose ("Pending call completed by dispatch\n");
        _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
        dbus_pending_call_unref (pending);
    }

```

```

        return;
    }

    if (status == DBUS_DISPATCH_DATA_REMAINS)
    {
        if (check_for_reply_and_update_dispatch_unlocked (connection,
pending))
            return;
    }

    _dbus_get_monotonic_time (&tv_sec, &tv_usec);
    elapsed_milliseconds = (tv_sec - start_tv_sec) * 1000 +
        (tv_usec - start_tv_usec) / 1000;

    if (!_dbus_connection_get_is_connected_unlocked (connection))
    {
        DBusMessage *error_msg;

        error_msg = generate_local_error_message (client_serial,
DBUS_ERROR_DISCONNECTED,
"Connection was
disconnected before a reply was received");

        /* on OOM error_msg is set to NULL */
        complete_pending_call_and_unlock (connection, pending,
error_msg);
        _dbus_pending_call_unref (pending);
        return;
    }
    else if (connection->disconnect_message_link == NULL)
        _dbus_verbose ("dbus_connection_send_with_reply_and_block():
disconnected\n");
    else if (timeout == NULL)
    {
        if (status == DBUS_DISPATCH_NEED_MEMORY)
        {
            /* Try sleeping a bit, as we aren't sure we need to block
for reading,
* we may already have a reply in the buffer and just can't
process
* it.
*/
            _dbus_verbose ("dbus_connection_send_with_reply_and_block()
waiting for more memory\n");

            _dbus_memory_pause_based_on_timeout (timeout_milliseconds -
elapsed_milliseconds);
        }
        else
        {
            /* block again, we don't have the reply buffered yet. */

```

```

        _dbus_connection_do_iteration_unlocked (connection,
                                                pending,

DBUS_ITERATION_DO_READING |

DBUS_ITERATION_BLOCK,

                                                timeout_milliseconds
- elapsed_milliseconds);
    }

    goto recheck_status;
}
else if (tv_sec < start_tv_sec)
    _dbus_verbose ("dbus_connection_send_with_reply_and_block(): clock
set backward\n");
else if (elapsed_milliseconds < timeout_milliseconds)
{
    _dbus_verbose ("dbus_connection_send_with_reply_and_block(): %d
milliseconds remain\n", timeout_milliseconds - elapsed_milliseconds);

    if (status == DBUS_DISPATCH_NEED_MEMORY)
    {
        /* Try sleeping a bit, as we aren't sure we need to block
for reading,
* we may already have a reply in the buffer and just can't
process
* it.
*/
        _dbus_verbose ("dbus_connection_send_with_reply_and_block()
waiting for more memory\n");

        _dbus_memory_pause_based_on_timeout (timeout_milliseconds -
elapsed_milliseconds);
    }
    else
    {
        /* block again, we don't have the reply buffered yet. */
        _dbus_connection_do_iteration_unlocked (connection,
                                                NULL,

DBUS_ITERATION_DO_READING |

DBUS_ITERATION_BLOCK,

                                                timeout_milliseconds
- elapsed_milliseconds);
    }

    goto recheck_status;
}

_dbus_verbose ("dbus_connection_send_with_reply_and_block(): Waited
%d milliseconds and got no reply\n",

```

```

        elapsed_milliseconds);

    _dbus_assert (!_dbus_pending_call_get_completed_unlocked (pending));

    /* unlock and call user code */
    complete_pending_call_and_unlock (connection, pending, NULL);

    /* update user code on dispatch status */
    CONNECTION_LOCK (connection);
    status = _dbus_connection_get_dispatch_status_unlocked (connection);
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
    dbus_pending_call_unref (pending);
}

/** @} */

/**
 * @addtogroup DBusConnection
 *
 * @{
 */

/**
 * Gets a connection to a remote address. If a connection to the given
 * address already exists, returns the existing connection with its
 * reference count incremented. Otherwise, returns a new connection
 * and saves the new connection for possible re-use if a future call
 * to dbus_connection_open() asks to connect to the same server.
 *
 * Use dbus_connection_open_private() to get a dedicated connection
 * not shared with other callers of dbus_connection_open().
 *
 * If the open fails, the function returns #NULL, and provides a
 * reason for the failure in the error parameter. Pass #NULL for the
 * error parameter if you aren't interested in the reason for
 * failure.
 *
 * Because this connection is shared, no user of the connection
 * may call dbus_connection_close(). However, when you are done with
the
 * connection you should call dbus_connection_unref().
 *
 * @note Prefer dbus_connection_open() to
dbus_connection_open_private()
 * unless you have good reason; connections are expensive enough
 * that it's wasteful to create lots of connections to the same
 * server.
 *
 * @param address the address.
 * @param error address where an error can be returned.
 * @returns new connection, or #NULL on failure.

```



```

*/
DBusConnection*
dbus_connection_open (const char      *address,
                     DBusError      *error)
{
    DBusConnection *connection;

    _dbus_return_val_if_fail (address != NULL, NULL);
    _dbus_return_val_if_error_is_set (error, NULL);

    connection = _dbus_connection_open_internal (address,
                                                TRUE,
                                                error);

    return connection;
}

/**
 * Opens a new, dedicated connection to a remote address. Unlike
 * dbus_connection_open(), always creates a new connection.
 * This connection will not be saved or recycled by libdbus.
 *
 * If the open fails, the function returns #NULL, and provides a
 * reason for the failure in the error parameter. Pass #NULL for the
 * error parameter if you aren't interested in the reason for
 * failure.
 *
 * When you are done with this connection, you must
 * dbus_connection_close() to disconnect it,
 * and dbus_connection_unref() to free the connection object.
 *
 * (The dbus_connection_close() can be skipped if the
 * connection is already known to be disconnected, for example
 * if you are inside a handler for the Disconnected signal.)
 *
 * @note Prefer dbus_connection_open() to
dbus_connection_open_private()
 * unless you have good reason; connections are expensive enough
 * that it's wasteful to create lots of connections to the same
 * server.
 *
 * @param address the address.
 * @param error address where an error can be returned.
 * @returns new connection, or #NULL on failure.
 */
DBusConnection*
dbus_connection_open_private (const char      *address,
                             DBusError      *error)
{
    DBusConnection *connection;

    _dbus_return_val_if_fail (address != NULL, NULL);

```

```

_dbus_return_val_if_error_is_set (error, NULL);

connection = _dbus_connection_open_internal (address,
                                           FALSE,
                                           error);

return connection;
}

/**
 * Increments the reference count of a DBusConnection.
 *
 * @param connection the connection.
 * @returns the connection.
 */
DBusConnection *
dbus_connection_ref (DBusConnection *connection)
{
    dbus_int32_t old_refcount;

    _dbus_return_val_if_fail (connection != NULL, NULL);
    _dbus_return_val_if_fail (connection->generation ==
_dbus_current_generation, NULL);
    old_refcount = _dbus_atomic_inc (&connection->refcount);
    _dbus_connection_trace_ref (connection, old_refcount, old_refcount +
1,
    "ref");

    return connection;
}

static void
free_outgoing_message (void *element,
                      void *data)
{
    DBusMessage *message = element;
    DBusConnection *connection = data;

    _dbus_message_remove_counter (message, connection->outgoing_counter);
    dbus_message_unref (message);
}

/* This is run without the mutex held, but after the last reference
 * to the connection has been dropped we should have no thread-related
 * problems
 */
static void
_dbus_connection_last_unref (DBusConnection *connection)
{
    DBusList *link;

```

```

_dbus_verbose ("Finalizing connection %p\n", connection);

_dbus_assert (_dbus_atomic_get (&connection->refcount) == 0);

/* You have to disconnect the connection before unref:ing it.
Otherwise
* you won't get the disconnected message.
*/
_dbus_assert (!_dbus_transport_get_is_connected (connection-
>transport));
_dbus_assert (connection->server_guid == NULL);

/* ---- We're going to call various application callbacks here, hope
it doesn't break anything... */
_dbus_object_tree_free_all_unlocked (connection->objects);

dbus_connection_set_dispatch_status_function (connection, NULL,
NULL, NULL);
dbus_connection_set_wakeup_main_function (connection, NULL, NULL,
NULL);
dbus_connection_set_unix_user_function (connection, NULL, NULL,
NULL);

_dbus_watch_list_free (connection->watches);
connection->watches = NULL;

_dbus_timeout_list_free (connection->timeouts);
connection->timeouts = NULL;

_dbus_data_slot_list_free (&connection->slot_list);

link = _dbus_list_get_first_link (&connection->filter_list);
while (link != NULL)
{
    DBusMessageFilter *filter = link->data;
    DBusList *next = _dbus_list_get_next_link (&connection-
>filter_list, link);

    filter->function = NULL;
    _dbus_message_filter_unref (filter); /* calls app callback */
    link->data = NULL;

    link = next;
}
_dbus_list_clear (&connection->filter_list);

/* ---- Done with stuff that invokes application callbacks */

_dbus_object_tree_unref (connection->objects);

_dbus_hash_table_unref (connection->pending_replies);
connection->pending_replies = NULL;

```

```

    _dbus_list_clear (&connection->filter_list);

    _dbus_list_foreach (&connection->outgoing_messages,
                        free_outgoing_message,
                        connection);
    _dbus_list_clear (&connection->outgoing_messages);

    _dbus_list_foreach (&connection->incoming_messages,
                        (DBusForeachFunction) dbus_message_unref,
                        NULL);
    _dbus_list_clear (&connection->incoming_messages);

    _dbus_counter_unref (connection->outgoing_counter);

    _dbus_transport_unref (connection->transport);

    if (connection->disconnect_message_link)
    {
        DBusMessage *message = connection->disconnect_message_link-
>data;
        dbus_message_unref (message);
        _dbus_list_free_link (connection->disconnect_message_link);
    }

    _dbus_condvar_free_at_location (&connection->dispatch_cond);
    _dbus_condvar_free_at_location (&connection->io_path_cond);

    _dbus_cmutex_free_at_location (&connection->io_path_mutex);
    _dbus_cmutex_free_at_location (&connection->dispatch_mutex);

    _dbus_rmutex_free_at_location (&connection->slot_mutex);

    _dbus_rmutex_free_at_location (&connection->mutex);

    dbus_free (connection);
}

/**
 * Decrements the reference count of a DBusConnection, and finalizes
 * it if the count reaches zero.
 *
 * Note: it is a bug to drop the last reference to a connection that
 * is still connected.
 *
 * For shared connections, libdbus will own a reference
 * as long as the connection is connected, so you can know that either
 * you don't have the last reference, or it's OK to drop the last
reference.
 * Most connections are shared. dbus_connection_open() and
dbus_bus_get()
 * return shared connections.

```

```

*
* For private connections, the creator of the connection must arrange
for
* dbus_connection_close() to be called prior to dropping the last
reference.
* Private connections come from dbus_connection_open_private() or
dbus_bus_get_private().
*
* @param connection the connection.
*/
void
dbus_connection_unref (DBusConnection *connection)
{
    dbus_int32_t old_refcount;

    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (connection->generation ==
_dbus_current_generation);

    old_refcount = _dbus_atomic_dec (&connection->refcount);

    _dbus_connection_trace_ref (connection, old_refcount, old_refcount -
1,
    "unref");

    if (old_refcount == 1)
    {
#ifdef DBUS_DISABLE_CHECKS
        if (_dbus_transport_get_is_connected (connection->transport))
        {
            _dbus_warn_check_failed ("The last reference on a connection
was dropped without closing the connection. This is a bug in an
application. See dbus_connection_unref() documentation for
details.\n%s",
                connection->shareable ?
                "Most likely, the application
called unref() too many times and removed a reference belonging to
libdbus, since this is a shared connection.\n" :
                "Most likely, the application was
supposed to call dbus_connection_close(), since this is a private
connection.\n");
            return;
        }
    }
#endif
    _dbus_connection_last_unref (connection);
}

/*
* Note that the transport can disconnect itself (other end drops us)
* and in that case this function never runs. So this function must
* not do anything more than disconnect the transport and update the

```

```

* dispatch status.
*
* If the transport self-disconnects, then we assume someone will
* dispatch the connection to cause the dispatch status update.
*/
static void
_dbus_connection_close_possibly_shared_and_unlock (DBusConnection
*connection)
{
    DBusDispatchStatus status;

    HAVE_LOCK_CHECK (connection);

    _dbus_verbose ("Disconnecting %p\n", connection);

    /* We need to ref because update_dispatch_status_and_unlock will
    unref
    * the connection if it was shared and libdbus was the only
    remaining
    * refcount holder.
    */
    _dbus_connection_ref_unlocked (connection);

    _dbus_transport_disconnect (connection->transport);

    /* This has the side effect of queuing the disconnect message link
    * (unless we don't have enough memory, possibly, so don't assert
    it).
    * After the disconnect message link is queued,
    dbus_bus_get/dbus_connection_open
    * should never again return the newly-disconnected connection.
    *
    * However, we only unref the shared connection and
    exit_on_disconnect when
    * the disconnect message reaches the head of the message queue,
    * NOT when it's first queued.
    */
    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* This calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
    status);

    /* Could also call out to user code */
    dbus_connection_unref (connection);
}

/**
* Closes a private connection, so no further data can be sent or
received.
* This disconnects the transport (such as a socket) underlying the
* connection.

```

```

*
* Attempts to send messages after closing a connection are safe, but
will result in
* error replies generated locally in libdbus.
*
* This function does not affect the connection's reference count.
It's
* safe to close a connection more than once; all calls after the
* first do nothing. It's impossible to "reopen" a connection, a
* new connection must be created. This function may result in a call
* to the DBusDispatchStatusFunction set with
* dbus_connection_set_dispatch_status_function(), as the disconnect
* message it generates needs to be dispatched.
*
* If a connection is dropped by the remote application, it will
* close itself.
*
* You must close a connection prior to releasing the last reference
to
* the connection. If you dbus_connection_unref() for the last time
* without closing the connection, the results are undefined; it
* is a bug in your program and libdbus will try to print a warning.
*
* You may not close a shared connection. Connections created with
* dbus_connection_open() or dbus_bus_get() are shared.
* These connections are owned by libdbus, and applications should
* only unref them, never close them. Applications can know it is
* safe to unref these connections because libdbus will be holding a
* reference as long as the connection is open. Thus, either the
* connection is closed and it is OK to drop the last reference,
* or the connection is open and the app knows it does not have the
* last reference.
*
* Connections created with dbus_connection_open_private() or
* dbus_bus_get_private() are not kept track of or referenced by
* libdbus. The creator of these connections is responsible for
* calling dbus_connection_close() prior to releasing the last
* reference, if the connection is not already disconnected.
*
* @param connection the private (unshared) connection to close
*/
void
dbus_connection_close (DBusConnection *connection)
{
    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (connection->generation ==
_dbus_current_generation);

    CONNECTION_LOCK (connection);

#ifdef DBUS_DISABLE_CHECKS
    if (connection->shareable)

```

```

    {
        CONNECTION_UNLOCK (connection);

        _dbus_warn_check_failed ("Applications must not close shared
connections - see dbus_connection_close() docs. This is a bug in the
application.\n");
        return;
    }
#endif

    _dbus_connection_close_possibly_shared_and_unlock (connection);
}

static dbus_bool_t
_dbus_connection_get_is_connected_unlocked (DBusConnection
*connection)
{
    HAVE_LOCK_CHECK (connection);
    return _dbus_transport_get_is_connected (connection->transport);
}

/**
 * Gets whether the connection is currently open. A connection may
 * become disconnected when the remote application closes its end, or
 * exits; a connection may also be disconnected with
 * dbus_connection_close().
 *
 * There are not separate states for "closed" and "disconnected," the
two
 * terms are synonymous. This function should really be called
 * get_is_open() but for historical reasons is not.
 *
 * @param connection the connection.
 * @returns #TRUE if the connection is still alive.
 */
dbus_bool_t
dbus_connection_get_is_connected (DBusConnection *connection)
{
    dbus_bool_t res;

    _dbus_return_val_if_fail (connection != NULL, FALSE);

    CONNECTION_LOCK (connection);
    res = _dbus_connection_get_is_connected_unlocked (connection);
    CONNECTION_UNLOCK (connection);

    return res;
}

/**
 * Gets whether the connection was authenticated. (Note that
 * if the connection was authenticated then disconnected,

```



```

* this function still returns #TRUE)
*
* @param connection the connection
* @returns #TRUE if the connection was ever authenticated
*/
dbus_bool_t
dbus_connection_get_is_authenticated (DBusConnection *connection)
{
    dbus_bool_t res;

    _dbus_return_val_if_fail (connection != NULL, FALSE);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_is_authenticated (connection->transport);
    CONNECTION_UNLOCK (connection);

    return res;
}

/**
 * Gets whether the connection is not authenticated as a specific
 * user.  If the connection is not authenticated, this function
 * returns #TRUE, and if it is authenticated but as an anonymous user,
 * it returns #FALSE.  If it is authenticated as a specific user, then
 * this returns #FALSE.  (Note that if the connection was authenticated
 * as anonymous then disconnected, this function still returns #TRUE.)
 *
 * If the connection is not anonymous, you can use
 * dbus_connection_get_unix_user() and
 * dbus_connection_get_windows_user() to see who it's authorized as.
 *
 * If you want to prevent non-anonymous authorization, use
 * dbus_server_set_auth_mechanisms() to remove the mechanisms that
 * allow proving user identity (i.e. only allow the ANONYMOUS
 * mechanism).
 *
 * @param connection the connection
 * @returns #TRUE if not authenticated or authenticated as anonymous
 */
dbus_bool_t
dbus_connection_get_is_anonymous (DBusConnection *connection)
{
    dbus_bool_t res;

    _dbus_return_val_if_fail (connection != NULL, FALSE);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_is_anonymous (connection->transport);
    CONNECTION_UNLOCK (connection);

    return res;
}

```

```

/**
 * Gets the ID of the server address we are authenticated to, if this
 * connection is on the client side. If the connection is on the
 * server side, this will always return #NULL - use
dbus_server_get_id()
 * to get the ID of your own server, if you are the server side.
 *
 * If a client-side connection is not authenticated yet, the ID may be
 * available if it was included in the server address, but may not be
 * available. The only way to be sure the server ID is available
 * is to wait for authentication to complete.
 *
 * In general, each mode of connecting to a given server will have
 * its own ID. So for example, if the session bus daemon is listening
 * on UNIX domain sockets and on TCP, then each of those modalities
 * will have its own server ID.
 *
 * If you want an ID that identifies an entire session bus, look at
 * dbus_bus_get_id() instead (which is just a convenience wrapper
 * around the org.freedesktop.DBus.GetId method invoked on the bus).
 *
 * You can also get a machine ID; see dbus_get_local_machine_id() to
 * get the machine you are on. There isn't a convenience wrapper, but
 * you can invoke org.freedesktop.DBus.Peer.GetMachineId on any peer
 * to get the machine ID on the other end.
 *
 * The D-Bus specification describes the server ID and other IDs in a
 * bit more detail.
 *
 * @param connection the connection
 * @returns the server ID or #NULL if no memory or the connection is
server-side
 */
char*
dbus_connection_get_server_id (DBusConnection *connection)
{
    char *id;

    _dbus_return_val_if_fail (connection != NULL, NULL);

    CONNECTION_LOCK (connection);
    id = _dbus_strdup (_dbus_transport_get_server_id (connection-
>transport));
    CONNECTION_UNLOCK (connection);

    return id;
}

/**
 * Tests whether a certain type can be send via the connection. This
 * will always return TRUE for all types, with the exception of

```

```

* DBUS_TYPE_UNIX_FD. The function will return TRUE for
* DBUS_TYPE_UNIX_FD only on systems that know Unix file descriptors
* and can send them via the chosen transport and when the remote side
* supports this.
*
* This function can be used to do runtime checking for types that
* might be unknown to the specific D-Bus client implementation
* version, i.e. it will return FALSE for all types this
* implementation does not know, including invalid or reserved types.
*
* @param connection the connection
* @param type the type to check
* @returns TRUE if the type may be send via the connection
*/
dbus_bool_t
dbus_connection_can_send_type(DBusConnection *connection,
                             int type)
{
    _dbus_return_val_if_fail (connection != NULL, FALSE);

    if (!dbus_type_is_valid (type))
        return FALSE;

    if (type != DBUS_TYPE_UNIX_FD)
        return TRUE;

#ifdef HAVE_UNIX_FD_PASSING
    {
        dbus_bool_t b;

        CONNECTION_LOCK(connection);
        b = _dbus_transport_can_pass_unix_fd(connection->transport);
        CONNECTION_UNLOCK(connection);

        return b;
    }
#endif

    return FALSE;
}

/**
* Set whether _exit() should be called when the connection receives a
* disconnect signal. The call to _exit() comes after any handlers for
* the disconnect signal run; handlers can cancel the exit by calling
* this function.
*
* By default, exit_on_disconnect is #FALSE; but for message bus
* connections returned from dbus_bus_get() it will be toggled on
* by default.
*
* @param connection the connection

```

```

    * @param exit_on_disconnect #TRUE if _exit() should be called after a
    disconnect signal
    */
void
dbus_connection_set_exit_on_disconnect (DBusConnection *connection,
                                       dbus_bool_t
exit_on_disconnect)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    connection->exit_on_disconnect = exit_on_disconnect != FALSE;
    CONNECTION_UNLOCK (connection);
}

/**
 * Preallocates resources needed to send a message, allowing the
message
 * to be sent without the possibility of memory allocation failure.
 * Allows apps to create a future guarantee that they can send
 * a message regardless of memory shortages.
 *
 * @param connection the connection we're preallocating for.
 * @returns the preallocated resources, or #NULL
 */
DBusPreallocatedSend*
dbus_connection_preallocate_send (DBusConnection *connection)
{
    DBusPreallocatedSend *preallocated;

    _dbus_return_val_if_fail (connection != NULL, NULL);

    CONNECTION_LOCK (connection);

    preallocated =
        _dbus_connection_preallocate_send_unlocked (connection);

    CONNECTION_UNLOCK (connection);

    return preallocated;
}

/**
 * Frees preallocated message-sending resources from
 * dbus_connection_preallocate_send(). Should only
 * be called if the preallocated resources are not used
 * to send a message.
 *
 * @param connection the connection
 * @param preallocated the resources
 */
void

```

```

dbus_connection_free_preallocated_send (DBusConnection
*connection,
                                     DBusPreallocatedSend
*preallocated)
{
    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (preallocated != NULL);
    _dbus_return_if_fail (connection == preallocated->connection);

    _dbus_list_free_link (preallocated->queue_link);
    _dbus_counter_unref (preallocated->counter_link->data);
    _dbus_list_free_link (preallocated->counter_link);
    dbus_free (preallocated);
}

/**
 * Sends a message using preallocated resources. This function cannot
 fail.
 * It works identically to dbus_connection_send() in other respects.
 * Preallocated resources comes from
 dbus_connection_preallocate_send().
 * This function "consumes" the preallocated resources, they need not
 * be freed separately.
 *
 * @param connection the connection
 * @param preallocated the preallocated resources
 * @param message the message to send
 * @param client_serial return location for client serial assigned to
 the message
 */
void
dbus_connection_send_preallocated (DBusConnection      *connection,
                                   DBusPreallocatedSend *preallocated,
                                   DBusMessage          *message,
                                   dbus_uint32_t
*client_serial)
{
    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (preallocated != NULL);
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (preallocated->connection == connection);
    _dbus_return_if_fail (dbus_message_get_type (message) !=
DBUS_MESSAGE_TYPE_METHOD_CALL ||
                           dbus_message_get_member (message) != NULL);
    _dbus_return_if_fail (dbus_message_get_type (message) !=
DBUS_MESSAGE_TYPE_SIGNAL ||
                           (dbus_message_get_interface (message) != NULL
&&
                            dbus_message_get_member (message) != NULL));

    CONNECTION_LOCK (connection);

```

```

#ifdef HAVE_UNIX_FD_PASSING

    if (!_dbus_transport_can_pass_unix_fd(connection->transport) &&
        message->n_unix_fds > 0)
    {
        /* Refuse to send fds on a connection that cannot handle
           them. Unfortunately we cannot return a proper error here, so
           the best we can is just return. */
        CONNECTION_UNLOCK (connection);
        return;
    }

#endif

    _dbus_connection_send_preallocated_and_unlock (connection,
                                                  preallocated,
                                                  message, client_serial);
}

static dbus_bool_t
_dbus_connection_send_unlocked_no_update (DBusConnection *connection,
                                          DBusMessage *message,
                                          dbus_uint32_t
*client_serial)
{
    DBusPreallocatedSend *preallocated;

    _dbus_assert (connection != NULL);
    _dbus_assert (message != NULL);

    preallocated = _dbus_connection_preallocate_send_unlocked
(connection);
    if (preallocated == NULL)
        return FALSE;

    _dbus_connection_send_preallocated_unlocked_no_update (connection,
                                                          preallocated,
                                                          message,
client_serial);
    return TRUE;
}

/**
 * Adds a message to the outgoing message queue. Does not block to
 * write the message to the network; that happens asynchronously. To
 * force the message to be written, call dbus_connection_flush()
however
 * it is not necessary to call dbus_connection_flush() by hand; the
 * message will be sent the next time the main loop is run.
 * dbus_connection_flush() should only be used, for example, if
 * the application was expected to exit before running the main loop.

```



```

static dbus_bool_t
reply_handler_timeout (void *data)
{
    DBusConnection *connection;
    DBusDispatchStatus status;
    DBusPendingCall *pending = data;

    connection = _dbus_pending_call_get_connection_and_lock (pending);
    _dbus_connection_ref_unlocked (connection);

    _dbus_pending_call_queue_timeout_error_unlocked (pending,
                                                    connection);
    _dbus_connection_remove_timeout_unlocked (connection,
                                             _dbus_pending_call_get_timeout_unlocked (pending));
    _dbus_pending_call_set_timeout_added_unlocked (pending, FALSE);

    _dbus_verbose ("middle\n");
    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* Unlocks, and calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
    dbus_connection_unref (connection);

    return TRUE;
}

/**
 * Queues a message to send, as with dbus_connection_send(),
 * but also returns a #DBusPendingCall used to receive a reply to the
 * message. If no reply is received in the given timeout_milliseconds,
 * this function expires the pending reply and generates a synthetic
 * error reply (generated in-process, not by the remote application)
 * indicating that a timeout occurred.
 *
 * A #DBusPendingCall will see a reply message before any filters or
 * registered object path handlers. See dbus_connection_dispatch() for
 * details on when handlers are run.
 *
 * A #DBusPendingCall will always see exactly one reply message,
 * unless it's cancelled with dbus_pending_call_cancel().
 *
 * If #NULL is passed for the pending_return, the #DBusPendingCall
 * will still be generated internally, and used to track
 * the message reply timeout. This means a timeout error will
 * occur if no reply arrives, unlike with dbus_connection_send().
 *
 * If -1 is passed for the timeout, a sane default timeout is used. -1
 * is typically the best value for the timeout for this reason, unless

```



```

* you want a very short or very long timeout. If
#DBUS_TIMEOUT_INFINITE is
* passed for the timeout, no timeout will be set and the call will
block
* forever.
*
* @warning if the connection is disconnected or you try to send Unix
* file descriptors on a connection that does not support them, the
* #DBusPendingCall will be set to #NULL, so be careful with this.
*
* @param connection the connection
* @param message the message to send
* @param pending_return return location for a #DBusPendingCall
* object, or #NULL if connection is disconnected or when you try to
* send Unix file descriptors on a connection that does not support
* them.
* @param timeout_milliseconds timeout in milliseconds, -1 (or
* #DBUS_TIMEOUT_USE_DEFAULT) for default or #DBUS_TIMEOUT_INFINITE
for no
* timeout
* @returns #FALSE if no memory, #TRUE otherwise.
*
*/
dbus_bool_t
dbus_connection_send_with_reply (DBusConnection      *connection,
                                DBusMessage         *message,
                                DBusPendingCall     **pending_return,
                                int
                                timeout_milliseconds)
{
    DBusPendingCall *pending;
    dbus_int32_t serial = -1;
    DBusDispatchStatus status;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (timeout_milliseconds >= 0 ||
timeout_milliseconds == -1, FALSE);

    if (pending_return)
        *pending_return = NULL;

    CONNECTION_LOCK (connection);

#ifdef HAVE_UNIX_FD_PASSING

    if (!_dbus_transport_can_pass_unix_fd(connection->transport) &&
        message->n_unix_fds > 0)
    {
        /* Refuse to send fds on a connection that cannot handle
        them. Unfortunately we cannot return a proper error here, so
        the best we can do is return TRUE but leave *pending_return

```



```

    if (pending_return)
        *pending_return = pending; /* hand off refcount */
    else
    {
        _dbus_connection_detach_pending_call_unlocked (connection,
pending);
        /* we still have a ref to the pending call in this case, we
unref
        * after unlocking, below
        */
    }

    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* this calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    if (pending_return == NULL)
        dbus_pending_call_unref (pending);

    return TRUE;

error:
    CONNECTION_UNLOCK (connection);
error_unlocked:
    dbus_pending_call_unref (pending);
    return FALSE;
}

/**
 * Sends a message and blocks a certain time period while waiting for
 * a reply. This function does not reenter the main loop,
 * i.e. messages other than the reply are queued up but not
 * processed. This function is used to invoke method calls on a
 * remote object.
 *
 * If a normal reply is received, it is returned, and removed from the
 * incoming message queue. If it is not received, #NULL is returned
 * and the error is set to #DBUS_ERROR_NO_REPLY. If an error reply is
 * received, it is converted to a #DBusError and returned as an error,
 * then the reply message is deleted and #NULL is returned. If
 * something else goes wrong, result is set to whatever is
 * appropriate, such as #DBUS_ERROR_NO_MEMORY or
 * #DBUS_ERROR_DISCONNECTED.
 *
 * @warning While this function blocks the calling thread will not be
 * processing the incoming message queue. This means you can end up
 * deadlocked if the application you're talking to needs you to reply
 * to a method. To solve this, either avoid the situation, block in a
 * separate thread from the main connection-dispatching thread, or use
 * dbus_pending_call_set_notify() to avoid blocking.

```

```

*
* @param connection the connection
* @param message the message to send
* @param timeout_milliseconds timeout in milliseconds, -1 (or
* #DBUS_TIMEOUT_USE_DEFAULT) for default or #DBUS_TIMEOUT_INFINITE
for no
* timeout
* @param error return location for error message
* @returns the message that is the reply or #NULL with an error code
if the
* function fails.
*/
DBusMessage*
dbus_connection_send_with_reply_and_block (DBusConnection
*connection,
DBusMessage
*message,
int
timeout_milliseconds,
DBusError *error)
{
    DBusMessage *reply;
    DBusPendingCall *pending;

    _dbus_return_val_if_fail (connection != NULL, NULL);
    _dbus_return_val_if_fail (message != NULL, NULL);
    _dbus_return_val_if_fail (timeout_milliseconds >= 0 ||
timeout_milliseconds == -1, NULL);
    _dbus_return_val_if_error_is_set (error, NULL);

#ifdef HAVE_UNIX_FD_PASSING

    CONNECTION_LOCK (connection);
    if (!_dbus_transport_can_pass_unix_fd(connection->transport) &&
        message->n_unix_fds > 0)
    {
        CONNECTION_UNLOCK (connection);
        dbus_set_error(error, DBUS_ERROR_FAILED, "Cannot send file
descriptors on this connection.");
        return NULL;
    }
    CONNECTION_UNLOCK (connection);

#endif

    if (!dbus_connection_send_with_reply (connection, message,
&pending,
timeout_milliseconds))
    {
        _DBUS_SET_OOM (error);
        return NULL;
    }
}

```

```

    if (pending == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_DISCONNECTED, "Connection is
closed");
        return NULL;
    }

    dbus_pending_call_block (pending);

    reply = dbus_pending_call_steal_reply (pending);
    dbus_pending_call_unref (pending);

    /* call_complete_and_unlock() called from pending_call_block()
should
    * always fill this in.
    */
    _dbus_assert (reply != NULL);

    if (dbus_set_error_from_message (error, reply))
    {
        dbus_message_unref (reply);
        return NULL;
    }
    else
        return reply;
}

/**
 * Blocks until the outgoing message queue is empty.
 * Assumes connection lock already held.
 *
 * If you call this, you MUST call update_dispatch_status afterword...
 *
 * @param connection the connection.
 */
static DBusDispatchStatus
_dbus_connection_flush_unlocked (DBusConnection *connection)
{
    /* We have to specify DBUS_ITERATION_DO_READING here because
    * otherwise we could have two apps deadlock if they are both doing
    * a flush(), and the kernel buffers fill up. This could change the
    * dispatch status.
    */
    DBusDispatchStatus status;

    HAVE_LOCK_CHECK (connection);

    while (connection->n_outgoing > 0 &&
        _dbus_connection_get_is_connected_unlocked (connection))
    {
        _dbus_verbose ("doing iteration in\n");

```

```

        HAVE_LOCK_CHECK (connection);
        _dbus_connection_do_iteration_unlocked (connection,
                                                NULL,

DBUS_ITERATION_DO_READING |

DBUS_ITERATION_DO_WRITING |

                                                DBUS_ITERATION_BLOCK,
                                                -1);
    }

    HAVE_LOCK_CHECK (connection);
    _dbus_verbose ("middle\n");
    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    HAVE_LOCK_CHECK (connection);
    return status;
}

/**
 * Blocks until the outgoing message queue is empty.
 *
 * @param connection the connection.
 */
void
dbus_connection_flush (DBusConnection *connection)
{
    /* We have to specify DBUS_ITERATION_DO_READING here because
     * otherwise we could have two apps deadlock if they are both doing
     * a flush(), and the kernel buffers fill up. This could change the
     * dispatch status.
     */
    DBusDispatchStatus status;

    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);

    status = _dbus_connection_flush_unlocked (connection);

    HAVE_LOCK_CHECK (connection);
    /* Unlocks and calls out to user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    _dbus_verbose ("end\n");
}

/**
 * This function implements dbus_connection_read_write_dispatch() and
 * dbus_connection_read_write() (they pass a different value for the
 * dispatch parameter).

```

```

*
* @param connection the connection
* @param timeout_milliseconds max time to block or -1 for infinite
* @param dispatch dispatch new messages or leave them on the incoming
queue
* @returns #TRUE if the disconnect message has not been processed
*/
static dbus_bool_t
_dbus_connection_read_write_dispatch (DBusConnection *connection,
                                     int
timeout_milliseconds,
                                     dbus_bool_t      dispatch)
{
    DBusDispatchStatus dstatus;
    dbus_bool_t progress_possible;

    /* Need to grab a ref here in case we're a private connection and
     * the user drops the last ref in a handler we call; see bug
     * https://bugs.freedesktop.org/show\_bug.cgi?id=15635
     */
    dbus_connection_ref (connection);
    dstatus = dbus_connection_get_dispatch_status (connection);

    if (dispatch && dstatus == DBUS_DISPATCH_DATA_REMAINS)
    {
        _dbus_verbose ("doing dispatch\n");
        dbus_connection_dispatch (connection);
        CONNECTION_LOCK (connection);
    }
    else if (dstatus == DBUS_DISPATCH_NEED_MEMORY)
    {
        _dbus_verbose ("pausing for memory\n");
        _dbus_memory_pause_based_on_timeout (timeout_milliseconds);
        CONNECTION_LOCK (connection);
    }
    else
    {
        CONNECTION_LOCK (connection);
        if (_dbus_connection_get_is_connected_unlocked (connection))
        {
            _dbus_verbose ("doing iteration\n");
            _dbus_connection_do_iteration_unlocked (connection,
                                                    NULL,
                                                    DBUS_ITERATION_DO_READING |
                                                    DBUS_ITERATION_DO_WRITING |
                                                    DBUS_ITERATION_BLOCK,
                                                    timeout_milliseconds);
        }
    }
}

```

```

    }

    HAVE_LOCK_CHECK (connection);
    /* If we can dispatch, we can make progress until the Disconnected
message
    * has been processed; if we can only read/write, we can make
progress
    * as long as the transport is open.
    */
    if (dispatch)
        progress_possible = connection->n_incoming != 0 ||
            connection->disconnect_message_link != NULL;
    else
        progress_possible = _dbus_connection_get_is_connected_unlocked
(connection);

    CONNECTION_UNLOCK (connection);

    dbus_connection_unref (connection);

    return progress_possible; /* TRUE if we can make more progress */
}

/**
 * This function is intended for use with applications that don't want
 * to write a main loop and deal with #DBusWatch and #DBusTimeout. An
 * example usage would be:
 *
 * @code
 *     while (dbus_connection_read_write_dispatch (connection, -1))
 *         ; // empty loop body
 * @endcode
 *
 * In this usage you would normally have set up a filter function to
look
 * at each message as it is dispatched. The loop terminates when the
last
 * message from the connection (the disconnected signal) is processed.
 *
 * If there are messages to dispatch, this function will
 * dbus_connection_dispatch() once, and return. If there are no
 * messages to dispatch, this function will block until it can read or
 * write, then read or write, then return.
 *
 * The way to think of this function is that it either makes some sort
 * of progress, or it blocks. Note that, while it is blocked on I/O,
it
 * cannot be interrupted (even by other threads), which makes this
function
 * unsuitable for applications that do more than just react to
received

```



```

* messages.
*
* The return value indicates whether the disconnect message has been
* processed, NOT whether the connection is connected. This is
* important because even after disconnecting, you want to process any
* messages you received prior to the disconnect.
*
* @param connection the connection
* @param timeout_milliseconds max time to block or -1 for infinite
* @returns #TRUE if the disconnect message has not been processed
*/
dbus_bool_t
dbus_connection_read_write_dispatch (DBusConnection *connection,
                                     int
                                     timeout_milliseconds)
{
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (timeout_milliseconds >= 0 ||
    timeout_milliseconds == -1, FALSE);
    return _dbus_connection_read_write_dispatch(connection,
    timeout_milliseconds, TRUE);
}

/**
 * This function is intended for use with applications that don't want
to
 * write a main loop and deal with #DBusWatch and #DBusTimeout. See
also
 * dbus_connection_read_write_dispatch().
 *
 * As long as the connection is open, this function will block until
it can
 * read or write, then read or write, then return #TRUE.
 *
 * If the connection is closed, the function returns #FALSE.
 *
 * The return value indicates whether reading or writing is still
 * possible, i.e. whether the connection is connected.
 *
 * Note that even after disconnection, messages may remain in the
 * incoming queue that need to be
 * processed. dbus_connection_read_write_dispatch() dispatches
 * incoming messages for you; with dbus_connection_read_write() you
 * have to arrange to drain the incoming queue yourself.
 *
 * @param connection the connection
 * @param timeout_milliseconds max time to block or -1 for infinite
 * @returns #TRUE if still connected
 */
dbus_bool_t
dbus_connection_read_write (DBusConnection *connection,
                             int
                             timeout_milliseconds)

```

```

{
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (timeout_milliseconds >= 0 ||
timeout_milliseconds == -1, FALSE);
    return _dbus_connection_read_write_dispatch(connection,
timeout_milliseconds, FALSE);
}

/* We need to call this anytime we pop the head of the queue, and then
 * update_dispatch_status_and_unlock needs to be called afterward
 * which will "process" the disconnected message and set
 * disconnected_message_processed.
 */
static void
check_disconnected_message_arrived_unlocked (DBusConnection
*connection,
                                             DBusMessage
*head_of_queue)
{
    HAVE_LOCK_CHECK (connection);

    /* checking that the link is NULL is an optimization to avoid the
is_signal call */
    if (connection->disconnect_message_link == NULL &&
        dbus_message_is_signal (head_of_queue,
                                DBUS_INTERFACE_LOCAL,
                                "Disconnected"))
    {
        connection->disconnected_message_arrived = TRUE;
    }
}

/**
 * Returns the first-received message from the incoming message queue,
 * leaving it in the queue. If the queue is empty, returns #NULL.
 *
 * The caller does not own a reference to the returned message, and
 * must either return it using dbus_connection_return_message() or
 * keep it after calling dbus_connection_steal_borrowed_message(). No
 * one can get at the message while its borrowed, so return it as
 * quickly as possible and don't keep a reference to it after
 * returning it. If you need to keep the message, make a copy of it.
 *
 * dbus_connection_dispatch() will block if called while a borrowed
 * message is outstanding; only one piece of code can be playing with
 * the incoming queue at a time. This function will block if called
 * during a dbus_connection_dispatch().
 *
 * @param connection the connection.
 * @returns next message in the incoming queue.
 */
DBusMessage*

```

```

dbus_connection_borrow_message (DBusConnection *connection)
{
    DBusDispatchStatus status;
    DBusMessage *message;

    _dbus_return_val_if_fail (connection != NULL, NULL);

    _dbus_verbose ("start\n");

    /* this is called for the side effect that it queues
     * up any messages from the transport
     */
    status = dbus_connection_get_dispatch_status (connection);
    if (status != DBUS_DISPATCH_DATA_REMAINS)
        return NULL;

    CONNECTION_LOCK (connection);

    _dbus_connection_acquire_dispatch (connection);

    /* While a message is outstanding, the dispatch lock is held */
    _dbus_assert (connection->message_borrowed == NULL);

    connection->message_borrowed = _dbus_list_get_first (&connection-
>incoming_messages);

    message = connection->message_borrowed;

    check_disconnected_message_arrived_unlocked (connection, message);

    /* Note that we KEEP the dispatch lock until the message is returned
    */
    if (message == NULL)
        _dbus_connection_release_dispatch (connection);

    CONNECTION_UNLOCK (connection);

    _dbus_message_trace_ref (message, -1, -1,
"dbus_connection_borrow_message");

    /* We don't update dispatch status until it's returned or stolen */

    return message;
}

/**
 * Used to return a message after peeking at it using
 * dbus_connection_borrow_message(). Only called if
 * message from dbus_connection_borrow_message() was non-#NULL.
 *
 * @param connection the connection
 * @param message the message from dbus_connection_borrow_message()

```

```

*/
void
dbus_connection_return_message (DBusConnection *connection,
                               DBusMessage     *message)
{
    DBusDispatchStatus status;

    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (message == connection->message_borrowed);
    _dbus_return_if_fail (connection->dispatch_acquired);

    CONNECTION_LOCK (connection);

    _dbus_assert (message == connection->message_borrowed);

    connection->message_borrowed = NULL;

    _dbus_connection_release_dispatch (connection);

    status = _dbus_connection_get_dispatch_status_unlocked (connection);
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    _dbus_message_trace_ref (message, -1, -1,
"dbus_connection_return_message");
}

/**
 * Used to keep a message after peeking at it using
 * dbus_connection_borrow_message(). Before using this function, see
 * the caveats/warnings in the documentation for
 * dbus_connection_pop_message().
 *
 * @param connection the connection
 * @param message the message from dbus_connection_borrow_message()
 */
void
dbus_connection_steal_borrowed_message (DBusConnection *connection,
                                       DBusMessage     *message)
{
    DBusMessage *pop_message;
    DBusDispatchStatus status;

    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (message == connection->message_borrowed);
    _dbus_return_if_fail (connection->dispatch_acquired);

    CONNECTION_LOCK (connection);

    _dbus_assert (message == connection->message_borrowed);

```

```

pop_message = _dbus_list_pop_first (&connection->incoming_messages);
_dbus_assert (message == pop_message);
(void) pop_message; /* unused unless asserting */

connection->n_incoming -= 1;

_dbus_verbose ("Incoming message %p stolen from queue, %d
incoming\n",
              message, connection->n_incoming);

connection->message_borrowed = NULL;

_dbus_connection_release_dispatch (connection);

status = _dbus_connection_get_dispatch_status_unlocked (connection);
_dbus_connection_update_dispatch_status_and_unlock (connection,
status);
_dbus_message_trace_ref (message, -1, -1,
                        "dbus_connection_steal_borrowed_message");
}

/* See dbus_connection_pop_message, but requires the caller to own
 * the lock before calling. May drop the lock while running.
 */
static DBusList*
_dbus_connection_pop_message_link_unlocked (DBusConnection
*connection)
{
    HAVE_LOCK_CHECK (connection);

    _dbus_assert (connection->message_borrowed == NULL);

    if (connection->n_incoming > 0)
    {
        DBusList *link;

        link = _dbus_list_pop_first_link (&connection-
>incoming_messages);
        connection->n_incoming -= 1;

        _dbus_verbose ("Message %p (%s %s %s %s '%s') removed from
incoming queue %p, %d incoming\n",
                      link->data,
                      dbus_message_type_to_string
(dbus_message_get_type (link->data)),
                      dbus_message_get_path (link->data) ?
dbus_message_get_path (link->data) :
                      "no path",
                      dbus_message_get_interface (link->data) ?
dbus_message_get_interface (link->data) :
                      "no interface",

```

```

        dbus_message_get_member (link->data) ?
        dbus_message_get_member (link->data) :
        "no member",
        dbus_message_get_signature (link->data),
        connection, connection->n_incoming);

    _dbus_message_trace_ref (link->data, -1, -1,
        "_dbus_connection_pop_message_link_unlocked");

    check_disconnected_message_arrived_unlocked (connection, link->data);

    return link;
}
else
    return NULL;
}

/* See dbus_connection_pop_message, but requires the caller to own
 * the lock before calling. May drop the lock while running.
 */
static DBusMessage*
_dbus_connection_pop_message_unlocked (DBusConnection *connection)
{
    DBusList *link;

    HAVE_LOCK_CHECK (connection);

    link = _dbus_connection_pop_message_link_unlocked (connection);

    if (link != NULL)
    {
        DBusMessage *message;

        message = link->data;

        _dbus_list_free_link (link);

        return message;
    }
    else
        return NULL;
}

static void
_dbus_connection_putback_message_link_unlocked (DBusConnection
*connection,
                                                DBusList
*message_link)
{
    HAVE_LOCK_CHECK (connection);

```

```

    _dbus_assert (message_link != NULL);
    /* You can't borrow a message while a link is outstanding */
    _dbus_assert (connection->message_borrowed == NULL);
    /* We had to have the dispatch lock across the pop/putback */
    _dbus_assert (connection->dispatch_acquired);

    _dbus_list_prepend_link (&connection->incoming_messages,
                             message_link);
    connection->n_incoming += 1;

    _dbus_verbose ("Message %p (%s %s %s '%s') put back into queue %p,
%d incoming\n",
                  message_link->data,
                  dbus_message_type_to_string (dbus_message_get_type
(message_link->data)),
                  dbus_message_get_interface (message_link->data) ?
dbus_message_get_interface (message_link->data) :
                  "no interface",
                  dbus_message_get_member (message_link->data) ?
dbus_message_get_member (message_link->data) :
                  "no member",
                  dbus_message_get_signature (message_link->data),
                  connection, connection->n_incoming);

    _dbus_message_trace_ref (message_link->data, -1, -1,
                             "_dbus_connection_putback_message_link_unlocked");
}

/**
 * Returns the first-received message from the incoming message queue,
 * removing it from the queue. The caller owns a reference to the
 * returned message. If the queue is empty, returns #NULL.
 *
 * This function bypasses any message handlers that are registered,
 * and so using it is usually wrong. Instead, let the main loop invoke
 * dbus_connection_dispatch(). Popping messages manually is only
 * useful in very simple programs that don't share a #DBusConnection
 * with any libraries or other modules.
 *
 * There is a lock that covers all ways of accessing the incoming
message
 * queue, so dbus_connection_dispatch(),
dbus_connection_pop_message(),
 * dbus_connection_borrow_message(), etc. will all block while one of
the others
 * in the group is running.
 *
 * @param connection the connection.
 * @returns next message in the incoming queue.
 */
DBusMessage*
dbus_connection_pop_message (DBusConnection *connection)

```

```

{
    DBusMessage *message;
    DBusDispatchStatus status;

    _dbus_verbose ("start\n");

    /* this is called for the side effect that it queues
     * up any messages from the transport
     */
    status = dbus_connection_get_dispatch_status (connection);
    if (status != DBUS_DISPATCH_DATA_REMAINS)
        return NULL;

    CONNECTION_LOCK (connection);
    _dbus_connection_acquire_dispatch (connection);
    HAVE_LOCK_CHECK (connection);

    message = _dbus_connection_pop_message_unlocked (connection);

    _dbus_verbose ("Returning popped message %p\n", message);

    _dbus_connection_release_dispatch (connection);

    status = _dbus_connection_get_dispatch_status_unlocked (connection);
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    return message;
}

/**
 * Acquire the dispatcher. This is a separate lock so the main
 * connection lock can be dropped to call out to application dispatch
 * handlers.
 *
 * @param connection the connection.
 */
static void
_dbus_connection_acquire_dispatch (DBusConnection *connection)
{
    HAVE_LOCK_CHECK (connection);

    _dbus_connection_ref_unlocked (connection);
    CONNECTION_UNLOCK (connection);

    _dbus_verbose ("locking dispatch_mutex\n");
    _dbus_cmutex_lock (connection->dispatch_mutex);

    while (connection->dispatch_acquired)
    {
        _dbus_verbose ("waiting for dispatch to be acquirable\n");
        _dbus_condvar_wait (connection->dispatch_cond,

```



```

        connection->dispatch_mutex);
    }

    _dbus_assert (!connection->dispatch_acquired);

    connection->dispatch_acquired = TRUE;

    _dbus_verbose ("unlocking dispatch_mutex\n");
    _dbus_cmutex_unlock (connection->dispatch_mutex);

    CONNECTION_LOCK (connection);
    _dbus_connection_unref_unlocked (connection);
}

/**
 * Release the dispatcher when you're done with it. Only call
 * after you've acquired the dispatcher. Wakes up at most one
 * thread currently waiting to acquire the dispatcher.
 *
 * @param connection the connection.
 */
static void
_dbus_connection_release_dispatch (DBusConnection *connection)
{
    HAVE_LOCK_CHECK (connection);

    _dbus_verbose ("locking dispatch_mutex\n");
    _dbus_cmutex_lock (connection->dispatch_mutex);

    _dbus_assert (connection->dispatch_acquired);

    connection->dispatch_acquired = FALSE;
    _dbus_condvar_wake_one (connection->dispatch_cond);

    _dbus_verbose ("unlocking dispatch_mutex\n");
    _dbus_cmutex_unlock (connection->dispatch_mutex);
}

static void
_dbus_connection_failed_pop (DBusConnection *connection,
                             DBusList      *message_link)
{
    _dbus_list_prepend_link (&connection->incoming_messages,
                             message_link);
    connection->n_incoming += 1;
}

/* Note this may be called multiple times since we don't track whether
we already did it */
static void
notify_disconnected_unlocked (DBusConnection *connection)
{

```

```

HAVE_LOCK_CHECK (connection);

/* Set the weakref in dbus-bus.c to NULL, so nobody will get a
disconnected
 * connection from dbus_bus_get(). We make the same guarantee for
 * dbus_connection_open() but in a different way since we don't want
to
 * unref right here; we instead check for connectedness before
returning
 * the connection from the hash.
 */
_dbus_bus_notify_shared_connection_disconnected_unlocked
(connection);

/* Dump the outgoing queue, we aren't going to be able to
 * send it now, and we'd like accessors like
 * dbus_connection_get_outgoing_size() to be accurate.
 */
if (connection->n_outgoing > 0)
{
    DBusList *link;

    _dbus_verbose ("Dropping %d outgoing messages since we're
disconnected\n",
                  connection->n_outgoing);

    while ((link = _dbus_list_get_last_link (&connection-
>outgoing_messages))
           {
                _dbus_connection_message_sent_unlocked (connection, link-
>data);
            }
    }
}

/* Note this may be called multiple times since we don't track whether
we already did it */
static DBusDispatchStatus
notify_disconnected_and_dispatch_complete_unlocked (DBusConnection
*connection)
{
    HAVE_LOCK_CHECK (connection);

    if (connection->disconnect_message_link != NULL)
    {
        _dbus_verbose ("Sending disconnect message\n");

        /* If we have pending calls, queue their timeouts - we want the
Disconnected
 * to be the last message, after these timeouts.
 */

```

```

        connection_timeout_and_complete_all_pending_calls_unlocked
(connection);

        /* We haven't sent the disconnect message already,
         * and all real messages have been queued up.
         */
        _dbus_connection_queue_synthesized_message_link (connection,
                                                         connection-
>disconnect_message_link);
        connection->disconnect_message_link = NULL;

        return DBUS_DISPATCH_DATA_REMAINS;
    }

    return DBUS_DISPATCH_COMPLETE;
}

static DBusDispatchStatus
_dbus_connection_get_dispatch_status_unlocked (DBusConnection
*connection)
{
    HAVE_LOCK_CHECK (connection);

    if (connection->n_incoming > 0)
        return DBUS_DISPATCH_DATA_REMAINS;
    else if (!_dbus_transport_queue_messages (connection->transport))
        return DBUS_DISPATCH_NEED_MEMORY;
    else
    {
        DBusDispatchStatus status;
        dbus_bool_t is_connected;

        status = _dbus_transport_get_dispatch_status (connection-
>transport);
        is_connected = _dbus_transport_get_is_connected (connection-
>transport);

        _dbus_verbose ("dispatch status = %s is_connected = %d\n",
                      DISPATCH_STATUS_NAME (status), is_connected);

        if (!is_connected)
        {
            /* It's possible this would be better done by having an
explicit
            * notification from _dbus_transport_disconnect() that would
            * synchronously do this, instead of waiting for the next
dispatch
            * status check. However, probably not good to change until
it causes
            * a problem.
            */
            notify_disconnected_unlocked (connection);

```

```

        /* I'm not sure this is needed; the idea is that we want to
        * queue the Disconnected only after we've read all the
        * messages, but if we're disconnected maybe we are
guaranteed
        * to have read them all ?
        */
        if (status == DBUS_DISPATCH_COMPLETE)
            status =
notify_disconnected_and_dispatch_complete_unlocked (connection);
    }

    if (status != DBUS_DISPATCH_COMPLETE)
        return status;
    else if (connection->n_incoming > 0)
        return DBUS_DISPATCH_DATA_REMAINS;
    else
        return DBUS_DISPATCH_COMPLETE;
}
}

static void
_dbus_connection_update_dispatch_status_and_unlock (DBusConnection
*connection,
                                                    DBusDispatchStatus
new_status)
{
    dbus_bool_t changed;
    DBusDispatchStatusFunction function;
    void *data;

    HAVE_LOCK_CHECK (connection);

    _dbus_connection_ref_unlocked (connection);

    changed = new_status != connection->last_dispatch_status;

    connection->last_dispatch_status = new_status;

    function = connection->dispatch_status_function;
    data = connection->dispatch_status_data;

    if (connection->disconnected_message_arrived &&
        !connection->disconnected_message_processed)
    {
        connection->disconnected_message_processed = TRUE;

        /* this does an unref, but we have a ref
        * so we should not run the finalizer here
        * inside the lock.
        */
        connection_forget_shared_unlocked (connection);
    }
}

```

```

    if (connection->exit_on_disconnect)
    {
        CONNECTION_UNLOCK (connection);

        _dbus_verbose ("Exiting on Disconnected signal\n");
        _dbus_exit (1);
        _dbus_assert_not_reached ("Call to exit() returned");
    }
}

/* We drop the lock */
CONNECTION_UNLOCK (connection);

if (changed && function)
{
    _dbus_verbose ("Notifying of change to dispatch status of %p now
%d (%s)\n",
                  connection, new_status,
                  DISPATCH_STATUS_NAME (new_status));
    (* function) (connection, new_status, data);
}

dbus_connection_unref (connection);
}

/**
 * Gets the current state of the incoming message queue.
 * #DBUS_DISPATCH_DATA_REMAINS indicates that the message queue
 * may contain messages. #DBUS_DISPATCH_COMPLETE indicates that the
 * incoming queue is empty. #DBUS_DISPATCH_NEED_MEMORY indicates that
 * there could be data, but we can't know for sure without more
 * memory.
 *
 * To process the incoming message queue, use
dbus_connection_dispatch()
 * or (in rare cases) dbus_connection_pop_message().
 *
 * Note, #DBUS_DISPATCH_DATA_REMAINS really means that either we
 * have messages in the queue, or we have raw bytes buffered up
 * that need to be parsed. When these bytes are parsed, they
 * may not add up to an entire message. Thus, it's possible
 * to see a status of #DBUS_DISPATCH_DATA_REMAINS but not
 * have a message yet.
 *
 * In particular this happens on initial connection, because all sorts
 * of authentication protocol stuff has to be parsed before the
 * first message arrives.
 *
 * @param connection the connection.
 * @returns current dispatch status
 */

```

```

DBusDispatchStatus
dbus_connection_get_dispatch_status (DBusConnection *connection)
{
    DBusDispatchStatus status;

    _dbus_return_val_if_fail (connection != NULL,
DBUS_DISPATCH_COMPLETE);

    _dbus_verbose ("start\n");

    CONNECTION_LOCK (connection);

    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    CONNECTION_UNLOCK (connection);

    return status;
}

/**
 * Filter function for handling the Peer standard interface.
 */
static DBusHandlerResult
_dbus_connection_peer_filter_unlocked_no_update (DBusConnection
*connection,
                                                DBusMessage
*message)
{
    dbus_bool_t sent = FALSE;
    DBusMessage *ret = NULL;
    DBusList *expire_link;

    if (connection->route_peer_messages && dbus_message_get_destination
(message) != NULL)
    {
        /* This means we're letting the bus route this message */
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
    }

    if (!dbus_message_has_interface (message, DBUS_INTERFACE_PEER))
    {
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
    }

    /* Preallocate a linked-list link, so that if we need to dispose of
a
 * message, we can attach it to the expired list */
    expire_link = _dbus_list_alloc_link (NULL);

    if (!expire_link)
        return DBUS_HANDLER_RESULT_NEED_MEMORY;
}

```

```

if (dbus_message_is_method_call (message,
                                DBUS_INTERFACE_PEER,
                                "Ping"))
{
    ret = dbus_message_new_method_return (message);
    if (ret == NULL)
        goto out;

    sent = _dbus_connection_send_unlocked_no_update (connection,
ret, NULL);
}
else if (dbus_message_is_method_call (message,
                                DBUS_INTERFACE_PEER,
                                "GetMachineId"))
{
    DBusString uuid;

    ret = dbus_message_new_method_return (message);
    if (ret == NULL)
        goto out;

    _dbus_string_init (&uuid);
    if (_dbus_get_local_machine_uuid_encoded (&uuid))
    {
        const char *v_STRING = _dbus_string_get_const_data (&uuid);
        if (dbus_message_append_args (ret,
                                DBUS_TYPE_STRING, &v_STRING,
                                DBUS_TYPE_INVALID))
        {
            sent = _dbus_connection_send_unlocked_no_update
(connection, ret, NULL);
        }
        _dbus_string_free (&uuid);
    }
    else
    {
        /* We need to bounce anything else with this interface,
otherwise apps
* could start extending the interface and when we added
extensions
* here to DBusConnection we'd break those apps.
*/
        ret = dbus_message_new_error (message,
                                DBUS_ERROR_UNKNOWN_METHOD,
                                "Unknown method invoked on
org.freedesktop.DBus.Peer interface");
        if (ret == NULL)
            goto out;

        sent = _dbus_connection_send_unlocked_no_update (connection,
ret, NULL);

```

```

    }

out:
    if (ret == NULL)
    {
        _dbus_list_free_link (expire_link);
    }
    else
    {
        /* It'll be safe to unref the reply when we unlock */
        expire_link->data = ret;
        _dbus_list_prepend_link (&connection->expired_messages,
expire_link);
    }

    if (!sent)
        return DBUS_HANDLER_RESULT_NEED_MEMORY;

    return DBUS_HANDLER_RESULT_HANDLED;
}

/**
 * Processes all builtin filter functions
 *
 * If the spec specifies a standard interface
 * they should be processed from this method
 */
static DBusHandlerResult
_dbus_connection_run_builtin_filters_unlocked_no_update
(DBusConnection *connection,
                                     DBusMessage
*message)
{
    /* We just run one filter for now but have the option to run more
       if the spec calls for it in the future */

    return _dbus_connection_peer_filter_unlocked_no_update (connection,
message);
}

/**
 * Processes any incoming data.
 *
 * If there's incoming raw data that has not yet been parsed, it is
 * parsed, which may or may not result in adding messages to the
 * incoming queue.
 *
 * The incoming data buffer is filled when the connection reads from
 * its underlying transport (such as a socket). Reading usually
 * happens in dbus_watch_handle() or dbus_connection_read_write().
 *
 * If there are complete messages in the incoming queue,

```



```

* dbus_connection_dispatch() removes one message from the queue and
* processes it. Processing has three steps.
*
* First, any method replies are passed to #DBusPendingCall or
* dbus_connection_send_with_reply_and_block() in order to
* complete the pending method call.
*
* Second, any filters registered with dbus_connection_add_filter()
* are run. If any filter returns #DBUS_HANDLER_RESULT_HANDLED
* then processing stops after that filter.
*
* Third, if the message is a method call it is forwarded to
* any registered object path handlers added with
* dbus_connection_register_object_path() or
* dbus_connection_register_fallback().
*
* A single call to dbus_connection_dispatch() will process at most
* one message; it will not clear the entire message queue.
*
* Be careful about calling dbus_connection_dispatch() from inside a
* message handler, i.e. calling dbus_connection_dispatch()
* recursively. If threads have been initialized with a recursive
* mutex function, then this will not deadlock; however, it can
* certainly confuse your application.
*
* @todo some FIXME in here about handling
DBUS_HANDLER_RESULT_NEED_MEMORY
*
* @param connection the connection
* @returns dispatch status, see dbus_connection_get_dispatch_status()
*/
DBusDispatchStatus
dbus_connection_dispatch (DBusConnection *connection)
{
    DBusMessage *message;
    DBusList *link, *filter_list_copy, *message_link;
    DBusHandlerResult result;
    DBusPendingCall *pending;
    dbus_int32_t reply_serial;
    DBusDispatchStatus status;
    dbus_bool_t found_object;

    _dbus_return_val_if_fail (connection != NULL,
DBUS_DISPATCH_COMPLETE);

    _dbus_verbose ("\n");

    CONNECTION_LOCK (connection);
    status = _dbus_connection_get_dispatch_status_unlocked (connection);
    if (status != DBUS_DISPATCH_DATA_REMAINS)
    {
        /* unlocks and calls out to user code */

```

```

        _dbus_connection_update_dispatch_status_and_unlock (connection,
status);
        return status;
    }

    /* We need to ref the connection since the callback could
potentially
    * drop the last ref to it
    */
    _dbus_connection_ref_unlocked (connection);

    _dbus_connection_acquire_dispatch (connection);
HAVE_LOCK_CHECK (connection);

    message_link = _dbus_connection_pop_message_link_unlocked
(connection);
    if (message_link == NULL)
    {
        /* another thread dispatched our stuff */

        _dbus_verbose ("another thread dispatched message (during
acquire_dispatch above)\n");

        _dbus_connection_release_dispatch (connection);

        status = _dbus_connection_get_dispatch_status_unlocked
(connection);

        _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

        dbus_connection_unref (connection);

        return status;
    }

    message = message_link->data;

    _dbus_verbose (" dispatching message %p (%s %s %s '%s')\n",
message,
dbus_message_type_to_string (dbus_message_get_type
(message)),
dbus_message_get_interface (message) ?
dbus_message_get_interface (message) :
"no interface",
dbus_message_get_member (message) ?
dbus_message_get_member (message) :
"no member",
dbus_message_get_signature (message));

    result = DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

```

```

/* Pending call handling must be first, because if you do
 * dbus_connection_send_with_reply_and_block() or
 * dbus_pending_call_block() then no handlers/filters will be run on
 * the reply. We want consistent semantics in the case where we
 * dbus_connection_dispatch() the reply.
 */

reply_serial = dbus_message_get_reply_serial (message);
pending = _dbus_hash_table_lookup_int (connection->pending_replies,
                                     reply_serial);

if (pending)
{
    _dbus_verbose ("Dispatching a pending reply\n");
    complete_pending_call_and_unlock (connection, pending, message);
    pending = NULL; /* it's probably unref'd */

    CONNECTION_LOCK (connection);
    _dbus_verbose ("pending call completed in dispatch\n");
    result = DBUS_HANDLER_RESULT_HANDLED;
    goto out;
}

result = _dbus_connection_run_builtin_filters_unlocked_no_update
(connection, message);
if (result != DBUS_HANDLER_RESULT_NOT_YET_HANDLED)
    goto out;

if (!_dbus_list_copy (&connection->filter_list, &filter_list_copy))
{
    _dbus_connection_release_dispatch (connection);
    HAVE_LOCK_CHECK (connection);

    _dbus_connection_failed_pop (connection, message_link);

    /* unlocks and calls user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
DBUS_DISPATCH_NEED_MEMORY);
    dbus_connection_unref (connection);

    return DBUS_DISPATCH_NEED_MEMORY;
}

_dbus_list_foreach (&filter_list_copy,
                  (DBusForeachFunction)_dbus_message_filter_ref,
                  NULL);

/* We're still protected from dispatch() reentrancy here
 * since we acquired the dispatcher
 */
CONNECTION_UNLOCK (connection);

```

```

link = _dbus_list_get_first_link (&filter_list_copy);
while (link != NULL)
{
    DBusMessageFilter *filter = link->data;
    DBusList *next = _dbus_list_get_next_link (&filter_list_copy,
link);

    if (filter->function == NULL)
    {
        _dbus_verbose (" filter was removed in a callback
function\n");
        link = next;
        continue;
    }

    _dbus_verbose (" running filter on message %p\n", message);
    result = (* filter->function) (connection, message, filter-
>user_data);

    if (result != DBUS_HANDLER_RESULT_NOT_YET_HANDLED)
        break;

    link = next;
}

_dbus_list_foreach (&filter_list_copy,
                    (DBusForeachFunction) _dbus_message_filter_unref,
                    NULL);
_dbus_list_clear (&filter_list_copy);

CONNECTION_LOCK (connection);

if (result == DBUS_HANDLER_RESULT_NEED_MEMORY)
{
    _dbus_verbose ("No memory\n");
    goto out;
}
else if (result == DBUS_HANDLER_RESULT_HANDLED)
{
    _dbus_verbose ("filter handled message in dispatch\n");
    goto out;
}

/* We're still protected from dispatch() reentrancy here
 * since we acquired the dispatcher
 */
_dbus_verbose (" running object path dispatch on message %p (%s %s
%s '%s')\n",
                message,
                dbus_message_type_to_string (dbus_message_get_type
(message)),
                dbus_message_get_interface (message) ?

```

```

        dbus_message_get_interface (message) :
        "no interface",
        dbus_message_get_member (message) ?
        dbus_message_get_member (message) :
        "no member",
        dbus_message_get_signature (message));

HAVE_LOCK_CHECK (connection);
result = _dbus_object_tree_dispatch_and_unlock (connection->objects,
                                                message,
                                                &found_object);

CONNECTION_LOCK (connection);

if (result != DBUS_HANDLER_RESULT_NOT_YET_HANDLED)
{
    _dbus_verbose ("object tree handled message in dispatch\n");
    goto out;
}

if (dbus_message_get_type (message) ==
    DBUS_MESSAGE_TYPE_METHOD_CALL)
{
    DBusMessage *reply;
    DBusString str;
    DBusPreallocatedSend *preallocated;
    DBusList *expire_link;

    _dbus_verbose (" sending error %s\n",
                  DBUS_ERROR_UNKNOWN_METHOD);

    if (!_dbus_string_init (&str))
    {
        result = DBUS_HANDLER_RESULT_NEED_MEMORY;
        _dbus_verbose ("no memory for error string in dispatch\n");
        goto out;
    }

    if (!_dbus_string_append_printf (&str,
                                     "Method \"%s\" with signature\n"
                                     "\"%s\" on interface \"%s\" doesn't exist\n",
                                     dbus_message_get_member
(message),
                                     dbus_message_get_signature
(message),
                                     dbus_message_get_interface
(message)))
    {
        _dbus_string_free (&str);
        result = DBUS_HANDLER_RESULT_NEED_MEMORY;
        _dbus_verbose ("no memory for error string in dispatch\n");
        goto out;
    }
}

```

```

    }

    reply = dbus_message_new_error (message,
                                   found_object ?
DBUS_ERROR_UNKNOWN_METHOD : DBUS_ERROR_UNKNOWN_OBJECT,
                                   _dbus_string_get_const_data
(&str));
    _dbus_string_free (&str);

    if (reply == NULL)
    {
        result = DBUS_HANDLER_RESULT_NEED_MEMORY;
        _dbus_verbose ("no memory for error reply in dispatch\n");
        goto out;
    }

    expire_link = _dbus_list_alloc_link (reply);

    if (expire_link == NULL)
    {
        dbus_message_unref (reply);
        result = DBUS_HANDLER_RESULT_NEED_MEMORY;
        _dbus_verbose ("no memory for error send in dispatch\n");
        goto out;
    }

    preallocated = _dbus_connection_preallocate_send_unlocked
(connection);

    if (preallocated == NULL)
    {
        _dbus_list_free_link (expire_link);
        /* It's OK that this is finalized, because it hasn't been
seen by
        * anything that could attach user callbacks */
        dbus_message_unref (reply);
        result = DBUS_HANDLER_RESULT_NEED_MEMORY;
        _dbus_verbose ("no memory for error send in dispatch\n");
        goto out;
    }

    _dbus_connection_send_preallocated_unlocked_no_update
(connection, preallocated,
                                                    reply,
NULL);
    /* reply will be freed when we release the lock */
    _dbus_list_prepend_link (&connection->expired_messages,
expire_link);

    result = DBUS_HANDLER_RESULT_HANDLED;
}

```

```

_dbus_verbose (" done dispatching %p (%s %s %s '%s') on connection
%p\n", message,
              dbus_message_type_to_string (dbus_message_get_type
(message)),
              dbus_message_get_interface (message) ?
dbus_message_get_interface (message) :
              "no interface",
              dbus_message_get_member (message) ?
dbus_message_get_member (message) :
              "no member",
              dbus_message_get_signature (message),
              connection);

out:
if (result == DBUS_HANDLER_RESULT_NEED_MEMORY)
{
    _dbus_verbose ("out of memory\n");

    /* Put message back, and we'll start over.
     * Yes this means handlers must be idempotent if they
     * don't return HANDLED; c'est la vie.
     */
    _dbus_connection_putback_message_link_unlocked (connection,
                                                    message_link);

    /* now we don't want to free them */
    message_link = NULL;
    message = NULL;
}
else
{
    _dbus_verbose (" ... done dispatching\n");
}

_dbus_connection_release_dispatch (connection);
HAVE_LOCK_CHECK (connection);

if (message != NULL)
{
    /* We don't want this message to count in maximum message limits
when
    * computing the dispatch status, below. We have to drop the
lock
    * temporarily, because finalizing a message can trigger
callbacks.
    *
    * We have a reference to the connection, and we don't use any
cached
    * pointers to the connection's internals below this point, so
it should
    * be safe to drop the lock and take it back. */
    CONNECTION_UNLOCK (connection);
    dbus_message_unref (message);
}

```

```

        CONNECTION_LOCK (connection);
    }

    if (message_link != NULL)
        _dbus_list_free_link (message_link);

    _dbus_verbose ("before final status update\n");
    status = _dbus_connection_get_dispatch_status_unlocked (connection);

    /* unlocks and calls user code */
    _dbus_connection_update_dispatch_status_and_unlock (connection,
status);

    dbus_connection_unref (connection);

    return status;
}

/**
 * Sets the watch functions for the connection. These functions are
 * responsible for making the application's main loop aware of file
 * descriptors that need to be monitored for events, using select() or
 * poll(). When using Qt, typically the DBusAddWatchFunction would
 * create a QSocketNotifier. When using GLib, the DBusAddWatchFunction
 * could call g_io_add_watch(), or could be used as part of a more
 * elaborate GSource. Note that when a watch is added, it may
 * not be enabled.
 *
 * The DBusWatchToggledFunction notifies the application that the
 * watch has been enabled or disabled. Call dbus_watch_get_enabled()
 * to check this. A disabled watch should have no effect, and enabled
 * watch should be added to the main loop. This feature is used
 * instead of simply adding/removing the watch because
 * enabling/disabling can be done without memory allocation. The
 * toggled function may be NULL if a main loop re-queries
 * dbus_watch_get_enabled() every time anyway.
 *
 * The DBusWatch can be queried for the file descriptor to watch using
 * dbus_watch_get_unix_fd() or dbus_watch_get_socket(), and for the
 * events to watch for using dbus_watch_get_flags(). The flags
 * returned by dbus_watch_get_flags() will only contain
 * DBUS_WATCH_READABLE and DBUS_WATCH_WRITABLE, never
 * DBUS_WATCH_HANGUP or DBUS_WATCH_ERROR; all watches implicitly
 * include a watch for hangups, errors, and other exceptional
 * conditions.
 *
 * Once a file descriptor becomes readable or writable, or an
exception
 * occurs, dbus_watch_handle() should be called to
 * notify the connection of the file descriptor's condition.
 *
 * dbus_watch_handle() cannot be called during the

```



```

* DbusAddWatchFunction, as the connection will not be ready to handle
* that watch yet.
*
* It is not allowed to reference a DbusWatch after it has been passed
* to remove_function.
*
* If #FALSE is returned due to lack of memory, the failure may be due
* to a #FALSE return from the new add_function. If so, the
* add_function may have been called successfully one or more times,
* but the remove_function will also have been called to remove any
* successful adds. i.e. if #FALSE is returned the net result
* should be that dbus_connection_set_watch_functions() has no effect,
* but the add_function and remove_function may have been called.
*
* @note The thread lock on DbusConnection is held while
* watch functions are invoked, so inside these functions you
* may not invoke any methods on DbusConnection or it will deadlock.
* See the comments in the code or
http://lists.freedesktop.org/archives/dbus/2007-July/tread.html#8144
* if you encounter this issue and want to attempt writing a patch.
*
* @param connection the connection.
* @param add_function function to begin monitoring a new descriptor.
* @param remove_function function to stop monitoring a descriptor.
* @param toggled_function function to notify of enable/disable
* @param data data to pass to add_function and remove_function.
* @param free_data_function function to be called to free the data.
* @returns #FALSE on failure (no memory)
*/
dbus_bool_t
dbus_connection_set_watch_functions (DBusConnection
*connection,
                                DbusAddWatchFunction
add_function,
                                DbusRemoveWatchFunction
remove_function,
                                DbusWatchToggledFunction
toggled_function,
                                void
*data,
                                DbusFreeFunction
free_data_function)
{
    dbus_bool_t retval;

    _dbus_return_val_if_fail (connection != NULL, FALSE);

    CONNECTION_LOCK (connection);

    retval = _dbus_watch_list_set_functions (connection->watches,
add_function,
remove_function,

```

```

        toggled_function,
        data, free_data_function);

    CONNECTION_UNLOCK (connection);

    return retval;
}

/**
 * Sets the timeout functions for the connection. These functions are
 * responsible for making the application's main loop aware of
 * timeouts.
 * When using Qt, typically the DbusAddTimeoutFunction would create a
 * QTimer. When using GLib, the DbusAddTimeoutFunction would call
 * g_timeout_add.
 *
 * The DbusTimeoutToggledFunction notifies the application that the
 * timeout has been enabled or disabled. Call
 * dbus_timeout_get_enabled() to check this. A disabled timeout should
 * have no effect, and enabled timeout should be added to the main
 * loop. This feature is used instead of simply adding/removing the
 * timeout because enabling/disabling can be done without memory
 * allocation. With Qt, QTimer::start() and QTimer::stop() can be used
 * to enable and disable. The toggled function may be NULL if a main
 * loop re-queries dbus_timeout_get_enabled() every time anyway.
 * Whenever a timeout is toggled, its interval may change.
 *
 * The DbusTimeout can be queried for the timer interval using
 * dbus_timeout_get_interval(). dbus_timeout_handle() should be called
 * repeatedly, each time the interval elapses, starting after it has
 * elapsed once. The timeout stops firing when it is removed with the
 * given remove_function. The timer interval may change whenever the
 * timeout is added, removed, or toggled.
 *
 * @note The thread lock on DbusConnection is held while
 * timeout functions are invoked, so inside these functions you
 * may not invoke any methods on DbusConnection or it will deadlock.
 * See the comments in the code or
http://lists.freedesktop.org/archives/dbus/2007-July/thread.html#8144
 * if you encounter this issue and want to attempt writing a patch.
 *
 * @param connection the connection.
 * @param add_function function to add a timeout.
 * @param remove_function function to remove a timeout.
 * @param toggled_function function to notify of enable/disable
 * @param data data to pass to add_function and remove_function.
 * @param free_data_function function to be called to free the data.
 * @returns #FALSE on failure (no memory)
 */
dbus_bool_t
dbus_connection_set_timeout_functions (DBusConnection
*connection,

```

```

        DbusAddTimeoutFunction      add_function,
        DbusRemoveTimeoutFunction

remove_function,

        DbusTimeoutToggledFunction
toggled_function,

        void                        *data,
        DbusFreeFunction

free_data_function)
{
    dbus_bool_t retval;

    _dbus_return_val_if_fail (connection != NULL, FALSE);

    CONNECTION_LOCK (connection);

    retval = _dbus_timeout_list_set_functions (connection->timeouts,
remove_function,
        toggled_function,
        data,
free_data_function);

    CONNECTION_UNLOCK (connection);

    return retval;
}

/**
 * Sets the mainloop wakeup function for the connection. This function
 * is responsible for waking up the main loop (if its sleeping in
 * another thread) when some some change has happened to the
 * connection that the mainloop needs to reconsider (e.g. a message
 * has been queued for writing). When using Qt, this typically
 * results in a call to QEventLoop::wakeUp(). When using GLib, it
 * would call g_main_context_wakeup().
 *
 * @param connection the connection.
 * @param wakeup_main_function function to wake up the mainloop
 * @param data data to pass wakeup_main_function
 * @param free_data_function function to be called to free the data.
 */
void
dbus_connection_set_wakeup_main_function (DBusConnection
*connection,

        DbusWakeupMainFunction

wakeup_main_function,

        void                        *data,
        DbusFreeFunction

free_data_function)
{
    void *old_data;
    DbusFreeFunction old_free_data;

```

```

    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    old_data = connection->wakeup_main_data;
    old_free_data = connection->free_wakeup_main_data;

    connection->wakeup_main_function = wakeup_main_function;
    connection->wakeup_main_data = data;
    connection->free_wakeup_main_data = free_data_function;

    CONNECTION_UNLOCK (connection);

    /* Callback outside the lock */
    if (old_free_data)
        (*old_free_data) (old_data);
}

/**
 * Set a function to be invoked when the dispatch status changes.
 * If the dispatch status is #DBUS_DISPATCH_DATA_REMAINS, then
 * dbus_connection_dispatch() needs to be called to process incoming
 * messages. However, dbus_connection_dispatch() MUST NOT BE CALLED
 * from inside the DBusDispatchStatusFunction. Indeed, almost
 * any reentrancy in this function is a bad idea. Instead,
 * the DBusDispatchStatusFunction should simply save an indication
 * that messages should be dispatched later, when the main loop
 * is re-entered.
 *
 * If you don't set a dispatch status function, you have to be sure to
 * dispatch on every iteration of your main loop, especially if
 * dbus_watch_handle() or dbus_timeout_handle() were called.
 *
 * @param connection the connection
 * @param function function to call on dispatch status changes
 * @param data data for function
 * @param free_data_function free the function data
 */
void
dbus_connection_set_dispatch_status_function (DBusConnection
*connection,

DBusDispatchStatusFunction function,

void
*data,

DBusFreeFunction
free_data_function)
{
    void *old_data;
    DBusFreeFunction old_free_data;

    _dbus_return_if_fail (connection != NULL);

```

```

CONNECTION_LOCK (connection);
old_data = connection->dispatch_status_data;
old_free_data = connection->free_dispatch_status_data;

connection->dispatch_status_function = function;
connection->dispatch_status_data = data;
connection->free_dispatch_status_data = free_data_function;

CONNECTION_UNLOCK (connection);

/* Callback outside the lock */
if (old_free_data)
    (*old_free_data) (old_data);
}

/**
 * Get the UNIX file descriptor of the connection, if any. This can
 * be used for SELinux access control checks with getpeercon() for
 * example. DO NOT read or write to the file descriptor, or try to
 * select() on it; use DBusWatch for main loop integration. Not all
 * connections will have a file descriptor. So for adding descriptors
 * to the main loop, use dbus_watch_get_unix_fd() and so forth.
 *
 * If the connection is socket-based, you can also use
 * dbus_connection_get_socket(), which will work on Windows too.
 * This function always fails on Windows.
 *
 * Right now the returned descriptor is always a socket, but
 * that is not guaranteed.
 *
 * @param connection the connection
 * @param fd return location for the file descriptor.
 * @returns #TRUE if fd is successfully obtained.
 */
dbus_bool_t
dbus_connection_get_unix_fd (DBusConnection *connection,
                             int *fd)
{
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (connection->transport != NULL, FALSE);

#ifdef DBUS_WIN
    /* FIXME do this on a lower level */
    return FALSE;
#endif

    return dbus_connection_get_socket(connection, fd);
}

/**
 * Gets the underlying Windows or UNIX socket file descriptor

```

```

* of the connection, if any. DO NOT read or write to the file
descriptor, or try to
* select() on it; use DBusWatch for main loop integration. Not all
* connections will have a socket. So for adding descriptors
* to the main loop, use dbus_watch_get_socket() and so forth.
*
* If the connection is not socket-based, this function will return
FALSE,
* even if the connection does have a file descriptor of some kind.
* i.e. this function always returns specifically a socket file
descriptor.
*
* @param connection the connection
* @param fd return location for the file descriptor.
* @returns #TRUE if fd is successfully obtained.
*/
dbus_bool_t
dbus_connection_get_socket(DBusConnection          *connection,
                           int                    *fd)
{
    dbus_bool_t retval;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (connection->transport != NULL, FALSE);

    CONNECTION_LOCK (connection);

    retval = _dbus_transport_get_socket_fd (connection->transport,
                                           fd);

    CONNECTION_UNLOCK (connection);

    return retval;
}

/**
* Gets the UNIX user ID of the connection if known. Returns #TRUE if
* the uid is filled in. Always returns #FALSE on non-UNIX platforms
* for now, though in theory someone could hook Windows to NIS or
* something. Always returns #FALSE prior to authenticating the
* connection.
*
* The UID is only read by servers from clients; clients can't usually
* get the UID of servers, because servers do not authenticate to
* clients. The returned UID is the UID the connection authenticated
* as.
*
* The message bus is a server and the apps connecting to the bus
* are clients.
*
*/

```

```

* You can ask the bus to tell you the UID of another connection
though
* if you like; this is done with dbus_bus_get_unix_user().
*
* @param connection the connection
* @param uid return location for the user ID
* @returns #TRUE if uid is filled in with a valid user ID
*/
dbus_bool_t
dbus_connection_get_unix_user (DBusConnection *connection,
                               unsigned long *uid)
{
    dbus_bool_t result;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (uid != NULL, FALSE);

    CONNECTION_LOCK (connection);

    if (!_dbus_transport_get_is_authenticated (connection->transport))
        result = FALSE;
    else
        result = _dbus_transport_get_unix_user (connection->transport,
                                                uid);

#ifdef DBUS_WIN
    _dbus_assert (!result);
#endif

    CONNECTION_UNLOCK (connection);

    return result;
}

/**
* Gets the process ID of the connection if any.
* Returns #TRUE if the pid is filled in.
* Always returns #FALSE prior to authenticating the
* connection.
*
* @param connection the connection
* @param pid return location for the process ID
* @returns #TRUE if uid is filled in with a valid process ID
*/
dbus_bool_t
dbus_connection_get_unix_process_id (DBusConnection *connection,
                                     unsigned long *pid)
{
    dbus_bool_t result;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (pid != NULL, FALSE);

```

```

CONNECTION_LOCK (connection);

if (!_dbus_transport_get_is_authenticated (connection->transport))
    result = FALSE;
else
    result = _dbus_transport_get_unix_process_id (connection-
>transport,
                                                pid);

CONNECTION_UNLOCK (connection);

return result;
}

/**
 * Gets the ADT audit data of the connection if any.
 * Returns #TRUE if the structure pointer is returned.
 * Always returns #FALSE prior to authenticating the
 * connection.
 *
 * @param connection the connection
 * @param data return location for audit data
 * @returns #TRUE if audit data is filled in with a valid ucred
pointer
 */
dbus_bool_t
dbus_connection_get_adt_audit_session_data (DBusConnection
*connection,
                                           void          **data,
                                           dbus_int32_t  *data_size)
{
    dbus_bool_t result;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (data != NULL, FALSE);
    _dbus_return_val_if_fail (data_size != NULL, FALSE);

    CONNECTION_LOCK (connection);

    if (!_dbus_transport_get_is_authenticated (connection->transport))
        result = FALSE;
    else
        result = _dbus_transport_get_adt_audit_session_data (connection-
>transport,
                                                            data,
                                                            data_size);

    CONNECTION_UNLOCK (connection);

    return result;
}

```



```

/**
 * Sets a predicate function used to determine whether a given user ID
 * is allowed to connect. When an incoming connection has
 * authenticated with a particular user ID, this function is called;
 * if it returns #TRUE, the connection is allowed to proceed,
 * otherwise the connection is disconnected.
 *
 * If the function is set to #NULL (as it is by default), then
 * only the same UID as the server process will be allowed to
 * connect. Also, root is always allowed to connect.
 *
 * On Windows, the function will be set and its free_data_function
will
 * be invoked when the connection is freed or a new function is set.
 * However, the function will never be called, because there are
 * no UNIX user ids to pass to it, or at least none of the existing
 * auth protocols would allow authenticating as a UNIX user on
Windows.
 *
 * @param connection the connection
 * @param function the predicate
 * @param data data to pass to the predicate
 * @param free_data_function function to free the data
 */
void
dbus_connection_set_unix_user_function (DBusConnection
*connection,
                                     DBusAllowUnixUserFunction
function,
                                     void
*data,
                                     DBusFreeFunction
free_data_function)
{
    void *old_data = NULL;
    DBusFreeFunction old_free_function = NULL;

    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_unix_user_function (connection->transport,
function, data,
free_data_function,
                                     &old_data,
&old_free_function);
    CONNECTION_UNLOCK (connection);

    if (old_free_function != NULL)
        (* old_free_function) (old_data);
}
/**

```



```

#ifdef DBUS_UNIX
    _dbus_assert (!result);
#endif

    CONNECTION_UNLOCK (connection);

    return result;
}

/**
 * Sets a predicate function used to determine whether a given user ID
 * is allowed to connect. When an incoming connection has
 * authenticated with a particular user ID, this function is called;
 * if it returns #TRUE, the connection is allowed to proceed,
 * otherwise the connection is disconnected.
 *
 * If the function is set to #NULL (as it is by default), then
 * only the same user owning the server process will be allowed to
 * connect.
 *
 * On UNIX, the function will be set and its free_data_function will
 * be invoked when the connection is freed or a new function is set.
 * However, the function will never be called, because there is no
 * way right now to authenticate as a Windows user on UNIX.
 *
 * @param connection the connection
 * @param function the predicate
 * @param data data to pass to the predicate
 * @param free_data_function function to free the data
 */
void
dbus_connection_set_windows_user_function (DBusConnection
*connection,

DBusAllowWindowsUserFunction function,

void

*data,

DBusFreeFunction

free_data_function)
{
    void *old_data = NULL;
    DBusFreeFunction old_free_function = NULL;

    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_windows_user_function (connection->transport,
function, data,

free_data_function,

&old_data,

&old_free_function);
    CONNECTION_UNLOCK (connection);
}

```

```

    if (old_free_function != NULL)
        (* old_free_function) (old_data);
}

/**
 * This function must be called on the server side of a connection
when the
 * connection is first seen in the #DBusNewConnectionFunction. If set
to
 * #TRUE (the default is #FALSE), then the connection can proceed even
if
 * the client does not authenticate as some user identity, i.e.
clients
 * can connect anonymously.
 *
 * This setting interacts with the available authorization mechanisms
 * (see dbus_server_set_auth_mechanisms()). Namely, an auth mechanism
 * such as ANONYMOUS that supports anonymous auth must be included in
 * the list of available mechanisms for anonymous login to work.
 *
 * This setting also changes the default rule for connections
 * authorized as a user; normally, if a connection authorizes as
 * a user identity, it is permitted if the user identity is
 * root or the user identity matches the user identity of the server
 * process. If anonymous connections are allowed, however,
 * then any user identity is allowed.
 *
 * You can override the rules for connections authorized as a
 * user identity with dbus_connection_set_unix_user_function()
 * and dbus_connection_set_windows_user_function().
 *
 * @param connection the connection
 * @param value whether to allow authentication as an anonymous user
 */
void
dbus_connection_set_allow_anonymous (DBusConnection
*connection,
                                     dbus_bool_t
value)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_allow_anonymous (connection->transport, value);
    CONNECTION_UNLOCK (connection);
}

/**
 *
 * Normally #DBusConnection automatically handles all messages to the
 * org.freedesktop.DBus.Peer interface. However, the message bus wants

```

```

* to be able to route methods on that interface through the bus and
* to other applications. If routing peer messages is enabled, then
* messages with the org.freedesktop.DBus.Peer interface that also
* have a bus destination name set will not be automatically
* handled by the #DBusConnection and instead will be dispatched
* normally to the application.
*
* If a normal application sets this flag, it can break things badly.
* So don't set this unless you are the message bus.
*
* @param connection the connection
* @param value #TRUE to pass through org.freedesktop.DBus.Peer
messages with a bus name set
*/
void
dbus_connection_set_route_peer_messages (DBusConnection
*connection,
                                         dbus_bool_t
value)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    connection->route_peer_messages = TRUE;
    CONNECTION_UNLOCK (connection);
}

/**
* Adds a message filter. Filters are handlers that are run on all
* incoming messages, prior to the objects registered with
* dbus_connection_register_object_path(). Filters are run in the
* order that they were added. The same handler can be added as a
* filter more than once, in which case it will be run more than once.
* Filters added during a filter callback won't be run on the message
* being processed.
*
* @todo we don't run filters on messages while blocking without
* entering the main loop, since filters are run as part of
* dbus_connection_dispatch(). This is probably a feature, as filters
* could create arbitrary reentrancy. But kind of sucks if you're
* trying to filter METHOD_RETURN for some reason.
*
* @param connection the connection
* @param function function to handle messages
* @param user_data user data to pass to the function
* @param free_data_function function to use for freeing user data
* @returns #TRUE on success, #FALSE if not enough memory.
*/
dbus_bool_t
dbus_connection_add_filter (DBusConnection *connection,
                           DBusHandleMessageFunction function,
                           void *user_data,

```



```

                                void                                *user_data)
{
    DBusList *link;
    DBusMessageFilter *filter;

    _dbus_return_if_fail (connection != NULL);
    _dbus_return_if_fail (function != NULL);

    CONNECTION_LOCK (connection);

    filter = NULL;

    link = _dbus_list_get_last_link (&connection->filter_list);
    while (link != NULL)
    {
        filter = link->data;

        if (filter->function == function &&
            filter->user_data == user_data)
        {
            _dbus_list_remove_link (&connection->filter_list, link);
            filter->function = NULL;

            break;
        }

        link = _dbus_list_get_prev_link (&connection->filter_list,
link);
        filter = NULL;
    }

    CONNECTION_UNLOCK (connection);

#ifdef DBUS_DISABLE_CHECKS
    if (filter == NULL)
    {
        _dbus_warn_check_failed ("Attempt to remove filter function %p
user data %p, but no such filter has been added\n",
                                function, user_data);

        return;
    }
#endif

    /* Call application code */
    if (filter->free_user_data_function)
        (* filter->free_user_data_function) (filter->user_data);

    filter->free_user_data_function = NULL;
    filter->user_data = NULL;

    _dbus_message_filter_unref (filter);
}

```

```

/**
 * Registers a handler for a given path or subsection in the object
 * hierarchy. The given vtable handles messages sent to exactly the
 * given path or also for paths bellow that, depending on fallback
 * parameter.
 *
 * @param connection the connection
 * @param fallback whether to handle messages also for "subdirectory"
 * @param path a '/' delimited string of path elements
 * @param vtable the virtual table
 * @param user_data data to pass to functions in the vtable
 * @param error address where an error can be returned
 * @returns #FALSE if an error (#DBUS_ERROR_NO_MEMORY or
 *         #DBUS_ERROR_OBJECT_PATH_IN_USE) is reported
 */
static dbus_bool_t
_dbus_connection_register_object_path (DBusConnection
*connection,
                                     dbus_bool_t
fallback,
                                     const char
*path,
                                     const DBusObjectPathVTable
*vtable,
                                     void
*user_data,
                                     DBusError
*error)
{
    char **decomposed_path;
    dbus_bool_t retval;

    if (!_dbus_decompose_path (path, strlen (path), &decomposed_path,
NULL))
        return FALSE;

    CONNECTION_LOCK (connection);

    retval = _dbus_object_tree_register (connection->objects,
                                     fallback,
                                     (const char **)
decomposed_path, vtable,
                                     user_data, error);

    CONNECTION_UNLOCK (connection);

    dbus_free_string_array (decomposed_path);

    return retval;
}

```



```

/**
 * Registers a handler for a given path in the object hierarchy.
 * The given vtable handles messages sent to exactly the given path.
 *
 * @param connection the connection
 * @param path a '/' delimited string of path elements
 * @param vtable the virtual table
 * @param user_data data to pass to functions in the vtable
 * @param error address where an error can be returned
 * @returns #FALSE if an error (#DBUS_ERROR_NO_MEMORY or
 *         #DBUS_ERROR_OBJECT_PATH_IN_USE) is reported
 */
dbus_bool_t
dbus_connection_try_register_object_path (DBusConnection
*connection,
                                         const char
*path,
                                         const DBusObjectPathVTable
*vtable,
                                         void
*user_data,
                                         DBusError
*error)
{
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (path[0] == '/', FALSE);
    _dbus_return_val_if_fail (vtable != NULL, FALSE);

    return _dbus_connection_register_object_path (connection, FALSE,
path, vtable, user_data, error);
}

/**
 * Registers a handler for a given path in the object hierarchy.
 * The given vtable handles messages sent to exactly the given path.
 *
 * It is a bug to call this function for object paths which already
 * have a handler. Use dbus_connection_try_register_object_path() if
this
 * might be the case.
 *
 * @param connection the connection
 * @param path a '/' delimited string of path elements
 * @param vtable the virtual table
 * @param user_data data to pass to functions in the vtable
 * @returns #FALSE if an error (#DBUS_ERROR_NO_MEMORY or
 *         #DBUS_ERROR_OBJECT_PATH_IN_USE) occurred
 */
dbus_bool_t
dbus_connection_register_object_path (DBusConnection
*connection,

```

```

const char
*path,
const DBusObjectPathVTable
*vtable,
void
*user_data)
{
    dbus_bool_t retval;
    DBusError error = DBUS_ERROR_INIT;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (path[0] == '/', FALSE);
    _dbus_return_val_if_fail (vtable != NULL, FALSE);

    retval = _dbus_connection_register_object_path (connection, FALSE,
path, vtable, user_data, &error);

    if (dbus_error_has_name (&error, DBUS_ERROR_OBJECT_PATH_IN_USE))
    {
        _dbus_warn ("%s\n", error.message);
        dbus_error_free (&error);
        return FALSE;
    }

    return retval;
}

/**
 * Registers a fallback handler for a given subsection of the object
 * hierarchy. The given vtable handles messages at or below the given
 * path. You can use this to establish a default message handling
 * policy for a whole "subdirectory."
 *
 * @param connection the connection
 * @param path a '/' delimited string of path elements
 * @param vtable the virtual table
 * @param user_data data to pass to functions in the vtable
 * @param error address where an error can be returned
 * @returns #FALSE if an error (#DBUS_ERROR_NO_MEMORY or
 *         #DBUS_ERROR_OBJECT_PATH_IN_USE) is reported
 */
dbus_bool_t
dbus_connection_try_register_fallback (DBusConnection
*connection,
const char
*path,
const DBusObjectPathVTable
*vtable,
void
*user_data,

```

DBusError

```
*error)
{
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (path[0] == '/', FALSE);
    _dbus_return_val_if_fail (vtable != NULL, FALSE);

    return _dbus_connection_register_object_path (connection, TRUE,
path, vtable, user_data, error);
}

/**
 * Registers a fallback handler for a given subsection of the object
 * hierarchy. The given vtable handles messages at or below the given
 * path. You can use this to establish a default message handling
 * policy for a whole "subdirectory."
 *
 * It is a bug to call this function for object paths which already
 * have a handler. Use dbus_connection_try_register_fallback() if this
 * might be the case.
 *
 * @param connection the connection
 * @param path a '/' delimited string of path elements
 * @param vtable the virtual table
 * @param user_data data to pass to functions in the vtable
 * @returns #FALSE if an error (#DBUS_ERROR_NO_MEMORY or
 *         #DBUS_ERROR_OBJECT_PATH_IN_USE) occurred
 */
dbus_bool_t
dbus_connection_register_fallback (DBusConnection
*connection,
                                const char
                                *path,
                                const DBusObjectPathVTable
*vtable,
                                void
*user_data)
{
    dbus_bool_t retval;
    DBusError error = DBUS_ERROR_INIT;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (path[0] == '/', FALSE);
    _dbus_return_val_if_fail (vtable != NULL, FALSE);

    retval = _dbus_connection_register_object_path (connection, TRUE,
path, vtable, user_data, &error);

    if (dbus_error_has_name (&error, DBUS_ERROR_OBJECT_PATH_IN_USE))
    {
        _dbus_warn ("%s\n", error.message);
    }
}
```

```

        dbus_error_free (&error);
        return FALSE;
    }

    return retval;
}

/**
 * Unregisters the handler registered with exactly the given path.
 * It's a bug to call this function for a path that isn't registered.
 * Can unregister both fallback paths and object paths.
 *
 * @param connection the connection
 * @param path a '/' delimited string of path elements
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_connection_unregister_object_path (DBusConnection
*connection,
                                     const char
*path)
{
    char **decomposed_path;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (path[0] == '/', FALSE);

    if (!_dbus_decompose_path (path, strlen (path), &decomposed_path,
NULL))
        return FALSE;

    CONNECTION_LOCK (connection);

    _dbus_object_tree_unregister_and_unlock (connection->objects, (const
char **) decomposed_path);

    dbus_free_string_array (decomposed_path);

    return TRUE;
}

/**
 * Gets the user data passed to dbus_connection_register_object_path()
 * or dbus_connection_register_fallback(). If nothing was registered
 * at this path, the data is filled in with #NULL.
 *
 * @param connection the connection
 * @param path the path you registered with
 * @param data_p location to store the user data, or #NULL
 * @returns #FALSE if not enough memory
 */

```

```

dbus_bool_t
dbus_connection_get_object_path_data (DBusConnection *connection,
                                     const char      *path,
                                     void            **data_p)
{
    char **decomposed_path;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (path != NULL, FALSE);
    _dbus_return_val_if_fail (data_p != NULL, FALSE);

    *data_p = NULL;

    if (!_dbus_decompose_path (path, strlen (path), &decomposed_path,
                              NULL))
        return FALSE;

    CONNECTION_LOCK (connection);

    *data_p = _dbus_object_tree_get_user_data_unlocked (connection-
>objects, (const char**) decomposed_path);

    CONNECTION_UNLOCK (connection);

    dbus_free_string_array (decomposed_path);

    return TRUE;
}

/**
 * Lists the registered fallback handlers and object path handlers at
 * the given parent_path. The returned array should be freed with
 * dbus_free_string_array().
 *
 * @param connection the connection
 * @param parent_path the path to list the child handlers of
 * @param child_entries returns #NULL-terminated array of children
 * @returns #FALSE if no memory to allocate the child entries
 */
dbus_bool_t
dbus_connection_list_registered (DBusConnection
 *connection,
                                const char
 *parent_path,
                                char
 ***child_entries)
{
    char **decomposed_path;
    dbus_bool_t retval;
    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (parent_path != NULL, FALSE);
    _dbus_return_val_if_fail (parent_path[0] == '/', FALSE);

```

```

    _dbus_return_val_if_fail (child_entries != NULL, FALSE);

    if (!_dbus_decompose_path (parent_path, strlen (parent_path),
&decomposed_path, NULL))
        return FALSE;

    CONNECTION_LOCK (connection);

    retval = _dbus_object_tree_list_registered_and_unlock (connection-
>objects,
                                                            (const char **)
decomposed_path,
                                                            child_entries);
    dbus_free_string_array (decomposed_path);

    return retval;
}

static DBusDataSlotAllocator slot_allocator;
_DBUS_DEFINE_GLOBAL_LOCK (connection_slots);

/**
 * Allocates an integer ID to be used for storing application-specific
 * data on any DBusConnection. The allocated ID may then be used
 * with dbus_connection_set_data() and dbus_connection_get_data().
 * The passed-in slot must be initialized to -1, and is filled in
 * with the slot ID. If the passed-in slot is not -1, it's assumed
 * to be already allocated, and its refcount is incremented.
 *
 * The allocated slot is global, i.e. all DBusConnection objects will
 * have a slot with the given integer ID reserved.
 *
 * @param slot_p address of a global variable storing the slot
 * @returns #FALSE on failure (no memory)
 */
dbus_bool_t
dbus_connection_allocate_data_slot (dbus_int32_t *slot_p)
{
    return _dbus_data_slot_allocator_alloc (&slot_allocator,
&_DBUS_LOCK_NAME
(connection_slots),
                                                            slot_p);
}

/**
 * Deallocates a global ID for connection data slots.
 * dbus_connection_get_data() and dbus_connection_set_data() may no
 * longer be used with this slot. Existing data stored on existing
 * DBusConnection objects will be freed when the connection is
 * finalized, but may not be retrieved (and may only be replaced if
 * someone else reallocates the slot). When the refcount on the
 * passed-in slot reaches 0, it is set to -1.

```

```

*
* @param slot_p address storing the slot to deallocate
*/
void
dbus_connection_free_data_slot (dbus_int32_t *slot_p)
{
    _dbus_return_if_fail (*slot_p >= 0);

    _dbus_data_slot_allocator_free (&slot_allocator, slot_p);
}

/**
 * Stores a pointer on a DBusConnection, along
 * with an optional function to be used for freeing
 * the data when the data is set again, or when
 * the connection is finalized. The slot number
 * must have been allocated with dbus_connection_allocate_data_slot().
 *
 * @note This function does not take the
 * main thread lock on DBusConnection, which allows it to be
 * used from inside watch and timeout functions. (See the
 * note in docs for dbus_connection_set_watch_functions().)
 * A side effect of this is that you need to know there's
 * a reference held on the connection while invoking
 * dbus_connection_set_data(), or the connection could be
 * finalized during dbus_connection_set_data().
 *
 * @param connection the connection
 * @param slot the slot number
 * @param data the data to store
 * @param free_data_func finalizer function for the data
 * @returns #TRUE if there was enough memory to store the data
 */
dbus_bool_t
dbus_connection_set_data (DBusConnection *connection,
                        dbus_int32_t slot,
                        void *data,
                        DBusFreeFunction free_data_func)
{
    DBusFreeFunction old_free_func;
    void *old_data;
    dbus_bool_t retval;

    _dbus_return_val_if_fail (connection != NULL, FALSE);
    _dbus_return_val_if_fail (slot >= 0, FALSE);

    SLOTS_LOCK (connection);

    retval = _dbus_data_slot_list_set (&slot_allocator,
                                     &connection->slot_list,
                                     slot, data, free_data_func,
                                     &old_free_func, &old_data);
}

```

```

SLOTS_UNLOCK (connection);

if (retval)
{
    /* Do the actual free outside the connection lock */
    if (old_free_func)
        (* old_free_func) (old_data);
}

return retval;
}

/**
 * Retrieves data previously set with dbus_connection_set_data().
 * The slot must still be allocated (must not have been freed).
 *
 * @note This function does not take the
 * main thread lock on DBusConnection, which allows it to be
 * used from inside watch and timeout functions. (See the
 * note in docs for dbus_connection_set_watch_functions().)
 * A side effect of this is that you need to know there's
 * a reference held on the connection while invoking
 * dbus_connection_get_data(), or the connection could be
 * finalized during dbus_connection_get_data().
 *
 * @param connection the connection
 * @param slot the slot to get data from
 * @returns the data, or #NULL if not found
 */
void*
dbus_connection_get_data (DBusConnection *connection,
                        dbus_int32_t slot)
{
    void *res;

    _dbus_return_val_if_fail (connection != NULL, NULL);

    SLOTS_LOCK (connection);

    res = _dbus_data_slot_list_get (&slot_allocator,
                                    &connection->slot_list,
                                    slot);

    SLOTS_UNLOCK (connection);

    return res;
}

/**
 * This function sets a global flag for whether dbus_connection_new()
 * will set SIGPIPE behavior to SIG_IGN.

```



```

*
* @param will_modify_sigpipe #TRUE to allow sigpipe to be set to
SIG_IGN
*/
void
dbus_connection_set_change_sigpipe (dbus_bool_t will_modify_sigpipe)
{
    _dbus_modify_sigpipe = will_modify_sigpipe != FALSE;
}

/**
 * Specifies the maximum size message this connection is allowed to
 * receive. Larger messages will result in disconnecting the
 * connection.
 *
 * @param connection a #DBusConnection
 * @param size maximum message size the connection can receive, in
bytes
 */
void
dbus_connection_set_max_message_size (DBusConnection *connection,
                                     long size)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_max_message_size (connection->transport,
                                         size);
    CONNECTION_UNLOCK (connection);
}

/**
 * Gets the value set by dbus_connection_set_max_message_size().
 *
 * @param connection the connection
 * @returns the max size of a single message
 */
long
dbus_connection_get_max_message_size (DBusConnection *connection)
{
    long res;

    _dbus_return_val_if_fail (connection != NULL, 0);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_max_message_size (connection->transport);
    CONNECTION_UNLOCK (connection);
    return res;
}

/**
 * Specifies the maximum number of unix fds a message on this

```

```

* connection is allowed to receive. Messages with more unix fds will
* result in disconnecting the connection.
*
* @param connection a #DBusConnection
* @param size maximum message unix fds the connection can receive
*/
void
dbus_connection_set_max_message_unix_fds (DBusConnection *connection,
                                          long n)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_max_message_unix_fds (connection->transport,
                                              n);
    CONNECTION_UNLOCK (connection);
}

/**
 * Gets the value set by dbus_connection_set_max_message_unix_fds().
 *
 * @param connection the connection
 * @returns the max number of unix fds of a single message
 */
long
dbus_connection_get_max_message_unix_fds (DBusConnection *connection)
{
    long res;

    _dbus_return_val_if_fail (connection != NULL, 0);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_max_message_unix_fds (connection->transport);
    CONNECTION_UNLOCK (connection);
    return res;
}

/**
 * Sets the maximum total number of bytes that can be used for all
messages
 * received on this connection. Messages count toward the maximum
until
 * they are finalized. When the maximum is reached, the connection
will
 * not read more data until some messages are finalized.
 *
 * The semantics of the maximum are: if outstanding messages are
 * already above the maximum, additional messages will not be read.
 * The semantics are not: if the next message would cause us to exceed
 * the maximum, we don't read it. The reason is that we don't know the
 * size of a message until after we read it.

```

```

*
* Thus, the max live messages size can actually be exceeded
* by up to the maximum size of a single message.
*
* Also, if we read say 1024 bytes off the wire in a single read(),
* and that contains a half-dozen small messages, we may exceed the
* size max by that amount. But this should be inconsequential.
*
* This does imply that we can't call read() with a buffer larger
* than we're willing to exceed this limit by.
*
* @param connection the connection
* @param size the maximum size in bytes of all outstanding messages
*/
void
dbus_connection_set_max_received_size (DBusConnection *connection,
                                       long size)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_max_received_size (connection->transport,
                                           size);
    CONNECTION_UNLOCK (connection);
}

/**
 * Gets the value set by dbus_connection_set_max_received_size().
 *
 * @param connection the connection
 * @returns the max size of all live messages
 */
long
dbus_connection_get_max_received_size (DBusConnection *connection)
{
    long res;

    _dbus_return_val_if_fail (connection != NULL, 0);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_max_received_size (connection->transport);
    CONNECTION_UNLOCK (connection);
    return res;
}

/**
 * Sets the maximum total number of unix fds that can be used for all
 * messages
 * received on this connection. Messages count toward the maximum
 * until
 * they are finalized. When the maximum is reached, the connection
 * will

```

```

* not read more data until some messages are finalized.
*
* The semantics are analogous to those of
dbus_connection_set_max_received_size().
*
* @param connection the connection
* @param size the maximum size in bytes of all outstanding messages
*/
void
dbus_connection_set_max_received_unix_fds (DBusConnection *connection,
                                           long n)
{
    _dbus_return_if_fail (connection != NULL);

    CONNECTION_LOCK (connection);
    _dbus_transport_set_max_received_unix_fds (connection->transport,
                                               n);
    CONNECTION_UNLOCK (connection);
}

/**
* Gets the value set by dbus_connection_set_max_received_unix_fds().
*
* @param connection the connection
* @returns the max unix fds of all live messages
*/
long
dbus_connection_get_max_received_unix_fds (DBusConnection *connection)
{
    long res;

    _dbus_return_val_if_fail (connection != NULL, 0);

    CONNECTION_LOCK (connection);
    res = _dbus_transport_get_max_received_unix_fds (connection-
>transport);
    CONNECTION_UNLOCK (connection);
    return res;
}

/**
* Gets the approximate size in bytes of all messages in the outgoing
* message queue. The size is approximate in that you shouldn't use
* it to decide how many bytes to read off the network or anything
* of that nature, as optimizations may choose to tell small white
lies
* to avoid performance overhead.
*
* @param connection the connection
* @returns the number of bytes that have been queued up but not sent
*/
long

```

```

dbus_connection_get_outgoing_size (DBusConnection *connection)
{
    long res;

    _dbus_return_val_if_fail (connection != NULL, 0);

    CONNECTION_LOCK (connection);
    res = _dbus_counter_get_size_value (connection->outgoing_counter);
    CONNECTION_UNLOCK (connection);
    return res;
}

#ifdef DBUS_ENABLE_STATS
void
_dbus_connection_get_stats (DBusConnection *connection,
                            dbus_uint32_t *in_messages,
                            dbus_uint32_t *in_bytes,
                            dbus_uint32_t *in_fds,
                            dbus_uint32_t *in_peak_bytes,
                            dbus_uint32_t *in_peak_fds,
                            dbus_uint32_t *out_messages,
                            dbus_uint32_t *out_bytes,
                            dbus_uint32_t *out_fds,
                            dbus_uint32_t *out_peak_bytes,
                            dbus_uint32_t *out_peak_fds)
{
    CONNECTION_LOCK (connection);

    if (in_messages != NULL)
        *in_messages = connection->n_incoming;

    _dbus_transport_get_stats (connection->transport,
                               in_bytes, in_fds, in_peak_bytes,
in_peak_fds);

    if (out_messages != NULL)
        *out_messages = connection->n_outgoing;

    if (out_bytes != NULL)
        *out_bytes = _dbus_counter_get_size_value (connection-
>outgoing_counter);

    if (out_fds != NULL)
        *out_fds = _dbus_counter_get_unix_fd_value (connection-
>outgoing_counter);

    if (out_peak_bytes != NULL)
        *out_peak_bytes = _dbus_counter_get_peak_size_value (connection-
>outgoing_counter);

    if (out_peak_fds != NULL)

```

```
    *out_peak_fds = _dbus_counter_get_peak_unix_fd_value (connection->outgoing_counter);
```

```
    CONNECTION_UNLOCK (connection);  
}  
#endif /* DBUS_ENABLE_STATS */
```

```
/**  
 * Gets the approximate number of uni fds of all messages in the  
 * outgoing message queue.  
 *  
 * @param connection the connection  
 * @returns the number of unix fds that have been queued up but not  
sent  
 */
```

```
long  
dbus_connection_get_outgoing_unix_fds (DBusConnection *connection)
```

```
{  
    long res;  
  
    _dbus_return_val_if_fail (connection != NULL, 0);  
  
    CONNECTION_LOCK (connection);  
    res = _dbus_counter_get_unix_fd_value (connection->outgoing_counter);  
    CONNECTION_UNLOCK (connection);  
    return res;  
}
```

```
#ifdef DBUS_BUILD_TESTS
```

```
/**  
 * Returns the address of the transport object of this connection  
 *  
 * @param connection the connection  
 * @returns the address string  
 */
```

```
const char*  
_dbus_connection_get_address (DBusConnection *connection)  
{  
    return _dbus_transport_get_address (connection->transport);  
}
```

```
#endif
```

```
/** @} */
```

File = dbus-connection.h

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* dbus-connection.h DBusConnection object  
 *
```

```

* Copyright (C) 2002, 2003 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifdef DBUS_CONNECTION_H
#define DBUS_CONNECTION_H

#include <dbus/dbus-errors.h>
#include <dbus/dbus-memory.h>
#include <dbus/dbus-message.h>
#include <dbus/dbus-shared.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusConnection
 * @{
 */

/* documented in dbus-watch.c */
typedef struct DBusWatch DBusWatch;
/* documented in dbus-timeout.c */
typedef struct DBusTimeout DBusTimeout;
/** Opaque type representing preallocated resources so a message can
be sent without further memory allocation. */
typedef struct DBusPreallocatedSend DBusPreallocatedSend;
/** Opaque type representing a method call that has not yet received a
reply. */
typedef struct DBusPendingCall DBusPendingCall;

```



```

/** Called when dbus_watch_get_enabled() may return a different value
 * than it did before. Set by dbus_connection_set_watch_functions()
 * or dbus_server_set_watch_functions().
 */
typedef void          (* DBusWatchToggledFunction) (DBusWatch
*watch,
                                                    void
*data);
/** Called when libdbus no longer needs a watch to be monitored by the
 * main loop. Set by dbus_connection_set_watch_functions() or
 * dbus_server_set_watch_functions().
 */
typedef void          (* DBusRemoveWatchFunction) (DBusWatch
*watch,
                                                    void
*data);
/** Called when libdbus needs a new timeout to be monitored by the
main
 * loop. Returns #FALSE if it lacks enough memory to add the
 * watch. Set by dbus_connection_set_timeout_functions() or
 * dbus_server_set_timeout_functions().
 */
typedef dbus_bool_t (* DBusAddTimeoutFunction) (DBusTimeout
*timeout,
                                                void
*data);
/** Called when dbus_timeout_get_enabled() may return a different
 * value than it did before.
 * Set by dbus_connection_set_timeout_functions() or
 * dbus_server_set_timeout_functions().
 */
typedef void          (* DBusTimeoutToggledFunction) (DBusTimeout
*timeout,
                                                       void
*data);
/** Called when libdbus no longer needs a timeout to be monitored by
the
 * main loop. Set by dbus_connection_set_timeout_functions() or
 * dbus_server_set_timeout_functions().
 */
typedef void          (* DBusRemoveTimeoutFunction) (DBusTimeout
*timeout,
                                                     void
*data);
/** Called when the return value of
dbus_connection_get_dispatch_status()
 * may have changed. Set with
dbus_connection_set_dispatch_status_function().
 */
typedef void          (* DBusDispatchStatusFunction) (DBusConnection
*connection,

```

```

                                                                    DbusDispatchStatus
new_status,
                                                                    void
*data);
/**
 * Called when the main loop's thread should be notified that there's
now work
 * to do. Set with dbus_connection_set_wakeup_main_function().
 */
typedef void          (* DbusWakeupMainFunction)      (void
*data);

/**
 * Called during authentication to check whether the given UNIX user
 * ID is allowed to connect, if the client tried to auth as a UNIX
 * user ID. Normally on Windows this would never happen. Set with
 * dbus_connection_set_unix_user_function().
 */
typedef dbus_bool_t (* DbusAllowUnixUserFunction)    (DBusConnection
*connection,
                                                                    unsigned long
uid,
                                                                    void
*data);

/**
 * Called during authentication to check whether the given Windows
user
 * ID is allowed to connect, if the client tried to auth as a Windows
 * user ID. Normally on UNIX this would never happen. Set with
 * dbus_connection_set_windows_user_function().
 */
typedef dbus_bool_t (* DbusAllowWindowsUserFunction) (DBusConnection
*connection,
                                                                    const char
*user_sid,
                                                                    void
*data);

/**
 * Called when a pending call now has a reply available. Set with
 * dbus_pending_call_set_notify().
 */
typedef void (* DbusPendingCallNotifyFunction)       (DBusPendingCall
*pending,
                                                                    void
*user_data);

/**
 * Called when a message needs to be handled. The result indicates
whether or

```



```

                                                                    int
timeout_milliseconds);
DBUS_EXPORT
DBusMessage *          dbus_connection_send_with_reply_and_block
(DBusConnection      *connection,

DBusMessage          *message,

                                                                    int
timeout_milliseconds,

DBusError            *error);
DBUS_EXPORT
dbus_bool_t          dbus_connection_set_watch_functions
(DBusConnection      *connection,

DBusAddWatchFunction  add_function,

DBusRemoveWatchFunction  remove_function,

DBusWatchToggledFunction  toggled_function,

                                                                    void
*data,

DBusFreeFunction      free_data_function);
DBUS_EXPORT
dbus_bool_t          dbus_connection_set_timeout_functions
(DBusConnection      *connection,

DBusAddTimeoutFunction  add_function,

DBusRemoveTimeoutFunction  remove_function,

DBusTimeoutToggledFunction  toggled_function,

                                                                    void
*data,

DBusFreeFunction      free_data_function);
DBUS_EXPORT
void                dbus_connection_set_wakeup_main_function
(DBusConnection      *connection,

DBusWakeupMainFunction  wakeup_main_function,

                                                                    void
*data,

DBusFreeFunction      free_data_function);
DBUS_EXPORT
void                dbus_connection_set_dispatch_status_function
(DBusConnection      *connection,

DBusDispatchStatusFunction  function,

```

```

void
*data,

DBusFreeFunction          free_data_function);
DBUS_EXPORT
dbus_bool_t              dbus_connection_get_unix_user
(DBusConnection          *connection,

unsigned long            *uid);
DBUS_EXPORT
dbus_bool_t              dbus_connection_get_unix_process_id
(DBusConnection          *connection,

unsigned long            *pid);
DBUS_EXPORT
dbus_bool_t              dbus_connection_get_adt_audit_session_data
(DBusConnection          *connection,

**data,

dbus_int32_t             *data_size);
DBUS_EXPORT
void                    dbus_connection_set_unix_user_function
(DBusConnection          *connection,

DBusAllowUnixUserFunction function,

*data,

DBusFreeFunction          free_data_function);
DBUS_EXPORT
dbus_bool_t              dbus_connection_get_windows_user
(DBusConnection          *connection,

**windows_sid_p);
DBUS_EXPORT
void                    dbus_connection_set_windows_user_function
(DBusConnection          *connection,

DBusAllowWindowsUserFunction function,

*data,

DBusFreeFunction          free_data_function);
DBUS_EXPORT
void                    dbus_connection_set_allow_anonymous
(DBusConnection          *connection,

dbus_bool_t              value);
DBUS_EXPORT
void                    dbus_connection_set_route_peer_messages
(DBusConnection          *connection,

```

```

dbus_bool_t          value);

/* Filters */

DBUS_EXPORT
dbus_bool_t dbus_connection_add_filter      (DBusConnection
*connection,
                                           DbusHandleMessageFunction
function,
                                           void
*user_data,
                                           DbusFreeFunction
free_data_function);
DBUS_EXPORT
void        dbus_connection_remove_filter  (DBusConnection
*connection,
                                           DbusHandleMessageFunction
function,
                                           void
*user_data);

/* Other */
DBUS_EXPORT
dbus_bool_t dbus_connection_allocate_data_slot (dbus_int32_t
*slot_p);
DBUS_EXPORT
void        dbus_connection_free_data_slot   (dbus_int32_t
*slot_p);
DBUS_EXPORT
dbus_bool_t dbus_connection_set_data        (DBusConnection
*connection,
                                           dbus_int32_t
slot,
                                           void
*data,
                                           DbusFreeFunction
free_data_func);
DBUS_EXPORT
void*       dbus_connection_get_data        (DBusConnection
*connection,
                                           dbus_int32_t
slot);

DBUS_EXPORT
void        dbus_connection_set_change_sigpipe (dbus_bool_t
will_modify_sigpipe);

DBUS_EXPORT

```

```

void dbus_connection_set_max_message_size (DBusConnection
*connection,
                                         long          size);

DBUS_EXPORT
long dbus_connection_get_max_message_size (DBusConnection
*connection);
DBUS_EXPORT
void dbus_connection_set_max_received_size (DBusConnection
*connection,
                                         long          size);

DBUS_EXPORT
long dbus_connection_get_max_received_size (DBusConnection
*connection);

DBUS_EXPORT
void dbus_connection_set_max_message_unix_fds (DBusConnection
*connection,
                                              long          n);

DBUS_EXPORT
long dbus_connection_get_max_message_unix_fds (DBusConnection
*connection);
DBUS_EXPORT
void dbus_connection_set_max_received_unix_fds (DBusConnection
*connection,
                                              long          n);

DBUS_EXPORT
long dbus_connection_get_max_received_unix_fds (DBusConnection
*connection);

DBUS_EXPORT
long dbus_connection_get_outgoing_size      (DBusConnection
*connection);
DBUS_EXPORT
long dbus_connection_get_outgoing_unix_fds  (DBusConnection
*connection);

DBUS_EXPORT
DBusPreallocatedSend* dbus_connection_preallocate_send
(DBusConnection      *connection);
DBUS_EXPORT
void                  dbus_connection_free_preallocated_send
(DBusConnection      *connection,

DBusPreallocatedSend *preallocated);
DBUS_EXPORT
void                  dbus_connection_send_preallocated
(DBusConnection      *connection,

DBusPreallocatedSend *preallocated,

DBusMessage          *message,

```



```

dbus_uint32_t      *client_serial);

/* Object tree functionality */

/**
 * Called when a #DBusObjectPathVTable is unregistered (or its
 connection is freed).
 * Found in #DBusObjectPathVTable.
 */
typedef void      (* DBusObjectPathUnregisterFunction)
(DBusConnection  *connection,

                                                         void
 *user_data);
/**
 * Called when a message is sent to a registered object path. Found in
 * #DBusObjectPathVTable which is registered with
 dbus_connection_register_object_path()
 * or dbus_connection_register_fallback().
 */
typedef DBusHandlerResult (* DBusObjectPathMessageFunction)
(DBusConnection  *connection,

DBusMessage      *message,

                                                         void
 *user_data);

/**
 * Virtual table that must be implemented to handle a portion of the
 * object path hierarchy. Attach the vtable to a particular path using
 * dbus_connection_register_object_path() or
 * dbus_connection_register_fallback().
 */
struct DBusObjectPathVTable
{
    DBusObjectPathUnregisterFunction  unregister_function; /**<
Function to unregister this handler */
    DBusObjectPathMessageFunction     message_function; /**< Function
to handle messages */

    void (* dbus_internal_pad1) (void *); /**< Reserved for future
expansion */
    void (* dbus_internal_pad2) (void *); /**< Reserved for future
expansion */
    void (* dbus_internal_pad3) (void *); /**< Reserved for future
expansion */
    void (* dbus_internal_pad4) (void *); /**< Reserved for future
expansion */
};

DBUS_EXPORT

```

```

dbus_bool_t dbus_connection_try_register_object_path (DBusConnection
*connection,
                                                    const char
*path,
                                                    const
DBusObjectPathVTable *vtable,
                                                    void
*user_data,
                                                    DBusError
*error);

DBUS_EXPORT
dbus_bool_t dbus_connection_register_object_path (DBusConnection
*connection,
                                                    const char
*path,
                                                    const
DBusObjectPathVTable *vtable,
                                                    void
*user_data);

DBUS_EXPORT
dbus_bool_t dbus_connection_try_register_fallback (DBusConnection
*connection,
                                                    const char
*path,
                                                    const
DBusObjectPathVTable *vtable,
                                                    void
*user_data,
                                                    DBusError
*error);

DBUS_EXPORT
dbus_bool_t dbus_connection_register_fallback (DBusConnection
*connection,
                                                    const char
*path,
                                                    const
DBusObjectPathVTable *vtable,
                                                    void
*user_data);

DBUS_EXPORT
dbus_bool_t dbus_connection_unregister_object_path (DBusConnection
*connection,
                                                    const char
*path);

DBUS_EXPORT
dbus_bool_t dbus_connection_get_object_path_data (DBusConnection
*connection,

```

```

const char
*path,
**data_p);

DBUS_EXPORT
dbus_bool_t dbus_connection_list_registered (DBusConnection
*connection,
const char
*parent_path,
char
***child_entries);

DBUS_EXPORT
dbus_bool_t dbus_connection_get_unix_fd (DBusConnection
*connection,
int
*fd);
DBUS_EXPORT
dbus_bool_t dbus_connection_get_socket (DBusConnection
*connection,
int
*fd);

/** @} */

/**
 * @addtogroup DBusWatch
 * @{
 */

#ifdef DBUS_DISABLE_DEPRECATED
DBUS_EXPORT
DBUS_DEPRECATED int dbus_watch_get_fd (DBusWatch *watch);
#endif

DBUS_EXPORT
int dbus_watch_get_unix_fd (DBusWatch *watch);
DBUS_EXPORT
int dbus_watch_get_socket (DBusWatch *watch);
DBUS_EXPORT
unsigned int dbus_watch_get_flags (DBusWatch *watch);
DBUS_EXPORT
void* dbus_watch_get_data (DBusWatch *watch);
DBUS_EXPORT
void dbus_watch_set_data (DBusWatch *watch,
void *data,
DBusFreeFunction
free_data_function);
DBUS_EXPORT
dbus_bool_t dbus_watch_handle (DBusWatch *watch,

```

```

                                unsigned int    flags);
DBUS_EXPORT
dbus_bool_t  dbus_watch_get_enabled (DBusWatch    *watch);

/** @} */

/**
 * @addtogroup DBusTimeout
 * @{
 */

DBUS_EXPORT
int  dbus_timeout_get_interval (DBusTimeout    *timeout);
DBUS_EXPORT
void* dbus_timeout_get_data    (DBusTimeout    *timeout);
DBUS_EXPORT
void  dbus_timeout_set_data    (DBusTimeout    *timeout,
                                void            *data,
                                DBusFreeFunction
                                free_data_function);
DBUS_EXPORT
dbus_bool_t  dbus_timeout_handle    (DBusTimeout    *timeout);
DBUS_EXPORT
dbus_bool_t  dbus_timeout_get_enabled (DBusTimeout    *timeout);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_CONNECTION_H */

File = dbus-credentials-util.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-credentials-util.c Would be in dbus-credentials.c, but only
used for tests/bus
 *
 * Copyright (C) 2007 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of

```

```
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-internals.h"
#include "dbus-test.h"
#include "dbus-credentials.h"
```

```
/**
 * @addtogroup DBusCredentials
 * @{
 */
```

```
/** @} */
```

```
#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>
#include <string.h>
```

```
static DBusCredentials*
make_credentials(dbus_uid_t  unix_uid,
                 dbus_pid_t  unix_pid,
                 const char *windows_sid)
{
    DBusCredentials *credentials;

    credentials = _dbus_credentials_new ();

    if (unix_uid != DBUS_UID_UNSET)
    {
        if (!_dbus_credentials_add_unix_uid (credentials, unix_uid))
        {
            _dbus_credentials_unref (credentials);
            return NULL;
        }
    }

    if (unix_pid != DBUS_PID_UNSET)
    {
        if (!_dbus_credentials_add_unix_pid (credentials, unix_pid))
        {
            _dbus_credentials_unref (credentials);
            return NULL;
        }
    }
}
```

```

    }

    if (windows_sid != NULL)
    {
        if (!_dbus_credentials_add_windows_sid (credentials,
windows_sid))
        {
            _dbus_credentials_unref (credentials);
            return NULL;
        }
    }

    return credentials;
}

#define SAMPLE_SID "whatever a windows sid looks like"
#define OTHER_SAMPLE_SID "whatever else"

dbus_bool_t
_dbus_credentials_test (const char *test_data_dir)
{
    DBusCredentials *creds;
    DBusCredentials *creds2;

    if (test_data_dir == NULL)
        return TRUE;

    creds = make_credentials (12, 511, SAMPLE_SID);
    if (creds == NULL)
        _dbus_assert_not_reached ("oom");

    /* test refcounting */
    _dbus_credentials_ref (creds);
    _dbus_credentials_unref (creds);

    _dbus_assert (_dbus_credentials_include (creds,
DBUS_CREDENTIAL_UNIX_USER_ID));
    _dbus_assert (_dbus_credentials_include (creds,
DBUS_CREDENTIAL_UNIX_PROCESS_ID));
    _dbus_assert (_dbus_credentials_include (creds,
DBUS_CREDENTIAL_WINDOWS_SID));

    _dbus_assert (_dbus_credentials_get_unix_uid (creds) == 12);
    _dbus_assert (_dbus_credentials_get_unix_pid (creds) == 511);
    _dbus_assert (strcmp (_dbus_credentials_get_windows_sid (creds),
SAMPLE_SID) == 0);

    _dbus_assert (!_dbus_credentials_are_empty (creds));
    _dbus_assert (!_dbus_credentials_are_anonymous (creds));

    /* Test copy */
    creds2 = _dbus_credentials_copy (creds);

```

```

if (creds2 == NULL)
    _dbus_assert_not_reached ("oom");

    _dbus_assert (_dbus_credentials_include (creds2,
DBUS_CREDENTIAL_UNIX_USER_ID));
    _dbus_assert (_dbus_credentials_include (creds2,
DBUS_CREDENTIAL_UNIX_PROCESS_ID));
    _dbus_assert (_dbus_credentials_include (creds2,
DBUS_CREDENTIAL_WINDOWS_SID));

    _dbus_assert (_dbus_credentials_get_unix_uid (creds2) == 12);
    _dbus_assert (_dbus_credentials_get_unix_pid (creds2) == 511);
    _dbus_assert (strcmp (_dbus_credentials_get_windows_sid (creds2),
SAMPLE_SID) == 0);

    _dbus_assert (_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

/* Same user if both unix and windows are the same */
creds2 = make_credentials (12, DBUS_PID_UNSET, SAMPLE_SID);
if (creds2 == NULL)
    _dbus_assert_not_reached ("oom");

    _dbus_assert (_dbus_credentials_same_user (creds, creds2));

    _dbus_credentials_unref (creds2);

/* Not the same user if Windows is missing */
creds2 = make_credentials (12, DBUS_PID_UNSET, NULL);
if (creds2 == NULL)
    _dbus_assert_not_reached ("oom");

    _dbus_assert (!_dbus_credentials_same_user (creds, creds2));
    _dbus_assert (_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

/* Not the same user if Windows is different */
creds2 = make_credentials (12, DBUS_PID_UNSET, OTHER_SAMPLE_SID);
if (creds2 == NULL)
    _dbus_assert_not_reached ("oom");

    _dbus_assert (!_dbus_credentials_same_user (creds, creds2));
    _dbus_assert (!_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

/* Not the same user if Unix is missing */
creds2 = make_credentials (DBUS_UID_UNSET, DBUS_PID_UNSET,
SAMPLE_SID);
if (creds2 == NULL)

```

```

    _dbus_assert_not_reached ("oom");

    _dbus_assert (!_dbus_credentials_same_user (creds, creds2));
    _dbus_assert (_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

    /* Not the same user if Unix is different */
    creds2 = make_credentials (15, DBUS_PID_UNSET, SAMPLE_SID);
    if (creds2 == NULL)
        _dbus_assert_not_reached ("oom");

    _dbus_assert (!_dbus_credentials_same_user (creds, creds2));
    _dbus_assert (!_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

    /* Not the same user if both are missing */
    creds2 = make_credentials (DBUS_UID_UNSET, DBUS_PID_UNSET, NULL);
    if (creds2 == NULL)
        _dbus_assert_not_reached ("oom");

    _dbus_assert (!_dbus_credentials_same_user (creds, creds2));
    _dbus_assert (_dbus_credentials_are_superset (creds, creds2));

    _dbus_credentials_unref (creds2);

    /* Clearing credentials works */
    _dbus_credentials_clear (creds);

    _dbus_assert (!_dbus_credentials_include (creds,
DBUS_CREDENTIAL_UNIX_USER_ID));
    _dbus_assert (!_dbus_credentials_include (creds,
DBUS_CREDENTIAL_UNIX_PROCESS_ID));
    _dbus_assert (!_dbus_credentials_include (creds,
DBUS_CREDENTIAL_WINDOWS_SID));

    _dbus_assert (_dbus_credentials_get_unix_uid (creds) ==
DBUS_UID_UNSET);
    _dbus_assert (_dbus_credentials_get_unix_pid (creds) ==
DBUS_PID_UNSET);
    _dbus_assert (_dbus_credentials_get_windows_sid (creds) == NULL);

    _dbus_assert (_dbus_credentials_are_empty (creds));
    _dbus_assert (_dbus_credentials_are_anonymous (creds));

    _dbus_credentials_unref (creds);

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```


File = dbus-credentials.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-credentials.c Credentials provable through authentication
 *
 * Copyright (C) 2007 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */
#include <config.h>
#include <string.h>
#include "dbus-credentials.h"
#include "dbus-internals.h"

/**
 * @defgroup DBusCredentials Credentials provable through
authentication
 * @ingroup DBusInternals
 * @brief DBusCredentials object
 *
 * Credentials are what you have to prove you have in order to
 * authenticate. The main credentials right now are a unix user
 * account, a Windows user account, or a UNIX process ID.
 */

/**
 * @defgroup DBusCredentialsInternals Credentials implementation
details
 * @ingroup DBusInternals
 * @brief DBusCredentials implementation details
 */
```

```

* Private details of credentials code.
*
* @{}
*/

struct DBusCredentials {
    int refcount;
    dbus_uid_t unix_uid;
    dbus_pid_t unix_pid;
    char *windows_sid;
    void *adt_audit_data;
    dbus_int32_t adt_audit_data_size;
};

/** @} */

/**
 * @addtogroup DBusCredentials
 * @{}
 */

/**
 * Creates a new credentials object.
 *
 * @returns the new object or #NULL if no memory
 */
DBusCredentials*
_dbus_credentials_new (void)
{
    DBusCredentials *creds;

    creds = dbus_new (DBusCredentials, 1);
    if (creds == NULL)
        return NULL;

    creds->refcount = 1;
    creds->unix_uid = DBUS_UID_UNSET;
    creds->unix_pid = DBUS_PID_UNSET;
    creds->windows_sid = NULL;
    creds->adt_audit_data = NULL;
    creds->adt_audit_data_size = 0;

    return creds;
}

/**
 * Creates a new object with credentials (user ID and process ID) from
the current process.
 * @returns the new object or #NULL if no memory
 */
DBusCredentials*
_dbus_credentials_new_from_current_process (void)

```

```

{
    DBusCredentials *creds;

    creds = _dbus_credentials_new ();
    if (creds == NULL)
        return NULL;

    if (!_dbus_credentials_add_from_current_process (creds))
    {
        _dbus_credentials_unref (creds);
        return NULL;
    }

    return creds;
}

/**
 * Increment refcount on credentials.
 *
 * @param credentials the object
 */
void
_dbus_credentials_ref (DBusCredentials *credentials)
{
    _dbus_assert (credentials->refcount > 0);
    credentials->refcount += 1;
}

/**
 * Decrement refcount on credentials.
 *
 * @param credentials the object
 */
void
_dbus_credentials_unref (DBusCredentials *credentials)
{
    _dbus_assert (credentials->refcount > 0);

    credentials->refcount -= 1;
    if (credentials->refcount == 0)
    {
        dbus_free (credentials->windows_sid);
        dbus_free (credentials->adt_audit_data);
        dbus_free (credentials);
    }
}

/**
 * Add a UNIX process ID to the credentials.
 *
 * @param credentials the object
 * @param pid the process ID

```

```

    * @returns #FALSE if no memory
    */
dbus_bool_t
_dbus_credentials_add_unix_pid (DBusCredentials      *credentials,
                               dbus_pid_t           pid)
{
    credentials->unix_pid = pid;
    return TRUE;
}

/**
 * Add a UNIX user ID to the credentials.
 *
 * @param credentials the object
 * @param uid the user ID
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_credentials_add_unix_uid (DBusCredentials      *credentials,
                               dbus_uid_t           uid)
{
    credentials->unix_uid = uid;
    return TRUE;
}

/**
 * Add a Windows user SID to the credentials.
 *
 * @param credentials the object
 * @param windows_sid the user SID
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_credentials_add_windows_sid (DBusCredentials      *credentials,
                                   const char            *windows_sid)
{
    char *copy;

    copy = _dbus_strdup (windows_sid);
    if (copy == NULL)
        return FALSE;

    dbus_free (credentials->windows_sid);
    credentials->windows_sid = copy;

    return TRUE;
}

/**
 * Add ADT audit data to the credentials.
 *

```

```

* @param credentials the object
* @param audit_data the audit data
* @param size the length of audit data
* @returns #FALSE if no memory
*/
dbus_bool_t
_dbus_credentials_add_adt_audit_data (DBusCredentials      *credentials,
                                      void                  *audit_data,
                                      dbus_int32_t          size)
{
    void *copy;
    copy = _dbus_memdup (audit_data, size);
    if (copy == NULL)
        return FALSE;

    dbus_free (credentials->adt_audit_data);
    credentials->adt_audit_data = copy;
    credentials->adt_audit_data_size = size;

    return TRUE;
}

/**
 * Checks whether the given credential is present.
 *
 * @param credentials the object
 * @param type the credential to check for
 * @returns #TRUE if the credential is present
 */
dbus_bool_t
_dbus_credentials_include (DBusCredentials      *credentials,
                          DBusCredentialType  type)
{
    switch (type)
    {
        case DBUS_CREDENTIAL_UNIX_PROCESS_ID:
            return credentials->unix_pid != DBUS_PID_UNSET;
        case DBUS_CREDENTIAL_UNIX_USER_ID:
            return credentials->unix_uid != DBUS_UID_UNSET;
        case DBUS_CREDENTIAL_WINDOWS_SID:
            return credentials->windows_sid != NULL;
        case DBUS_CREDENTIAL_ADT_AUDIT_DATA_ID:
            return credentials->adt_audit_data != NULL;
    }

    _dbus_assert_not_reached ("Unknown credential enum value");
    return FALSE;
}

/**
 * Gets the UNIX process ID in the credentials, or #DBUS_PID_UNSET if
 * the credentials object doesn't contain a process ID.

```

```

*
* @param credentials the object
* @returns UNIX process ID
*/
dbus_pid_t
_dbus_credentials_get_unix_pid (DBusCredentials      *credentials)
{
    return credentials->unix_pid;
}

/**
 * Gets the UNIX user ID in the credentials, or #DBUS_UID_UNSET if
 * the credentials object doesn't contain a user ID.
 *
 * @param credentials the object
 * @returns UNIX user ID
 */
dbus_uid_t
_dbus_credentials_get_unix_uid (DBusCredentials      *credentials)
{
    return credentials->unix_uid;
}

/**
 * Gets the Windows user SID in the credentials, or #NULL if
 * the credentials object doesn't contain a Windows user SID.
 *
 * @param credentials the object
 * @returns Windows user SID
 */
const char*
_dbus_credentials_get_windows_sid (DBusCredentials      *credentials)
{
    return credentials->windows_sid;
}

/**
 * Gets the ADT audit data in the credentials, or #NULL if
 * the credentials object doesn't contain ADT audit data.
 *
 * @param credentials the object
 * @returns Solaris ADT audit data
 */
void *
_dbus_credentials_get_adt_audit_data (DBusCredentials      *credentials)
{
    return credentials->adt_audit_data;
}

/**
 * Gets the ADT audit data size in the credentials, or 0 if
 * the credentials object doesn't contain ADT audit data.

```

```

*
* @param credentials the object
* @returns Solaris ADT audit data size
*/
dbus_int32_t
_dbus_credentials_get_adt_audit_data_size (DBusCredentials
*credentials)
{
    return credentials->adt_audit_data_size;
}

/**
* Checks whether the first credentials object contains
* all the credentials found in the second credentials object.
*
* @param credentials the object
* @param possible_subset see if credentials in here are also in the
first arg
* @returns #TRUE if second arg is contained in first
*/
dbus_bool_t
_dbus_credentials_are_superset (DBusCredentials    *credentials,
                               DBusCredentials    *possible_subset)
{
    return
        (possible_subset->unix_pid == DBUS_PID_UNSET ||
         possible_subset->unix_pid == credentials->unix_pid) &&
        (possible_subset->unix_uid == DBUS_UID_UNSET ||
         possible_subset->unix_uid == credentials->unix_uid) &&
        (possible_subset->windows_sid == NULL ||
         (credentials->windows_sid && strcmp (possible_subset-
>windows_sid,
                                               credentials->windows_sid) ==
0)) &&
        (possible_subset->adt_audit_data == NULL ||
         (credentials->adt_audit_data && memcmp (possible_subset-
>adt_audit_data,
                                               credentials-
>adt_audit_data,
                                               credentials-
>adt_audit_data_size) == 0));
}

/**
* Checks whether a credentials object contains anything.
*
* @param credentials the object
* @returns #TRUE if there are no credentials in the object
*/
dbus_bool_t
_dbus_credentials_are_empty (DBusCredentials    *credentials)
{

```



```

* Merge the given credential found in the second object into the
first object,
* overwriting the first object's value for that credential.
*
* Does nothing if the second object does not contain the specified
credential.
* i.e., will never delete a credential from the first object.
*
* @param credentials the object
* @param which the credential to overwrite
* @param other_credentials credentials to merge
* @returns #FALSE if no memory
*/
dbus_bool_t
_dbus_credentials_add_credential (DBusCredentials *credentials,
                                 DBusCredentialType which,
                                 DBusCredentials
*other_credentials)
{
    if (which == DBUS_CREDENTIAL_UNIX_PROCESS_ID &&
        other_credentials->unix_pid != DBUS_PID_UNSET)
    {
        if (!_dbus_credentials_add_unix_pid (credentials,
other_credentials->unix_pid))
            return FALSE;
    }
    else if (which == DBUS_CREDENTIAL_UNIX_USER_ID &&
             other_credentials->unix_uid != DBUS_UID_UNSET)
    {
        if (!_dbus_credentials_add_unix_uid (credentials,
other_credentials->unix_uid))
            return FALSE;
    }
    else if (which == DBUS_CREDENTIAL_WINDOWS_SID &&
             other_credentials->windows_sid != NULL)
    {
        if (!_dbus_credentials_add_windows_sid (credentials,
other_credentials->windows_sid))
            return FALSE;
    }
    else if (which == DBUS_CREDENTIAL_ADT_AUDIT_DATA_ID &&
             other_credentials->adt_audit_data != NULL)
    {
        if (!_dbus_credentials_add_adt_audit_data (credentials,
other_credentials->adt_audit_data, other_credentials-
>adt_audit_data_size))
            return FALSE;
    }

    return TRUE;
}

```

```

/**
 * Clear all credentials in the object.
 *
 * @param credentials the object
 */
void
_dbus_credentials_clear (DBusCredentials *credentials)
{
    credentials->unix_pid = DBUS_PID_UNSET;
    credentials->unix_uid = DBUS_UID_UNSET;
    dbus_free (credentials->windows_sid);
    credentials->windows_sid = NULL;
    dbus_free (credentials->adt_audit_data);
    credentials->adt_audit_data = NULL;
    credentials->adt_audit_data_size = 0;
}

/**
 * Copy a credentials object.
 *
 * @param credentials the object
 * @returns the copy or #NULL
 */
DBusCredentials*
_dbus_credentials_copy (DBusCredentials *credentials)
{
    DBusCredentials *copy;

    copy = _dbus_credentials_new ();
    if (copy == NULL)
        return NULL;

    if (!_dbus_credentials_add_credentials (copy, credentials))
    {
        _dbus_credentials_unref (copy);
        return NULL;
    }

    return copy;
}

/**
 * Check whether the user-identifying credentials in two credentials
 * objects are identical. Credentials that are not related to the
 * user are ignored, but any kind of user ID credentials must be the
 * same (UNIX user ID, Windows user SID, etc.) and present in both
 * objects for the function to return #TRUE.
 *
 * @param credentials the object
 * @param other_credentials credentials to compare
 * @returns #TRUE if the two credentials refer to the same user
 */

```

```

dbus_bool_t
_dbus_credentials_same_user (DBusCredentials      *credentials,
                            DBusCredentials      *other_credentials)
{
    /* both windows and unix user must be the same (though pretty much
     * in all conceivable cases, one will be unset)
     */
    return credentials->unix_uid == other_credentials->unix_uid &&
        ((!(credentials->windows_sid || other_credentials->windows_sid)
||
        (credentials->windows_sid && other_credentials->windows_sid &&
        strcmp (credentials->windows_sid, other_credentials-
>windows_sid) == 0));
}

/**
 * Convert the credentials in this object to a human-readable
 * string format, and append to the given string.
 *
 * @param credentials the object
 * @param string append to this string
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_credentials_to_string_append (DBusCredentials      *credentials,
                                  DBusString              *string)
{
    dbus_bool_t join;

    join = FALSE;
    if (credentials->unix_uid != DBUS_UID_UNSET)
        {
            if (!_dbus_string_append_printf (string, "uid=" DBUS_UID_FORMAT,
credentials->unix_uid))
                goto oom;
            join = TRUE;
        }
    if (credentials->unix_pid != DBUS_PID_UNSET)
        {
            if (!_dbus_string_append_printf (string, "%spid="
DBUS_PID_FORMAT, join ? " " : "", credentials->unix_pid))
                goto oom;
            join = TRUE;
        }
    else
        join = FALSE;
    if (credentials->windows_sid != NULL)
        {
            if (!_dbus_string_append_printf (string, "%ssid=%s", join ? " "
: "", credentials->windows_sid))
                goto oom;
            join = TRUE;
        }
}

```

```

    }
else
    join = FALSE;

    return TRUE;
oom:
    return FALSE;
}

/** @} */

/* tests in dbus-credentials-util.c */

File = dbus-credentials.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-credentials.h Credentials provable through authentication
 *
 * Copyright (C) 2007 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifndef DBUS_CREDENTIALS_H
#define DBUS_CREDENTIALS_H

#include <dbus/dbus-macros.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-sysdeps.h>

DBUS_BEGIN_DECLS

```

```

typedef enum {
    DBUS_CREDENTIAL_UNIX_PROCESS_ID,
    DBUS_CREDENTIAL_UNIX_USER_ID,
    DBUS_CREDENTIAL_ADT_AUDIT_DATA_ID,
    DBUS_CREDENTIAL_WINDOWS_SID
} DBusCredentialType;

DBusCredentials* _dbus_credentials_new_from_current_process (void);
DBusCredentials* _dbus_credentials_new (void);
void
_dbus_credentials_ref
(DBusCredentials *credentials);
void
_dbus_credentials_unref
(DBusCredentials *credentials);
dbus_bool_t
_dbus_credentials_add_unix_pid
(DBusCredentials *credentials,

dbus_pid_t pid);
dbus_bool_t
_dbus_credentials_add_unix_uid
(DBusCredentials *credentials,

dbus_uid_t uid);
dbus_bool_t
_dbus_credentials_add_windows_sid
(DBusCredentials *credentials,

char *windows_sid);
dbus_bool_t
_dbus_credentials_add_adt_audit_data
(DBusCredentials *credentials,

*audit_data,

dbus_int32_t size);
dbus_bool_t
_dbus_credentials_include
(DBusCredentials *credentials,

DBusCredentialType type);
dbus_pid_t
_dbus_credentials_get_unix_pid
(DBusCredentials *credentials);
dbus_uid_t
_dbus_credentials_get_unix_uid
(DBusCredentials *credentials);
const char*
_dbus_credentials_get_windows_sid
(DBusCredentials *credentials);
void *
_dbus_credentials_get_adt_audit_data
(DBusCredentials *credentials);
dbus_int32_t
_dbus_credentials_get_adt_audit_data_size
(DBusCredentials *credentials);
dbus_bool_t
_dbus_credentials_are_superset
(DBusCredentials *credentials,

DBusCredentials *possible_subset);
dbus_bool_t
_dbus_credentials_are_empty
(DBusCredentials *credentials);

```

```

dbus_bool_t      _dbus_credentials_are_anonymous
(DBusCredentials *credentials);
dbus_bool_t      _dbus_credentials_add_credentials
(DBusCredentials *credentials,

DBusCredentials *other_credentials);
/* must silently allow 'which' to not exist */
dbus_bool_t      _dbus_credentials_add_credential
(DBusCredentials *credentials,

DBusCredentialType which,

DBusCredentials *other_credentials);
void             _dbus_credentials_clear
(DBusCredentials *credentials);
DBusCredentials* _dbus_credentials_copy
(DBusCredentials *credentials);
dbus_bool_t      _dbus_credentials_same_user
(DBusCredentials *credentials,

DBusCredentials *other_credentials);
dbus_bool_t      _dbus_credentials_to_string_append
(DBusCredentials *credentials,

DBusString      *string);

DBUS_END_DECLS

#endif /* DBUS_CREDENTIALS_H */

```

File = dbus-daemon.1

```

.\"
.\" dbus\--daemon manual page.
.\" Copyright (C) 2003,2008 Red Hat, Inc.
.\"
.TH dbus\--daemon 1
.SH NAME
dbus\--daemon \-- Message bus daemon
.SH SYNOPSIS
.PP
.B dbus\--daemon
dbus\--daemon [\--version] [\--session] [\--system] [\--config\--
file=FILE]
[\--print\--address[=DESCRIPTOR]] [\--print\--pid[=DESCRIPTOR]] [\--\--
fork]

.SH DESCRIPTION
\fi dbus\--daemon\fp is the D\--Bus message bus daemon. See
http://www.freedesktop.org/software/dbus/ for more information about

```

the big picture. D\ -Bus is first a library that provides one\ -to\ -one communication between any two applications; \fIdbus\ -daemon\ fP is an application that uses this library to implement a message bus daemon. Multiple programs connect to the message bus daemon and can exchange messages with one another.

.PP

There are two standard message bus instances: the systemwide message bus

(installed on many systems as the "messagebus" init service) and the per\ -user\ -login\ -session message bus (started each time a user logs in).

\fIdbus\ -daemon\ fP is used for both of these instances, but with a different configuration file.

.PP

The \-\ -session option is equivalent to

"\-\ -config\ -file=/etc/dbus\ -1/session.conf" and the \-\ -system option is equivalent to

"\-\ -config\ -file=/etc/dbus\ -1/system.conf". By creating additional configuration files and using the \-\ -config\ -file option, additional special\ -purpose message bus daemons could be created.

.PP

The systemwide daemon is normally launched by an init script, standardly called simply "messagebus".

.PP

The systemwide daemon is largely used for broadcasting system events, such as changes to the printer queue, or adding/removing devices.

.PP

The per\ -session daemon is used for various interprocess communication among desktop applications (however, it is not tied to X or the GUI in any way).

.PP

SIGHUP will cause the D\ -Bus daemon to PARTIALLY reload its configuration file and to flush its user/group information caches.

Some configuration changes would require kicking all apps off the bus; so they will

only take effect if you restart the daemon. Policy changes should take effect

with SIGHUP.

.SH OPTIONS

The following options are supported:

.TP

.I "\-\ -config\ -file=FILE"

Use the given configuration file.

.TP

.I "\-\ -fork"

Force the message bus to fork and become a daemon, even if the configuration file does not specify that it should.

In most contexts the configuration file already gets this right, though.

.I "\-\ -nofork"

Force the message bus not to fork and become a daemon, even if the configuration file specifies that it should.

.TP

.I "\-\'-print\'-address[=DESCRIPTOR]"

Print the address of the message bus to standard output, or to the given file descriptor. This is used by programs that launch the message bus.

.TP

.I "\-\'-print\'-pid[=DESCRIPTOR]"

Print the process ID of the message bus to standard output, or to the given file descriptor. This is used by programs that launch the message bus.

.TP

.I "\-\'-session"

Use the standard configuration file for the per\'-login\'-session message bus.

.TP

.I "\-\'-system"

Use the standard configuration file for the systemwide message bus.

.TP

.I "\-\'-version"

Print the version of the daemon.

.TP

.I "\-\'-introspect"

Print the introspection information for all D\'-Bus internal interfaces.

.TP

.I "\-\'-address[=ADDRESS]"

Set the address to listen on. This option overrides the address configured in the configuration file.

.TP

.I "\-\'-systemd\'-activation"

Enable systemd\'-style service activation. Only useful in conjunction with the systemd system and session manager on Linux.

.TP

.I "\-\'-nopicfile"

Don't write a PID file even if one is configured in the configuration files.

.SH CONFIGURATION FILE

A message bus daemon has a configuration file that specializes it for a particular application. For example, one configuration file might set up the message bus to be a systemwide message bus, while another might set it up to be a per\'-user\'-login\'-session bus.

.PP

The configuration file also establishes resource limits, security parameters, and so forth.

.PP

The configuration file is not part of any interoperability specification and its backward compatibility is not guaranteed; this

document is documentation, not specification.

.PP

The standard systemwide and per-session message bus setups are configured in the files `"/etc/dbus-1/system.conf"` and `"/etc/dbus-1/session.conf"`. These files normally `<include>` a `system-1-local.conf` or `session-1-local.conf`; you can put local overrides in those files to avoid modifying the primary configuration files.

.PP

The configuration file is an XML document. It must have the following doctype declaration:

.nf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-Bus Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
```

.fi

.PP

The following elements may be present in the configuration file.

.TP

.I "<busconfig>"

.PP

Root element.

.TP

.I "<type>"

.PP

The well-known type of the message bus. Currently known values are "system" and "session"; if other values are set, they should be either added to the D-Bus specification, or namespaced. The last `<type>` element "wins" (previous values are ignored). This element only controls which message bus specific environment variables are set in activated clients. Most of the policy that distinguishes a session bus from the system bus is controlled from the other elements in the configuration file.

.PP

If the well-known type of the message bus is "session", then the `DBUS_STARTER_BUS_TYPE` environment variable will be set to "session" and the `DBUS_SESSION_BUS_ADDRESS` environment variable will be set to the address of the session bus. Likewise, if the type of the message bus is "system", then the `DBUS_STARTER_BUS_TYPE` environment variable will be set to "system" and the `DBUS_SESSION_BUS_ADDRESS` environment variable will be set to the address of the system bus (which is normally well known anyway).

.PP
Example: <type>session</type>

.TP
.I "<include>"

.PP
Include a file <include>filename.conf</include> at this point. If the filename is relative, it is located relative to the configuration file doing the including.

.PP
<include> has an optional attribute "ignore_missing=(yes|no)" which defaults to "no" if not provided. This attribute controls whether it's a fatal error for the included file to be absent.

.TP
.I "<includedir>"

.PP
Include all files in <includedir>foo.d</includedir> at this point. Files in the directory are included in undefined order. Only files ending in ".conf" are included.

.PP
This is intended to allow extension of the system bus by particular packages. For example, if CUPS wants to be able to send out notification of printer queue changes, it could install a file to /etc/dbus\-1/system.d that allowed all apps to receive this message and allowed the printer daemon user to send it.

.TP
.I "<user>"

.PP
The user account the daemon should run as, as either a username or a UID. If the daemon cannot change to this UID on startup, it will exit. If this element is not present, the daemon will not change or care about its UID.

.PP
The last <user> entry in the file "wins", the others are ignored.

.PP
The user is changed after the bus has completed initialization. So sockets etc. will be created before changing user, but no data will be read from clients before changing user. This means that sockets and PID files can be created in a location that requires root privileges for writing.

.TP
.I "<fork>"

.PP
If present, the bus daemon becomes a real daemon (forks into the background, etc.). This is generally used rather than the `--fork` command line option.

.TP
.I "<keep_umask>"

.PP
If present, the bus daemon keeps its original umask when forking. This may be useful to avoid affecting the behavior of child processes.

.TP
.I "<listen>"

.PP
Add an address that the bus should listen on. The address is in the standard D\--Bus format that contains a transport name plus possible parameters/options.

.PP
Example: `<listen>unix:path=/tmp/foo</listen>`

.PP
Example: `<listen>tcp:host=localhost,port=1234</listen>`

.PP
If there are multiple `<listen>` elements, then the bus listens on multiple addresses. The bus will pass its address to started services or other interested parties with the last address given in `<listen>` first. That is, apps will try to connect to the last `<listen>` address first.

.PP
tcp sockets can accept IPv4 addresses, IPv6 addresses or hostnames. If a hostname resolves to multiple addresses, the server will bind to all of them. The `family=ipv4` or `family=ipv6` options can be used to force it to bind to a subset of addresses

.PP
Example: `<listen>tcp:host=localhost,port=0,family=ipv4</listen>`

.PP
A special case is using a port number of zero (or omitting the port), which means to choose an available port selected by the operating system. The port number chosen can be obtained with the `--print-address` command line parameter and will be present in other cases where the server reports its own address, such as when `DBUS_SESSION_BUS_ADDRESS` is set.

.PP

Example: <listen>tcp:host=localhost,port=0</listen>

.PP

tcp addresses also allow a bind=hostname option, which will override the host option specifying what address to bind to, without changing the address reported by the bus. The bind option can also take a special name '*' to cause the bus to listen on all local address (INADDR_ANY). The specified host should be a valid name of the local machine or weird stuff will happen.

.PP

Example: <listen>tcp:host=localhost,bind=*,port=0</listen>

.TP

.I "<auth>"

.PP

Lists permitted authorization mechanisms. If this element doesn't exist, then all known mechanisms are allowed. If there are multiple <auth> elements, all the listed mechanisms are allowed. The order in which mechanisms are listed is not meaningful.

.PP

Example: <auth>EXTERNAL</auth>

.PP

Example: <auth>DBUS_COOKIE_SHA1</auth>

.TP

.I "<servicedir>"

.PP

Adds a directory to scan for .service files. Directories are scanned starting with the last to appear in the config file (the first .service file found that provides a particular service will be used).

.PP

Service files tell the bus how to automatically start a program. They are primarily used with the per\ -user\ -session bus, not the systemwide bus.

.TP

.I "<standard_session_servicedirs/>"

.PP

<standard_session_servicedirs/> is equivalent to specifying a series of <servicedir/> elements for each of the data directories in the "XDG Base Directory Specification" with the subdirectory "dbus\ -1/services",

so for example `"/usr/share/dbus-1/services"` would be among the directories searched.

.PP

The "XDG Base Directory Specification" can be found at <http://freedesktop.org/wiki/Standards/basedir-spec> if it hasn't moved, otherwise try your favorite search engine.

.PP

The `<standard_session_servicedirs/>` option is only relevant to the `per\-user\-session bus` daemon defined in `/etc/dbus-1/session.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "`<standard_system_servicedirs/>`"

.PP

`<standard_system_servicedirs/>` specifies the standard system-wide activation directories that should be searched for service files. This option defaults to `/usr/share/dbus-1/system-services`.

.PP

The `<standard_system_servicedirs/>` option is only relevant to the `per\-system bus` daemon defined in `/etc/dbus-1/system.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "`<servicehelper/>`"

.PP

`<servicehelper/>` specifies the setuid helper that is used to launch system daemons with an alternate user. Typically this should be the `dbus-daemon-launch-helper` executable located in `libexec`.

.PP

The `<servicehelper/>` option is only relevant to the `per\-system bus` daemon defined in `/etc/dbus-1/system.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "`<limit>`"

.PP

`<limit>` establishes a resource limit. For example:

.nf

```
<limit name="max_message_size">64</limit>
```

```
<limit name="max_completed_connections">512</limit>
```

.fi

.PP

The name attribute is mandatory.
Available limit names are:

```
.nf
    "max_incoming_bytes"      : total size in bytes of messages
                              incoming from a single connection
    "max_incoming_unix_fds"  : total number of unix fds of
messages
                              incoming from a single connection
    "max_outgoing_bytes"     : total size in bytes of messages
                              queued up for a single connection
    "max_outgoing_unix_fds"  : total number of unix fds of
messages
                              queued up for a single connection
    "max_message_size"       : max size of a single message in
                              bytes
    "max_message_unix_fds"   : max unix fds of a single message
    "service_start_timeout"  : milliseconds (thousandths) until
                              a started service has to connect
    "auth_timeout"           : milliseconds (thousandths) a
                              connection is given to
                              authenticate
    "max_completed_connections" : max number of authenticated
connections
    "max_incomplete_connections" : max number of unauthenticated
                              connections
    "max_connections_per_user" : max number of completed
connections from
                              the same user
    "max_pending_service_starts" : max number of service launches in
                              progress at the same time
    "max_names_per_connection" : max number of names a single
                              connection can own
    "max_match_rules_per_connection": max number of match rules for
a single
                              connection
    "max_replies_per_connection" : max number of pending method
                              replies per connection
                              (number of calls\-in\-progress)
    "reply_timeout"          : milliseconds (thousandths)
                              until a method call times out
.fi
```

.PP

The max incoming/outgoing queue sizes allow a new message to be queued if one byte remains below the max. So you can in fact exceed the max by max_message_size.

.PP

max_completed_connections divided by max_connections_per_user is the

number of users that can work together to denial\of\service all other users by using up all connections on the systemwide bus.

.PP

Limits are normally only of interest on the systemwide bus, not the user session buses.

.TP

.I "<policy>"

.PP

The <policy> element defines a security policy to be applied to a particular set of connections to the bus. A policy is made up of <allow> and <deny> elements. Policies are normally used with the systemwide bus; they are analogous to a firewall in that they allow expected traffic and prevent unexpected traffic.

.PP

Currently, the system bus has a default\deny policy for sending method calls and owning bus names. Everything else, in particular reply messages, receive checks, and signals has a default allow policy.

.PP

In general, it is best to keep system services as small, targeted programs which run in their own process and provide a single bus name. Then, all that is needed is an <allow> rule for the "own" permission to let the process claim the bus name, and a "send_destination" rule to allow traffic from some or all uids to your service.

.PP

The <policy> element has one of four attributes:

.nf

```
context="(default|mandatory)"
at_console="(true|false)"
user="username or userid"
group="group name or gid"
```

.fi

.PP

Policies are applied to a connection as follows:

.nf

```
\- all context="default" policies are applied
```

```
\- all group="connection's user's group" policies are applied
   in undefined order
\- all user="connection's auth user" policies are applied
   in undefined order
\- all at_console="true" policies are applied
\- all at_console="false" policies are applied
\- all context="mandatory" policies are applied
.fi
```

```
.PP
Policies applied later will override those applied earlier,
when the policies overlap. Multiple policies with the same
user/group/context are applied in the order they appear
in the config file.
```

```
.TP
.I "<deny>"
.I "<allow>"
```

```
.PP
A <deny> element appears below a <policy> element and prohibits some
action. The <allow> element makes an exception to previous <deny>
statements, and works just like <deny> but with the inverse meaning.
```

```
.PP
The possible attributes of these elements are:
.nf
```

```
send_interface="interface_name"
send_member="method_or_signal_name"
send_error="error_name"
send_destination="name"
send_type="method_call" | "method_return" | "signal" | "error"
send_path="/path/name"

receive_interface="interface_name"
receive_member="method_or_signal_name"
receive_error="error_name"
receive_sender="name"
receive_type="method_call" | "method_return" | "signal" | "error"
receive_path="/path/name"

send_requested_reply="true" | "false"
receive_requested_reply="true" | "false"

eavesdrop="true" | "false"

own="name"
own_prefix="name"
user="username"
group="groupname"
```

```
.fi
```


.PP

Examples:

.nf

```
<deny send_destination="org.freedesktop.Service"
send_interface="org.freedesktop.System" send_member="Reboot"/>
<deny send_destination="org.freedesktop.System"/>
<deny receive_sender="org.freedesktop.System"/>
<deny user="john"/>
<deny group="enemies"/>
```

.fi

.PP

The `<deny>` element's attributes determine whether the deny "matches" a particular action. If it matches, the action is denied (unless later rules in the config file allow it).

.PP

`send_destination` and `receive_sender` rules mean that messages may not be sent to or received from the *owner* of the given name, not that they may not be sent *to that name*. That is, if a connection owns services A, B, C, and sending to A is denied, sending to B or C will not work either.

.PP

The other `send_*` and `receive_*` attributes are purely textual/`by\`-value matches against the given field in the message header.

.PP

"Eavesdropping" occurs when an application receives a message that was explicitly addressed to a name the application does not own, or is a reply to such a message. Eavesdropping thus only applies to messages that are addressed to services and replies to such messages (i.e. it does not apply to signals).

.PP

For `<allow>`, `eavesdrop="true"` indicates that the rule matches even when eavesdropping. `eavesdrop="false"` is the default and means that the rule only allows messages to go to their specified recipient. For `<deny>`, `eavesdrop="true"` indicates that the rule matches only when eavesdropping. `eavesdrop="false"` is the default for `<deny>` also, but here it means that the rule applies always, even when not eavesdropping. The `eavesdrop` attribute can only be combined with `send` and `receive` rules (with `send_*` and `receive_*` attributes).

.PP

The `[send|receive]_requested_reply` attribute works similarly to the `eavesdrop` attribute. It controls whether the `<deny>` or `<allow>` matches a reply that is expected (corresponds to a previous method call message). This attribute only makes sense for reply messages (errors and method returns), and is ignored for other message types.

.PP

For `<allow>`, `[send|receive]_requested_reply="true"` is the default and indicates that only requested replies are allowed by the

rule. [send|receive]_requested_reply="false" means that the rule allows any reply even if unexpected.

.PP

For <deny>, [send|receive]_requested_reply="false" is the default but indicates that the rule matches only when the reply was not requested. [send|receive]_requested_reply="true" indicates that the rule applies always, regardless of pending reply state.

.PP

user and group denials mean that the given user or group may not connect to the message bus.

.PP

For "name", "username", "groupname", etc. the character "*" can be substituted, meaning "any." Complex globs like "foo.bar.*" aren't allowed for now because they'd be work to implement and maybe encourage sloppy security anyway.

.PP

<allow own_prefix="a.b"/> allows you to own the name "a.b" or any name whose first dot-separated elements are "a.b": in particular, you can own "a.b.c" or "a.b.c.d", but not "a.bc" or "a.c". This is useful when services like Telepathy and ReserveDevice define a meaning for subtrees of well-known names, such as org.freedesktop.Telepathy.ConnectionManager.(anything) and org.freedesktop.ReserveDevice1.(anything).

.PP

It does not make sense to deny a user or group inside a <policy> for a user or group; user/group denials can only be inside context="default" or context="mandatory" policies.

.PP

A single <deny> rule may specify combinations of attributes such as send_destination and send_interface and send_type. In this case, the denial applies only if both attributes match the message being denied. e.g. <deny send_interface="foo.bar" send_destination="foo.blah"/> would deny messages with the given interface AND the given bus name. To get an OR effect you specify multiple <deny> rules.

.PP

You can't include both send_ and receive_ attributes on the same rule, since "whether the message can be sent" and "whether it can be received" are evaluated separately.

.PP

Be careful with send_interface/receive_interface, because the

interface field in messages is optional. In particular, do NOT specify `<deny send_interface="org.foo.Bar"/>`! This will cause no\-interface messages to be blocked for all services, which is almost certainly not what you intended. Always use rules of the form: `<deny send_interface="org.foo.Bar" send_destination="org.foo.Service"/>`

.TP
.I "<selinux>"

.PP
The `<selinux>` element contains settings related to Security Enhanced Linux.
More details below.

.TP
.I "<associate>"

.PP
An `<associate>` element appears below an `<selinux>` element and creates a mapping. Right now only one kind of association is possible:
.nf
 `<associate own="org.freedesktop.Foobar" context="foo_t"/>`
.fi

.PP
This means that if a connection asks to own the name "org.freedesktop.Foobar" then the source context will be the context of the connection and the target context will be "foo_t" \- see the short discussion of SELinux below.

.PP
Note, the context here is the target context when requesting a name, NOT the context of the connection owning the name.

.PP
There's currently no way to set a default for owning any name, if we add this syntax it will look like:
.nf
 `<associate own="*" context="foo_t"/>`
.fi
If you find a reason this is useful, let the developers know. Right now the default will be the security context of the bus itself.

.PP
If two `<associate>` elements specify the same name, the element appearing later in the configuration file will be used.

.SH SELinux

.PP

See <http://www.nsa.gov/selinux/> for full details on SELinux. Some useful excerpts:

.IP "" 8

Every subject (process) and object (e.g. file, socket, IPC object, etc) in the system is assigned a collection of security attributes, known as a security context. A security context contains all of the security attributes associated with a particular subject or object that are relevant to the security policy.

.IP "" 8

In order to better encapsulate security contexts and to provide greater efficiency, the policy enforcement code of SELinux typically handles security identifiers (SIDs) rather than security contexts. A SID is an integer that is mapped by the security server to a security context at runtime.

.IP "" 8

When a security decision is required, the policy enforcement code passes a pair of SIDs (typically the SID of a subject and the SID of an object, but sometimes a pair of subject SIDs or a pair of object SIDs), and an object security class to the security server. The object security class indicates the kind of object, e.g. a process, a regular file, a directory, a TCP socket, etc.

.IP "" 8

Access decisions specify whether or not a permission is granted for a given pair of SIDs and class. Each object class has a set of associated permissions defined to control operations on objects with that class.

.PP

D\ -Bus performs SELinux security checks in two places.

.PP

First, any time a message is routed from one connection to another connection, the bus daemon will check permissions with the security context of the first connection as source, security context of the second connection as target, object class "dbus" and requested permission "send_msg".

.PP

If a security context is not available for a connection (impossible when using UNIX domain sockets), then the target context used is the context of the bus daemon itself. There is currently no way to change this default, because we're assuming that only UNIX domain sockets will be used to connect to the systemwide bus. If this changes, we'll probably add a way to set the default connection context.

.PP

Second, any time a connection asks to own a name, the bus daemon will check permissions with the security context of the connection as source, the security context specified for the name in the config file as target, object class "dbus" and requested permission "acquire_svc".

.PP

The security context for a bus name is specified with the <associate> element described earlier in this document. If a name has no security context associated in the configuration file, the security context of the bus daemon itself will be used.

.SH DEBUGGING

.PP

If you're trying to figure out where your messages are going or why you aren't getting messages, there are several things you can try.

.PP

Remember that the system bus is heavily locked down and if you haven't installed a security policy file to allow your message through, it won't work. For the session bus, this is not a concern.

.PP

The simplest way to figure out what's happening on the bus is to run the `\fIdbus\-monitor\` program, which comes with the `D\Bus` package. You can also send test messages with `\fIdbus\-send\`. These programs have their own man pages.

.PP

If you want to know what the daemon itself is doing, you might consider running a separate copy of the daemon to test against. This will allow you to put the daemon under a debugger, or run it with verbose output, without messing up your real session and system daemons.

.PP

To run a separate test copy of the daemon, for example you might open a terminal and type:

.nf

```
DBUS_VERBOSE=1 dbus\-daemon \-\-session \-\-print\-address
```

.fi

.PP

The test daemon address will be printed when the daemon starts. You will need to copy\and\paste this address and use it as the value of the `DBUS_SESSION_BUS_ADDRESS` environment variable when you launch the applications you want to test. This will cause those applications to connect to your test bus instead of the `DBUS_SESSION_BUS_ADDRESS` of your real session bus.

.PP

DBUS_VERBOSE=1 will have NO EFFECT unless your copy of D\Bus was compiled with verbose mode enabled. This is not recommended in production builds due to performance impact. You may need to rebuild D\Bus if your copy was not built with debugging in mind.

(DBUS_VERBOSE

also affects the D\Bus library and thus applications using D\Bus; it may

be useful to see verbose output on both the client side and from the daemon.)

.PP

If you want to get fancy, you can create a custom bus configuration for your test bus (see the session.conf and system.conf files that define the two default configurations for example). This would allow you to specify a different directory for .service files, for example.

.SH AUTHOR

See <http://www.freedesktop.org/software/dbus/doc/AUTHORS>

.SH BUGS

Please send bug reports to the D\Bus mailing list or bug tracker, see <http://www.freedesktop.org/software/dbus/>

File = dbus-daemon.1.in

.\

.\ dbus\daemon manual page.

.\ Copyright (C) 2003,2008 Red Hat, Inc.

.\

.TH dbus\daemon 1

.SH NAME

dbus\daemon \- Message bus daemon

.SH SYNOPSIS

.PP

.B dbus\daemon

dbus\daemon [\-\-version] [\-\-session] [\-\-system] [\-\-config\-

file=FILE]

[\-\-print\-address[=DESCRIPTOR]] [\-\-print\-pid[=DESCRIPTOR]] [\-\-

fork]

.SH DESCRIPTION

\fIdbus\daemon\fP is the D\Bus message bus daemon. See <http://www.freedesktop.org/software/dbus/> for more information about the big picture. D\Bus is first a library that provides one-to-one communication between any two applications; \fIdbus\daemon\fP is an application that uses this library to implement a message bus daemon. Multiple programs connect to the message bus daemon and can exchange messages with one another.

.PP

There are two standard message bus instances: the systemwide message bus

(installed on many systems as the "messagebus" init service) and the per\user\login\session message bus (started each time a user logs in).

\fIdbus\daemon\fp is used for both of these instances, but with a different configuration file.

.PP

The \-\-session option is equivalent to

"\-\-config\file=@EXPANDED_SYSCONFDIR@/dbus\-1/session.conf" and the \-\-system

option is equivalent to

"\-\-config\file=@EXPANDED_SYSCONFDIR@/dbus\-1/system.conf". By creating

additional configuration files and using the \-\-config\file option, additional special\purpose message bus daemons could be created.

.PP

The systemwide daemon is normally launched by an init script, standardly called simply "messagebus".

.PP

The systemwide daemon is largely used for broadcasting system events, such as changes to the printer queue, or adding/removing devices.

.PP

The per\session daemon is used for various interprocess communication among desktop applications (however, it is not tied to X or the GUI in any way).

.PP

SIGHUP will cause the D\Bus daemon to PARTIALLY reload its configuration file and to flush its user/group information caches.

Some

configuration changes would require kicking all apps off the bus; so they will

only take effect if you restart the daemon. Policy changes should take effect

with SIGHUP.

.SH OPTIONS

The following options are supported:

.TP

.I "\-\-config\file=FILE"

Use the given configuration file.

.TP

.I "\-\-fork"

Force the message bus to fork and become a daemon, even if the configuration file does not specify that it should.

In most contexts the configuration file already gets this right, though.

.I "\-\-nofork"

Force the message bus not to fork and become a daemon, even if the configuration file specifies that it should.

.TP

.I "\-\-print\address[=DESCRIPTOR]"

Print the address of the message bus to standard output, or to the given file descriptor. This is used by programs that launch the message bus.

.TP

.I "\-\\-print\\-pid[=DESCRIPTOR]"

Print the process ID of the message bus to standard output, or to the given file descriptor. This is used by programs that launch the message bus.

.TP

.I "\-\\-session"

Use the standard configuration file for the per\\-login\\-session message bus.

.TP

.I "\-\\-system"

Use the standard configuration file for the systemwide message bus.

.TP

.I "\-\\-version"

Print the version of the daemon.

.TP

.I "\-\\-introspect"

Print the introspection information for all D\\-Bus internal interfaces.

.TP

.I "\-\\-address[=ADDRESS]"

Set the address to listen on. This option overrides the address configured in the configuration file.

.TP

.I "\-\\-systemd\\-activation"

Enable systemd\\-style service activation. Only useful in conjunction with the systemd system and session manager on Linux.

.TP

.I "\-\\-nopidfile"

Don't write a PID file even if one is configured in the configuration files.

.SH CONFIGURATION FILE

A message bus daemon has a configuration file that specializes it for a particular application. For example, one configuration file might set up the message bus to be a systemwide message bus, while another might set it up to be a per\\-user\\-login\\-session bus.

.PP

The configuration file also establishes resource limits, security parameters, and so forth.

.PP

The configuration file is not part of any interoperability specification and its backward compatibility is not guaranteed; this document is documentation, not specification.

.PP

The standard systemwide and per\\-session message bus setups are

configured in the files "@EXPANDED_SYSCONFDIR@/dbus\-1/system.conf" and "@EXPANDED_SYSCONFDIR@/dbus\-1/session.conf". These files normally <include> a system\local.conf or session\local.conf; you can put local overrides in those files to avoid modifying the primary configuration files.

.PP

The configuration file is an XML document. It must have the following doctype declaration:

.nf

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D\Bus Bus
Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
```

.fi

.PP

The following elements may be present in the configuration file.

.TP

.I "<busconfig>"

.PP

Root element.

.TP

.I "<type>"

.PP

The well-known type of the message bus. Currently known values are "system" and "session"; if other values are set, they should be either added to the D\Bus specification, or namespaced. The last <type> element "wins" (previous values are ignored). This element only controls which message bus specific environment variables are set in activated clients. Most of the policy that distinguishes a session bus from the system bus is controlled from the other elements in the configuration file.

.PP

If the well-known type of the message bus is "session", then the DBUS_STARTER_BUS_TYPE environment variable will be set to "session" and the DBUS_SESSION_BUS_ADDRESS environment variable will be set to the address of the session bus. Likewise, if the type of the message bus is "system", then the DBUS_STARTER_BUS_TYPE environment variable will be set to "system" and the DBUS_SESSION_BUS_ADDRESS environment variable will be set to the address of the system bus (which is normally well known anyway).

.PP

Example: <type>session</type>

.TP
.I "<include>"

.PP
Include a file <include>filename.conf</include> at this point. If the filename is relative, it is located relative to the configuration file doing the including.

.PP
<include> has an optional attribute "ignore_missing=(yes|no)" which defaults to "no" if not provided. This attribute controls whether it's a fatal error for the included file to be absent.

.TP
.I "<includedir>"

.PP
Include all files in <includedir>foo.d</includedir> at this point. Files in the directory are included in undefined order. Only files ending in ".conf" are included.

.PP
This is intended to allow extension of the system bus by particular packages. For example, if CUPS wants to be able to send out notification of printer queue changes, it could install a file to @EXPANDED_SYSCONFDIR@/dbus\-1/system.d that allowed all apps to receive this message and allowed the printer daemon user to send it.

.TP
.I "<user>"

.PP
The user account the daemon should run as, as either a username or a UID. If the daemon cannot change to this UID on startup, it will exit. If this element is not present, the daemon will not change or care about its UID.

.PP
The last <user> entry in the file "wins", the others are ignored.

.PP
The user is changed after the bus has completed initialization. So sockets etc. will be created before changing user, but no data will be read from clients before changing user. This means that sockets and PID files can be created in a location that requires root privileges for writing.

.TP

.I "<fork>"

.PP

If present, the bus daemon becomes a real daemon (forks into the background, etc.). This is generally used rather than the `--fork` command line option.

.TP

.I "<keep_umask>"

.PP

If present, the bus daemon keeps its original umask when forking. This may be useful to avoid affecting the behavior of child processes.

.TP

.I "<listen>"

.PP

Add an address that the bus should listen on. The address is in the standard D\ -Bus format that contains a transport name plus possible parameters/options.

.PP

Example: `<listen>unix:path=/tmp/foo</listen>`

.PP

Example: `<listen>tcp:host=localhost,port=1234</listen>`

.PP

If there are multiple `<listen>` elements, then the bus listens on multiple addresses. The bus will pass its address to started services or other interested parties with the last address given in `<listen>` first. That is, apps will try to connect to the last `<listen>` address first.

.PP

tcp sockets can accept IPv4 addresses, IPv6 addresses or hostnames. If a hostname resolves to multiple addresses, the server will bind to all of them. The `family=ipv4` or `family=ipv6` options can be used to force it to bind to a subset of addresses

.PP

Example: `<listen>tcp:host=localhost,port=0,family=ipv4</listen>`

.PP

A special case is using a port number of zero (or omitting the port), which means to choose an available port selected by the operating system. The port number chosen can be obtained with the `--print-address` command line parameter and will be present in other cases where the server reports its own address, such as when `DBUS_SESSION_BUS_ADDRESS` is set.

.PP

Example: <listen>tcp:host=localhost,port=0</listen>

.PP

tcp addresses also allow a bind=hostname option, which will override the host option specifying what address to bind to, without changing the address reported by the bus. The bind option can also take a special name '*' to cause the bus to listen on all local address (INADDR_ANY). The specified host should be a valid name of the local machine or weird stuff will happen.

.PP

Example: <listen>tcp:host=localhost,bind=*,port=0</listen>

.TP

.I "<auth>"

.PP

Lists permitted authorization mechanisms. If this element doesn't exist, then all known mechanisms are allowed. If there are multiple <auth> elements, all the listed mechanisms are allowed. The order in which mechanisms are listed is not meaningful.

.PP

Example: <auth>EXTERNAL</auth>

.PP

Example: <auth>DBUS_COOKIE_SHA1</auth>

.TP

.I "<servicedir>"

.PP

Adds a directory to scan for .service files. Directories are scanned starting with the last to appear in the config file (the first .service file found that provides a particular service will be used).

.PP

Service files tell the bus how to automatically start a program. They are primarily used with the per\ -user\ -session bus, not the systemwide bus.

.TP

.I "<standard_session_servicedirs/>"

.PP

<standard_session_servicedirs/> is equivalent to specifying a series of <servicedir/> elements for each of the data directories in the "XDG Base Directory Specification" with the subdirectory "dbus\ -1/services", so for example "/usr/share/dbus\ -1/services" would be among the

directories searched.

.PP

The "XDG Base Directory Specification" can be found at <http://freedesktop.org/wiki/Standards/basedir\spec> if it hasn't moved, otherwise try your favorite search engine.

.PP

The `<standard_session_servicedirs/>` option is only relevant to the `per\user\session` bus daemon defined in `@EXPANDED_SYSCONFDIR@/dbus\1/session.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "<standard_system_servicedirs/>"

.PP

`<standard_system_servicedirs/>` specifies the standard system-wide activation directories that should be searched for service files. This option defaults to `@EXPANDED_DATADIR@/dbus\1/system\services`.

.PP

The `<standard_system_servicedirs/>` option is only relevant to the `per\system` bus daemon defined in `@EXPANDED_SYSCONFDIR@/dbus\1/system.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "<servicehelper/>"

.PP

`<servicehelper/>` specifies the setuid helper that is used to launch system daemons with an alternate user. Typically this should be the `dbus\daemon\launch\helper` executable in located in `libexec`.

.PP

The `<servicehelper/>` option is only relevant to the `per\system` bus daemon defined in `@EXPANDED_SYSCONFDIR@/dbus\1/system.conf`. Putting it in any other configuration file would probably be nonsense.

.TP

.I "<limit>"

.PP

`<limit>` establishes a resource limit. For example:

.nf

 <limit name="max_message_size">64</limit>

 <limit name="max_completed_connections">512</limit>

.fi

.PP

The name attribute is mandatory.
Available limit names are:

```
.nf
    "max_incoming_bytes"      : total size in bytes of messages
                              incoming from a single connection
    "max_incoming_unix_fds"  : total number of unix fds of
messages
                              incoming from a single connection
    "max_outgoing_bytes"     : total size in bytes of messages
                              queued up for a single connection
    "max_outgoing_unix_fds"  : total number of unix fds of
messages
                              queued up for a single connection
    "max_message_size"       : max size of a single message in
                              bytes
    "max_message_unix_fds"   : max unix fds of a single message
    "service_start_timeout"   : milliseconds (thousandths) until
                              a started service has to connect
    "auth_timeout"           : milliseconds (thousandths) a
                              connection is given to
                              authenticate
    "max_completed_connections" : max number of authenticated
connections
    "max_incomplete_connections" : max number of unauthenticated
                              connections
    "max_connections_per_user" : max number of completed
connections from
                              the same user
    "max_pending_service_starts" : max number of service launches in
                              progress at the same time
    "max_names_per_connection" : max number of names a single
                              connection can own
    "max_match_rules_per_connection": max number of match rules for
a single
                              connection
    "max_replies_per_connection" : max number of pending method
                              replies per connection
                              (number of calls\ -in\ -progress)
    "reply_timeout"           : milliseconds (thousandths)
                              until a method call times out
.fi
```

.PP

The max incoming/outgoing queue sizes allow a new message to be queued if one byte remains below the max. So you can in fact exceed the max by max_message_size.

.PP

max_completed_connections divided by max_connections_per_user is the

number of users that can work together to denial\of\service all other users by using up all connections on the systemwide bus.

.PP

Limits are normally only of interest on the systemwide bus, not the user session buses.

.TP

.I "<policy>"

.PP

The <policy> element defines a security policy to be applied to a particular set of connections to the bus. A policy is made up of <allow> and <deny> elements. Policies are normally used with the systemwide bus; they are analogous to a firewall in that they allow expected traffic and prevent unexpected traffic.

.PP

Currently, the system bus has a default\deny policy for sending method calls and owning bus names. Everything else, in particular reply messages, receive checks, and signals has a default allow policy.

.PP

In general, it is best to keep system services as small, targeted programs which run in their own process and provide a single bus name. Then, all that is needed is an <allow> rule for the "own" permission to let the process claim the bus name, and a "send_destination" rule to allow traffic from some or all uids to your service.

.PP

The <policy> element has one of four attributes:

.nf

```
context="(default|mandatory)"
at_console="(true|false)"
user="username or userid"
group="group name or gid"
```

.fi

.PP

Policies are applied to a connection as follows:

.nf

```
\- all context="default" policies are applied
```

```
\- all group="connection's user's group" policies are applied
   in undefined order
\- all user="connection's auth user" policies are applied
   in undefined order
\- all at_console="true" policies are applied
\- all at_console="false" policies are applied
\- all context="mandatory" policies are applied
.fi
```

.PP

Policies applied later will override those applied earlier, when the policies overlap. Multiple policies with the same user/group/context are applied in the order they appear in the config file.

.TP

```
.I "<deny>"
.I "<allow>"
```

.PP

A <deny> element appears below a <policy> element and prohibits some action. The <allow> element makes an exception to previous <deny> statements, and works just like <deny> but with the inverse meaning.

.PP

The possible attributes of these elements are:
.nf

```
send_interface="interface_name"
send_member="method_or_signal_name"
send_error="error_name"
send_destination="name"
send_type="method_call" | "method_return" | "signal" | "error"
send_path="/path/name"

receive_interface="interface_name"
receive_member="method_or_signal_name"
receive_error="error_name"
receive_sender="name"
receive_type="method_call" | "method_return" | "signal" | "error"
receive_path="/path/name"

send_requested_reply="true" | "false"
receive_requested_reply="true" | "false"

eavesdrop="true" | "false"

own="name"
own_prefix="name"
user="username"
group="groupname"
```

.fi

.PP

Examples:

.nf

```
<deny send_destination="org.freedesktop.Service"
send_interface="org.freedesktop.System" send_member="Reboot"/>
<deny send_destination="org.freedesktop.System"/>
<deny receive_sender="org.freedesktop.System"/>
<deny user="john"/>
<deny group="enemies"/>
```

.fi

.PP

The `<deny>` element's attributes determine whether the deny "matches" a particular action. If it matches, the action is denied (unless later rules in the config file allow it).

.PP

`send_destination` and `receive_sender` rules mean that messages may not be sent to or received from the *owner* of the given name, not that they may not be sent *to that name*. That is, if a connection owns services A, B, C, and sending to A is denied, sending to B or C will not work either.

.PP

The other `send_*` and `receive_*` attributes are purely textual/`by\`-value matches against the given field in the message header.

.PP

"Eavesdropping" occurs when an application receives a message that was explicitly addressed to a name the application does not own, or is a reply to such a message. Eavesdropping thus only applies to messages that are addressed to services and replies to such messages (i.e. it does not apply to signals).

.PP

For `<allow>`, `eavesdrop="true"` indicates that the rule matches even when eavesdropping. `eavesdrop="false"` is the default and means that the rule only allows messages to go to their specified recipient. For `<deny>`, `eavesdrop="true"` indicates that the rule matches only when eavesdropping. `eavesdrop="false"` is the default for `<deny>` also, but here it means that the rule applies always, even when not eavesdropping. The `eavesdrop` attribute can only be combined with `send` and `receive` rules (with `send_*` and `receive_*` attributes).

.PP

The `[send|receive]_requested_reply` attribute works similarly to the `eavesdrop` attribute. It controls whether the `<deny>` or `<allow>` matches a reply that is expected (corresponds to a previous method call message). This attribute only makes sense for reply messages (errors and method returns), and is ignored for other message types.

.PP

For `<allow>`, `[send|receive]_requested_reply="true"` is the default and indicates that only requested replies are allowed by the

rule. [send|receive]_requested_reply="false" means that the rule allows any reply even if unexpected.

.PP

For <deny>, [send|receive]_requested_reply="false" is the default but indicates that the rule matches only when the reply was not requested. [send|receive]_requested_reply="true" indicates that the rule applies always, regardless of pending reply state.

.PP

user and group denials mean that the given user or group may not connect to the message bus.

.PP

For "name", "username", "groupname", etc. the character "*" can be substituted, meaning "any." Complex globs like "foo.bar.*" aren't allowed for now because they'd be work to implement and maybe encourage sloppy security anyway.

.PP

<allow own_prefix="a.b"/> allows you to own the name "a.b" or any name whose first dot-separated elements are "a.b": in particular, you can own "a.b.c" or "a.b.c.d", but not "a.bc" or "a.c". This is useful when services like Telepathy and ReserveDevice define a meaning for subtrees of well-known names, such as org.freedesktop.Telepathy.ConnectionManager.(anything) and org.freedesktop.ReserveDevice1.(anything).

.PP

It does not make sense to deny a user or group inside a <policy> for a user or group; user/group denials can only be inside context="default" or context="mandatory" policies.

.PP

A single <deny> rule may specify combinations of attributes such as send_destination and send_interface and send_type. In this case, the denial applies only if both attributes match the message being denied. e.g. <deny send_interface="foo.bar" send_destination="foo.blah"/> would deny messages with the given interface AND the given bus name. To get an OR effect you specify multiple <deny> rules.

.PP

You can't include both send_ and receive_ attributes on the same rule, since "whether the message can be sent" and "whether it can be received" are evaluated separately.

.PP

Be careful with send_interface/receive_interface, because the

interface field in messages is optional. In particular, do NOT specify `<deny send_interface="org.foo.Bar"/>`! This will cause no\-interface messages to be blocked for all services, which is almost certainly not what you intended. Always use rules of the form: `<deny send_interface="org.foo.Bar" send_destination="org.foo.Service"/>`

.TP
.I "<selinux>"

.PP
The `<selinux>` element contains settings related to Security Enhanced Linux.
More details below.

.TP
.I "<associate>"

.PP
An `<associate>` element appears below an `<selinux>` element and creates a mapping. Right now only one kind of association is possible:
.nf
 `<associate own="org.freedesktop.Foobar" context="foo_t"/>`
.fi

.PP
This means that if a connection asks to own the name "org.freedesktop.Foobar" then the source context will be the context of the connection and the target context will be "foo_t" \- see the short discussion of SELinux below.

.PP
Note, the context here is the target context when requesting a name, NOT the context of the connection owning the name.

.PP
There's currently no way to set a default for owning any name, if we add this syntax it will look like:
.nf
 `<associate own="*" context="foo_t"/>`
.fi
If you find a reason this is useful, let the developers know. Right now the default will be the security context of the bus itself.

.PP
If two `<associate>` elements specify the same name, the element appearing later in the configuration file will be used.

.SH SELinux

.PP

See <http://www.nsa.gov/selinux/> for full details on SELinux. Some useful excerpts:

.IP "" 8

Every subject (process) and object (e.g. file, socket, IPC object, etc) in the system is assigned a collection of security attributes, known as a security context. A security context contains all of the security attributes associated with a particular subject or object that are relevant to the security policy.

.IP "" 8

In order to better encapsulate security contexts and to provide greater efficiency, the policy enforcement code of SELinux typically handles security identifiers (SIDs) rather than security contexts. A SID is an integer that is mapped by the security server to a security context at runtime.

.IP "" 8

When a security decision is required, the policy enforcement code passes a pair of SIDs (typically the SID of a subject and the SID of an object, but sometimes a pair of subject SIDs or a pair of object SIDs), and an object security class to the security server. The object security class indicates the kind of object, e.g. a process, a regular file, a directory, a TCP socket, etc.

.IP "" 8

Access decisions specify whether or not a permission is granted for a given pair of SIDs and class. Each object class has a set of associated permissions defined to control operations on objects with that class.

.PP

D\ -Bus performs SELinux security checks in two places.

.PP

First, any time a message is routed from one connection to another connection, the bus daemon will check permissions with the security context of the first connection as source, security context of the second connection as target, object class "dbus" and requested permission "send_msg".

.PP

If a security context is not available for a connection (impossible when using UNIX domain sockets), then the target context used is the context of the bus daemon itself. There is currently no way to change this default, because we're assuming that only UNIX domain sockets will be used to connect to the systemwide bus. If this changes, we'll probably add a way to set the default connection context.

.PP

Second, any time a connection asks to own a name, the bus daemon will check permissions with the security context of the connection as source, the security context specified for the name in the config file as target, object class "dbus" and requested permission "acquire_svc".

.PP

The security context for a bus name is specified with the <associate> element described earlier in this document. If a name has no security context associated in the configuration file, the security context of the bus daemon itself will be used.

.SH DEBUGGING

.PP

If you're trying to figure out where your messages are going or why you aren't getting messages, there are several things you can try.

.PP

Remember that the system bus is heavily locked down and if you haven't installed a security policy file to allow your message through, it won't work. For the session bus, this is not a concern.

.PP

The simplest way to figure out what's happening on the bus is to run the `\fIdbus\monitor` program, which comes with the `D-Bus` package. You can also send test messages with `\fIdbus\send`. These programs have their own man pages.

.PP

If you want to know what the daemon itself is doing, you might consider running a separate copy of the daemon to test against. This will allow you to put the daemon under a debugger, or run it with verbose output, without messing up your real session and system daemons.

.PP

To run a separate test copy of the daemon, for example you might open a terminal and type:

.nf

```
DBUS_VERBOSE=1 dbus\daemon \-\-session \-\-print\-\-address
```

.fi

.PP

The test daemon address will be printed when the daemon starts. You will need to copy\and\paste this address and use it as the value of the `DBUS_SESSION_BUS_ADDRESS` environment variable when you launch the applications you want to test. This will cause those applications to connect to your test bus instead of the `DBUS_SESSION_BUS_ADDRESS` of your real session bus.

.PP

DBUS_VERBOSE=1 will have NO EFFECT unless your copy of D\Bus was compiled with verbose mode enabled. This is not recommended in production builds due to performance impact. You may need to rebuild D\Bus if your copy was not built with debugging in mind.

(DBUS_VERBOSE

also affects the D\Bus library and thus applications using D\Bus; it may

be useful to see verbose output on both the client side and from the daemon.)

.PP

If you want to get fancy, you can create a custom bus configuration for your test bus (see the session.conf and system.conf files that define the two default configurations for example). This would allow you to specify a different directory for .service files, for example.

.SH AUTHOR

See <http://www.freedesktop.org/software/dbus/doc/AUTHORS>

.SH BUGS

Please send bug reports to the D\Bus mailing list or bug tracker, see <http://www.freedesktop.org/software/dbus/>

File = dbus-daemon.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://www.oasis-
open.org/docbook/xml/4.1.2/docbookx.dtd">
<!-- lifted from troff+man by doclifter -->
<refentry id='dbus-daemon'>
<!-- -->
<!-- dbus\daemon manual page. -->
<!-- Copyright (C) 2003 Red Hat, Inc. -->

<refmeta>
<refentrytitle>dbus-daemon</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>
<refnamediv id='name'>
<refname>dbus-daemon</refname>
<refpurpose>Message bus daemon</refpurpose>
</refnamediv>
<!-- body begins here -->
<refsynopsisdiv id='synopsis'>
<cmdsynopsis>
  <command>dbus-daemon</command></cmdsynopsis>
<cmdsynopsis>
  <command>dbus-daemon</command>    <arg choice='opt'>--version </arg>
```

```
<arg choice='opt'>--session </arg>
<arg choice='opt'>--system </arg>
<arg choice='opt'>--config-
file=<replaceable>FILE</replaceable></arg>
<arg choice='opt'><arg choice='plain'>--print-address </arg><arg
choice='opt'><replaceable>=DESCRIPTOR</replaceable></arg></arg>
<arg choice='opt'><arg choice='plain'>--print-pid </arg><arg
choice='opt'><replaceable>=DESCRIPTOR</replaceable></arg></arg>
<arg choice='opt'>--fork </arg>
<sbr/>
</cmdsynopsis>
</refsynopsisdiv>
```

<refsect1 id='description'><title>DESCRIPTION</title>
<para><command>dbus-daemon</command> is the D-Bus message bus daemon.
See
<ulink
url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop
.org/software/dbus/</ulink> for more information about
the big picture. D-Bus is first a library that provides one-to-one
communication between any two applications; <command>dbus-
daemon</command> is an
application that uses this library to implement a message bus
daemon. Multiple programs connect to the message bus daemon and can
exchange messages with one another.</para>

<para>There are two standard message bus instances: the systemwide
message bus
(installed on many systems as the "messagebus" init service) and the
per-user-login-session message bus (started each time a user logs in).
<command>dbus-daemon</command> is used for both of these instances,
but with
a different configuration file.</para>

<para>The --session option is equivalent to
"--config-file=/etc/dbus-1/session.conf" and the --system
option is equivalent to
"--config-file=/etc/dbus-1/system.conf". By creating
additional configuration files and using the --config-file option,
additional special-purpose message bus daemons could be
created.</para>

<para>The systemwide daemon is normally launched by an init script,
standardly called simply "messagebus".</para>

<para>The systemwide daemon is largely used for broadcasting system
events,

such as changes to the printer queue, or adding/removing devices.</para>

<para>The per-session daemon is used for various interprocess communication among desktop applications (however, it is not tied to X or the GUI in any way).</para>

<para>SIGHUP will cause the D-Bus daemon to PARTIALLY reload its configuration file and to flush its user/group information caches. Some configuration changes would require kicking all apps off the bus; so they will only take effect if you restart the daemon. Policy changes should take effect with SIGHUP.</para>

</refsect1>

<refsect1 id='options'><title>OPTIONS</title>

<para>The following options are supported:</para>

<variablelist remap='TP'>

<varlistentry>

<term><option>--config-file=FILE</option></term>

<listitem>

<para>Use the given configuration file.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--fork</option></term>

<listitem>

<para>Force the message bus to fork and become a daemon, even if the configuration file does not specify that it should.

In most contexts the configuration file already gets this right, though.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--print-address[=DESCRIPTOR]</option></term>

<listitem>

<para>Print the address of the message bus to standard output, or to the given file descriptor. This is used by programs that launch the message bus.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--print-pid[=DESCRIPTOR]</option></term>

<listitem>

<para>Print the process ID of the message bus to standard output, or to the given file descriptor. This is used by programs that

launch the message bus.</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--session</option></term>
<listitem>
<para>Use the standard configuration file for the per-login-session
message
bus.</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--system</option></term>
<listitem>
<para>Use the standard configuration file for the systemwide message
bus.</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--version</option></term>
<listitem>
<para>Print the version of the daemon.</para>

</listitem>
</varlistentry>
</variablelist>
</refsect1>

<refsect1 id='configuration_file'><title>CONFIGURATION FILE</title>
<para>A message bus daemon has a configuration file that specializes
it
for a particular application. For example, one configuration
file might set up the message bus to be a systemwide message bus,
while another might set it up to be a per-user-login-session
bus.</para>

<para>The configuration file also establishes resource limits,
security
parameters, and so forth.</para>

<para>The configuration file is not part of any interoperability
specification and its backward compatibility is not guaranteed; this
document is documentation, not specification.</para>

<para>The standard systemwide and per-session message bus setups are
configured in the files "/etc/dbus-1/system.conf" and
"/etc/dbus-1/session.conf". These files normally
<include> a system-local.conf or session-local.conf; you can put
local

overrides in those files to avoid modifying the primary configuration files.</para>

<para>The configuration file is an XML document. It must have the following

doctype declaration:</para>

```
<literallayout remap='.nf'>
```

```
    &lt;!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-Bus Bus Configuration 1.0//EN"
```

```
    "<ulink
```

```
url='http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd'>http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd</ulink>"&gt;
```

```
</literallayout> <!-- .fi -->
```

<para>The following elements may be present in the configuration file.</para>

```
<variablelist remap='TP'>
```

```
  <varlistentry>
```

```
    <term><emphasis remap='I'>&lt;busconfig&gt;</emphasis></term>
```

```
    <listitem>
```

```
<para></para>
```

```
  </listitem>
```

```
  </varlistentry>
```

```
</variablelist>
```

<para>Root element.</para>

```
<variablelist remap='TP'>
```

```
  <varlistentry>
```

```
    <term><emphasis remap='I'>&lt;type&gt;</emphasis></term>
```

```
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
```

```
  </listitem>
```

```
  </varlistentry>
```

```
</variablelist>
```

<para>The well-known type of the message bus. Currently known values are

"system" and "session"; if other values are set, they should be either added to the D-Bus specification, or namespaced. The last <type> element "wins" (previous values are ignored).</para>

<para>Example: <type>session</type></para>

```
<variablelist remap='TP'>
```

```
<varlistentry>
  <term><emphasis remap='I'>&lt;include&gt;</emphasis></term>
  <listitem>
<para></para>
  </listitem>
</varlistentry>
</variablelist>
```

<para>Include a file <include>filename.conf</include> at this point. If the filename is relative, it is located relative to the configuration file doing the including.</para>

<para><include> has an optional attribute "ignore_missing=(yes|no)" which defaults to "no" if not provided. This attribute controls whether it's a fatal error for the included file to be absent.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'>&lt;includedir&gt;</emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

<para>Include all files in <includedir>foo.d</includedir> at this point. Files in the directory are included in undefined order. Only files ending in ".conf" are included.</para>

<para>This is intended to allow extension of the system bus by particular packages. For example, if CUPS wants to be able to send out notification of printer queue changes, it could install a file to /etc/dbus-1/system.d that allowed all apps to receive this message and allowed the printer daemon user to send it.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'>&lt;user&gt;</emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

<para>The user account the daemon should run as, as either a username or a UID. If the daemon cannot change to this UID on startup, it will exit. If this element is not present, the daemon will not change or care about its UID.</para>

<para>The last <code>user</code> entry in the file "wins", the others are ignored.</para>

<para>The user is changed after the bus has completed initialization. So sockets etc. will be created before changing user, but no data will be read from clients before changing user. This means that sockets and PID files can be created in a location that requires root privileges for writing.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emph remap='I'><code>fork</code></emph></term>
    <listitem>
<para></para>
    </listitem>
  </varlistentry>
</variablelist>
```

<para>If present, the bus daemon becomes a real daemon (forks into the background, etc.). This is generally used rather than the --fork command line option.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emph remap='I'><code>listen</code></emph></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

<para>Add an address that the bus should listen on. The address is in the standard D-Bus format that contains a transport name plus possible parameters/options.</para>

<para>Example: <code>listen:unix:path=/tmp/foo</code></para>

<para>If there are multiple <code>listen</code> elements, then the bus listens

on multiple addresses. The bus will pass its address to started services or other interested parties with the last address given in `<listen>`; first. That is, apps will try to connect to the last `<listen>`; address first.

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'><auth></emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

Lists permitted authorization mechanisms. If this element doesn't exist, then all known mechanisms are allowed. If there are multiple `<auth>` elements, all the listed mechanisms are allowed. The order in which mechanisms are listed is not meaningful.

Example: `<auth>EXTERNAL</auth>`

Example: `<auth>DBUS_COOKIE_SHA1</auth>`

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'><servicedir></emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

Adds a directory to scan for `.service` files. Directories are scanned starting with the last to appear in the config file (the first `.service` file found that provides a particular service will be used).

Service files tell the bus how to automatically start a program. They are primarily used with the per-user-session bus, not the systemwide bus.

```
<variablelist remap='TP'>
  <varlistentry>
```

```
<term><emphasis
remap='I'>&lt;standard_session_servicedirs/&gt;</emphasis></term>
<listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
</listitem>
</varlistentry>
</variablelist>
```

```
<para>&lt;standard_session_servicedirs/&gt; is equivalent to
specifying a series
of &lt;servicedir/&gt; elements for each of the data directories in
the "XDG
Base Directory Specification" with the subdirectory "dbus-1/services",
so for example "/usr/share/dbus-1/services" would be among the
directories searched.</para>
```

```
<para>The "XDG Base Directory Specification" can be found at
<ulink url='http://freedesktop.org/wiki/Standards/basedir-
spec'>http://freedesktop.org/wiki/Standards/basedir-spec</ulink> if it
hasn't moved,
otherwise try your favorite search engine.</para>
```

```
<para>The &lt;standard_session_servicedirs/&gt; option is only
relevant to the
per-user-session bus daemon defined in
/etc/dbus-1/session.conf. Putting it in any other
configuration file would probably be nonsense.</para>
```

```
<variablelist remap='TP'>
<varlistentry>
<term><emphasis remap='I'>&lt;limit&gt;</emphasis></term>
<listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
</listitem>
</varlistentry>
</variablelist>
```

```
<para>&lt;limit&gt; establishes a resource limit. For example:</para>
<litterallayout remap='.nf'>
&lt;limit name="max_message_size"&gt;64&lt;/limit&gt;
&lt;limit name="max_completed_connections"&gt;512&lt;/limit&gt;
</litterallayout> <!-- .fi -->
```

```
<para>The name attribute is mandatory.
Available limit names are:</para>
```

```
<litterallayout remap='.nf'>
"max_incoming_bytes" : total size in bytes of messages
```

```

        "max_outgoing_bytes"      : total size in bytes of messages
        "max_message_size"       : max size of a single message in
        "service_start_timeout"  : milliseconds (thousandths) until
        "auth_timeout"          : milliseconds (thousandths) a
        "max_completed_connections" : max number of authenticated
connections
        "max_incomplete_connections" : max number of unauthenticated
connections
        "max_connections_per_user" : max number of completed
connections from
        "max_pending_service_starts" : max number of service launches in
        "max_names_per_connection"  : max number of names a single
        "max_match_rules_per_connection": max number of match rules for
a single
        "max_replies_per_connection" : max number of pending method
        "reply_timeout"           : milliseconds (thousandths)
incoming from a single connection
        until a method call times out
</literallayout> <!-- .fi -->

```

<para>The max incoming/outgoing queue sizes allow a new message to be queued if one byte remains below the max. So you can in fact exceed the max by max_message_size.</para>

<para>max_completed_connections divided by max_connections_per_user is the number of users that can work together to denial-of-service all other users by using up all connections on the systemwide bus.</para>

<para>Limits are normally only of interest on the systemwide bus, not the user session buses.</para>

```

<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'>&lt;policy&gt;</emphasis></term>

```

```
<listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
  </varlistentry>
</variablelist>
```

```
<para>The <code><policy></code> element defines a security policy to be
applied to a particular
set of connections to the bus. A policy is made up of
<code><allow></code> and <code><deny></code> elements. Policies are normally used
with the systemwide bus;
they are analogous to a firewall in that they allow expected traffic
and prevent unexpected traffic.</para>
```

```
<para>The <code><policy></code> element has one of three attributes:</para>
<literallayout remap='.nf'>
  context="(default|mandatory)"
  user="username or userid"
  group="group name or gid"
</literallayout> <!-- .fi -->
```

```
<para>
Policies are applied to a connection as follows:</para>
<literallayout remap='.nf'>
  - all context="default" policies are applied
  - all group="connection's user's group" policies are applied
    in undefined order
  - all user="connection's auth user" policies are applied
    in undefined order
  - all context="mandatory" policies are applied
</literallayout> <!-- .fi -->
```

```
<para>Policies applied later will override those applied earlier,
when the policies overlap. Multiple policies with the same
user/group/context are applied in the order they appear
in the config file.</para>
```

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'><code><deny></code></emphasis></term>
    <listitem>
<para><emphasis remap='I'><code><allow></code></emphasis></para>

  </listitem>
</varlistentry>
</variablelist>
```


<para>A <deny> element appears below a <policy> element and prohibits some action. The <allow> element makes an exception to previous <deny> statements, and works just like <deny> but with the inverse meaning.</para>

<para>The possible attributes of these elements are:</para>

```
<literallayout remap='.nf'>
  send_interface="interface_name"
  send_member="method_or_signal_name"
  send_error="error_name"
  send_destination="name"
  send_type="method_call" | "method_return" | "signal" | "error"
  send_path="/path/name"

  receive_interface="interface_name"
  receive_member="method_or_signal_name"
  receive_error="error_name"
  receive_sender="name"
  receive_type="method_call" | "method_return" | "signal" | "error"
  receive_path="/path/name"

  send_requested_reply="true" | "false"
  receive_requested_reply="true" | "false"

  eavesdrop="true" | "false"

  own="name"
  own_prefix="name"
  user="username"
  group="groupname"
</literallayout> <!-- .fi -->
```

<para>Examples:</para>

```
<literallayout remap='.nf'>
  <deny send_interface="org.freedesktop.System"
send_member="Reboot"/>
  <deny receive_interface="org.freedesktop.System"
receive_member="Reboot"/>
  <deny own="org.freedesktop.System"/>
  <deny send_destination="org.freedesktop.System"/>
  <deny receive_sender="org.freedesktop.System"/>
  <deny user="john"/>
  <deny group="enemies"/>
</literallayout> <!-- .fi -->
```

<para>The <deny> element's attributes determine whether the deny "matches" a

particular action. If it matches, the action is denied (unless later rules in the config file allow it).</para>

<para>send_destination and receive_sender rules mean that messages may not be sent to or received from the *owner* of the given name, not that they may not be sent *to that name*. That is, if a connection owns services A, B, C, and sending to A is denied, sending to B or C will not work either.</para>

<para>The other send_* and receive_* attributes are purely textual/by-value matches against the given field in the message header.</para>

<para>"Eavesdropping" occurs when an application receives a message that was explicitly addressed to a name the application does not own. Eavesdropping thus only applies to messages that are addressed to services (i.e. it does not apply to signals).</para>

<para>For <code><allow></code>, eavesdrop="true" indicates that the rule matches even when eavesdropping. eavesdrop="false" is the default and means that the rule only allows messages to go to their specified recipient. For <code><deny></code>, eavesdrop="true" indicates that the rule matches only when eavesdropping. eavesdrop="false" is the default for <code><deny></code> also, but here it means that the rule applies always, even when not eavesdropping. The eavesdrop attribute can only be combined with receive rules (with receive_* attributes).</para>

<para>The [send|receive]_requested_reply attribute works similarly to the eavesdrop attribute. It controls whether the <code><deny></code> or <code><allow></code> matches a reply that is expected (corresponds to a previous method call message). This attribute only makes sense for reply messages (errors and method returns), and is ignored for other message types.</para>

<para>For <code><allow></code>, [send|receive]_requested_reply="true" is the default and indicates that only requested replies are allowed by the rule. [send|receive]_requested_reply="false" means that the rule allows any reply even if unexpected.</para>

<para>For <code><deny></code>, [send|receive]_requested_reply="false" is the default but indicates that the rule matches only when the reply was not requested. [send|receive]_requested_reply="true" indicates that the rule applies always, regardless of pending reply state.</para>

<para>user and group denials mean that the given user or group may not connect to the message bus.</para>

<para>For "name", "username", "groupname", etc. the character "*" can be substituted, meaning "any." Complex globs like "foo.bar.*" aren't allowed for now because they'd be work to implement and maybe encourage sloppy security anyway.</para>

<para><code><allow own_prefix="a.b"/></code> allows you to own the name "a.b" or any name whose first dot-separated elements are "a.b": in particular, you can own "a.b.c" or "a.b.c.d", but not "a.bc" or "a.c". This is useful when services like Telepathy and ReserveDevice define a meaning for subtrees of well-known names, such as org.freedesktop.Telepathy.ConnectionManager.(anything) and org.freedesktop.ReserveDevice1.(anything).</para>

<para>It does not make sense to deny a user or group inside a <code><policy></code> for a user or group; user/group denials can only be inside context="default" or context="mandatory" policies.</para>

<para>A single <code><deny></code> rule may specify combinations of attributes such as send_destination and send_interface and send_type. In this case, the denial applies only if both attributes match the message being denied. e.g. <code><deny send_interface="foo.bar" send_destination="foo.blah"/></code> would deny messages with the given interface AND the given bus name. To get an OR effect you specify multiple <code><deny></code> rules.</para>

<para>You can't include both send_ and receive_ attributes on the same rule, since "whether the message can be sent" and "whether it can be received" are evaluated separately.</para>

<para>Be careful with send_interface/receive_interface, because the interface field in messages is optional.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'>&lt;selinux&gt;</emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

<para>The <selinux> element contains settings related to Security Enhanced Linux. More details below.</para>

```
<variablelist remap='TP'>
  <varlistentry>
    <term><emphasis remap='I'>&lt;associate&gt;</emphasis></term>
    <listitem>
```

```
<para></para> <!-- FIXME: blank list item -->
  </listitem>
</varlistentry>
</variablelist>
```

<para>An <associate> element appears below an <selinux> element and creates a mapping. Right now only one kind of association is possible:</para>

```
<literallayout remap='.nf'>
  &lt;associate own="org.freedesktop.Foobar" context="foo_t"/&gt;
</literallayout> <!-- .fi -->
```

<para>This means that if a connection asks to own the name "org.freedesktop.Foobar" then the source context will be the context of the connection and the target context will be "foo_t" - see the short discussion of SELinux below.</para>

<para>Note, the context here is the target context when requesting a name, NOT the context of the connection owning the name.</para>

<para>There's currently no way to set a default for owning any name, if

we add this syntax it will look like:</para>

```
<literallayout remap='.nf'>
  &lt;associate own="*" context="foo_t"/&gt;
</literallayout> <!-- .fi -->
```

<para>If you find a reason this is useful, let the developers know.

Right now the default will be the security context of the bus itself.</para>

<para>If two <code><lt;associate></code> elements specify the same name, the element appearing later in the configuration file will be used.</para>

</refsect1>

<refsect1 id='selinux'><title>SELinux</title>

<para>See <ulink url='http://www.nsa.gov/selinux/'>http://www.nsa.gov/selinux/</ulink> for full details on SELinux. Some useful excerpts:</para>

<para>Every subject (process) and object (e.g. file, socket, IPC object, etc) in the system is assigned a collection of security attributes, known as a security context. A security context contains all of the security attributes associated with a particular subject or object that are relevant to the security policy.</para>

<para>In order to better encapsulate security contexts and to provide greater efficiency, the policy enforcement code of SELinux typically handles security identifiers (SIDs) rather than security contexts. A SID is an integer that is mapped by the security server to a security context at runtime.</para>

<para>When a security decision is required, the policy enforcement code passes a pair of SIDs (typically the SID of a subject and the SID of an object, but sometimes a pair of subject SIDs or a pair of object SIDs), and an object security class to the security server. The object security class indicates the kind of object, e.g. a process, a regular file, a directory, a TCP socket, etc.</para>

<para>Access decisions specify whether or not a permission is granted for a given pair of SIDs and class. Each object class has a set of associated permissions defined to control operations on objects with that class.</para>

<para>D-Bus performs SELinux security checks in two places.</para>

<para>First, any time a message is routed from one connection to another

connection, the bus daemon will check permissions with the security context of the first connection as source, security context of the second connection as target, object class "dbus" and requested permission "send_msg".</para>

<para>If a security context is not available for a connection (impossible when using UNIX domain sockets), then the target context used is the context of the bus daemon itself. There is currently no way to change this default, because we're assuming that only UNIX domain sockets will be used to connect to the systemwide bus. If this changes, we'll probably add a way to set the default connection context.</para>

<para>Second, any time a connection asks to own a name, the bus daemon will check permissions with the security context of the connection as source, the security context specified for the name in the config file as target, object class "dbus" and requested permission "acquire_svc".</para>

<para>The security context for a bus name is specified with the <associate> element described earlier in this document. If a name has no security context associated in the configuration file, the security context of the bus daemon itself will be used.</para>

</refsect1>

<refsect1 id='author'><title>AUTHOR</title>

<para>See <ulink url='http://www.freedesktop.org/software/dbus/doc/AUTHORS'>http://www.freedesktop.org/software/dbus/doc/AUTHORS</ulink></para>

</refsect1>

<refsect1 id='bugs'><title>BUGS</title>

<para>Please send bug reports to the D-Bus mailing list or bug tracker, see <ulink url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop.org/software/dbus/</ulink></para>

</refsect1>

</refentry>

File = dbus-dataslot.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-dataslot.c  storing data on objects
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
*
*/

#include <config.h>
#include "dbus-dataslot.h"
#include "dbus-threads-internal.h"

/**
* @defgroup DBusDataSlot Data slots
* @ingroup DBusInternals
* @brief Storing data by ID
*
* Types and functions related to storing data by an
* allocated ID. This is used for dbus_connection_set_data(),
* dbus_server_set_data(), etc.
* @{
*/

/**
* Initializes a data slot allocator object, used to assign
* integer IDs for data slots.
*
* @param allocator the allocator to initialize
*/
dbus_bool_t
_dbus_data_slot_allocator_init (DBusDataSlotAllocator *allocator)
{
    allocator->allocated_slots = NULL;

```

```

    allocator->n_allocated_slots = 0;
    allocator->n_used_slots = 0;
    allocator->lock_loc = NULL;

    return TRUE;
}

/**
 * Allocates an integer ID to be used for storing data
 * in a #DBusDataSlotList. If the value at *slot_id_p is
 * not -1, this function just increments the refcount for
 * the existing slot ID. If the value is -1, a new slot ID
 * is allocated and stored at *slot_id_p.
 *
 * @param allocator the allocator
 * @param mutex_loc the location lock for this allocator
 * @param slot_id_p address to fill with the slot ID
 * @returns #TRUE on success
 */
dbus_bool_t
_dbus_data_slot_allocator_alloc (DBusDataSlotAllocator *allocator,
                                DBusRMutex             **mutex_loc,
                                dbus_int32_t            *slot_id_p)
{
    dbus_int32_t slot;

    _dbus_rmutex_lock (*mutex_loc);

    if (allocator->n_allocated_slots == 0)
    {
        _dbus_assert (allocator->lock_loc == NULL);
        allocator->lock_loc = mutex_loc;
    }
    else if (allocator->lock_loc != mutex_loc)
    {
        _dbus_warn_check_failed ("D-Bus threads were initialized after
first using the D-Bus library. If your application does not directly
initialize threads or use D-Bus, keep in mind that some library or
plugin may have used D-Bus or initialized threads behind your back.
You can often fix this problem by calling dbus_init_threads() or
dbus_g_threads_init() early in your main() method, before D-Bus is
used.\n");
        _dbus_assert_not_reached ("exiting");
    }

    if (*slot_id_p >= 0)
    {
        slot = *slot_id_p;

        _dbus_assert (slot < allocator->n_allocated_slots);
        _dbus_assert (allocator->allocated_slots[slot].slot_id == slot);
    }
}

```



```

    allocator->allocated_slots[slot].refcount += 1;

    goto out;
}

_dbus_assert (*slot_id_p < 0);

if (allocator->n_used_slots < allocator->n_allocated_slots)
{
    slot = 0;
    while (slot < allocator->n_allocated_slots)
    {
        if (allocator->allocated_slots[slot].slot_id < 0)
        {
            allocator->allocated_slots[slot].slot_id = slot;
            allocator->allocated_slots[slot].refcount = 1;
            allocator->n_used_slots += 1;
            break;
        }
        ++slot;
    }

    _dbus_assert (slot < allocator->n_allocated_slots);
}
else
{
    DBusAllocatedSlot *tmp;

    slot = -1;
    tmp = dbus_realloc (allocator->allocated_slots,
                       sizeof (DBusAllocatedSlot) * (allocator-
>n_allocated_slots + 1));
    if (tmp == NULL)
        goto out;

    allocator->allocated_slots = tmp;
    slot = allocator->n_allocated_slots;
    allocator->n_allocated_slots += 1;
    allocator->n_used_slots += 1;
    allocator->allocated_slots[slot].slot_id = slot;
    allocator->allocated_slots[slot].refcount = 1;
}

_dbus_assert (slot >= 0);
_dbus_assert (slot < allocator->n_allocated_slots);
_dbus_assert (*slot_id_p < 0);
_dbus_assert (allocator->allocated_slots[slot].slot_id == slot);
_dbus_assert (allocator->allocated_slots[slot].refcount == 1);

*slot_id_p = slot;

```

```

    _dbus_verbose ("Allocated slot %d on allocator %p total %d slots
allocated %d used\n",
                  slot, allocator, allocator->n_allocated_slots,
allocator->n_used_slots);

out:
    _dbus_rmutex_unlock (*(allocator->lock_loc));
    return slot >= 0;
}

/**
 * Deallocates an ID previously allocated with
 * _dbus_data_slot_allocator_alloc(). Existing data stored on
 * existing #DBusDataSlotList objects with this ID will be freed when
the
 * data list is finalized, but may not be retrieved (and may only be
 * replaced if someone else reallocates the slot).
 * The slot value is reset to -1 if this is the last unref.
 *
 * @param allocator the allocator
 * @param slot_id_p address where we store the slot
 */
void
_dbus_data_slot_allocator_free (DBusDataSlotAllocator *allocator,
                               dbus_int32_t          *slot_id_p)
{
    _dbus_rmutex_lock (*(allocator->lock_loc));

    _dbus_assert (*slot_id_p < allocator->n_allocated_slots);
    _dbus_assert (allocator->allocated_slots[*slot_id_p].slot_id ==
*slot_id_p);
    _dbus_assert (allocator->allocated_slots[*slot_id_p].refcount > 0);

    allocator->allocated_slots[*slot_id_p].refcount -= 1;

    if (allocator->allocated_slots[*slot_id_p].refcount > 0)
    {
        _dbus_rmutex_unlock (*(allocator->lock_loc));
        return;
    }

    /* refcount is 0, free the slot */
    _dbus_verbose ("Freeing slot %d on allocator %p total %d allocated
%d used\n",
                  *slot_id_p, allocator, allocator->n_allocated_slots,
allocator->n_used_slots);

    allocator->allocated_slots[*slot_id_p].slot_id = -1;
    *slot_id_p = -1;

    allocator->n_used_slots -= 1;

```



```

                                DBusFreeFunction    *old_free_func,
                                void                **old_data)
{
#ifdef DBUS_DISABLE_ASSERT
    /* We need to take the allocator lock here, because the allocator
    could
    * be e.g. realloc()ing allocated_slots. We avoid doing this if
    asserts
    * are disabled, since then the asserts are empty.
    */
    _dbus_rmutex_lock (*(allocator->lock_loc));
    _dbus_assert (slot < allocator->n_allocated_slots);
    _dbus_assert (allocator->allocated_slots[slot].slot_id == slot);
    _dbus_rmutex_unlock (*(allocator->lock_loc));
#endif

    if (slot >= list->n_slots)
    {
        DBusDataSlot *tmp;
        int i;

        tmp = dbus_realloc (list->slots,
                           sizeof (DBusDataSlot) * (slot + 1));
        if (tmp == NULL)
            return FALSE;

        list->slots = tmp;
        i = list->n_slots;
        list->n_slots = slot + 1;
        while (i < list->n_slots)
        {
            list->slots[i].data = NULL;
            list->slots[i].free_data_func = NULL;
            ++i;
        }
    }

    _dbus_assert (slot < list->n_slots);

    *old_data = list->slots[slot].data;
    *old_free_func = list->slots[slot].free_data_func;

    list->slots[slot].data = data;
    list->slots[slot].free_data_func = free_data_func;

    return TRUE;
}

/**
 * Retrieves data previously set with _dbus_data_slot_list_set_data().
 * The slot must still be allocated (must not have been freed).
 */

```

```

* @param allocator the allocator slot was allocated from
* @param list the data slot list
* @param slot the slot to get data from
* @returns the data, or #NULL if not found
*/
void*
_dbus_data_slot_list_get (DBusDataSlotAllocator *allocator,
                          DBusDataSlotList      *list,
                          int                     slot)
{
#ifdef DBUS_DISABLE_ASSERT
/* We need to take the allocator lock here, because the allocator
could
* be e.g. realloc()ing allocated_slots. We avoid doing this if
asserts
* are disabled, since then the asserts are empty.
*/
_dbus_rmutex_lock (*(allocator->lock_loc));
_dbus_assert (slot >= 0);
_dbus_assert (slot < allocator->n_allocated_slots);
_dbus_assert (allocator->allocated_slots[slot].slot_id == slot);
_dbus_rmutex_unlock (*(allocator->lock_loc));
#endif

if (slot >= list->n_slots)
return NULL;
else
return list->slots[slot].data;
}

/**
* Frees all data slots contained in the list, calling
* application-provided free functions if they exist.
*
* @param list the list to clear
*/
void
_dbus_data_slot_list_clear (DBusDataSlotList *list)
{
int i;

i = 0;
while (i < list->n_slots)
{
if (list->slots[i].free_data_func)
(* list->slots[i].free_data_func) (list->slots[i].data);
list->slots[i].data = NULL;
list->slots[i].free_data_func = NULL;
++i;
}
}

```

```

/**
 * Frees the data slot list and all data slots contained
 * in it, calling application-provided free functions
 * if they exist.
 *
 * @param list the list to free
 */
void
_dbus_data_slot_list_free (DBusDataSlotList *list)
{
    _dbus_data_slot_list_clear (list);

    dbus_free (list->slots);
    list->slots = NULL;
    list->n_slots = 0;
}

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

static int free_counter;

static void
test_free_slot_data_func (void *data)
{
    int i = _DBUS_POINTER_TO_INT (data);

    _dbus_assert (free_counter == i);
    ++free_counter;
}

/**
 * Test function for data slots
 */
dbus_bool_t
_dbus_data_slot_test (void)
{
    DBusDataSlotAllocator allocator;
    DBusDataSlotList list;
    int i;
    DBusFreeFunction old_free_func;
    void *old_data;
    DBusRMutex *mutex;

    if (!_dbus_data_slot_allocator_init (&allocator))
        _dbus_assert_not_reached ("no memory for allocator");

    _dbus_data_slot_list_init (&list);

```



```

    _dbus_assert_not_reached ("no memory to set data");

    _dbus_assert (old_free_func == test_free_slot_data_func);
    _dbus_assert (_DBUS_POINTER_TO_INT (old_data) == i);

    (* old_free_func) (old_data);
    _dbus_assert (i == (free_counter - 1));

    _dbus_assert (_dbus_data_slot_list_get (&allocator, &list, i) ==
                  _DBUS_INT_TO_POINTER (i));

    ++i;
}

free_counter = 0;
_dbus_data_slot_list_free (&list);

_dbus_assert (N_SLOTS == free_counter);

i = 0;
while (i < N_SLOTS)
{
    dbus_int32_t tmp = i;

    _dbus_data_slot_allocator_free (&allocator, &tmp);
    _dbus_assert (tmp == -1);
    ++i;
}

_dbus_rmutex_free_at_location (&mutex);

return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-dataslot.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-dataslot.h  storing data on objects
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or

```



```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_DATASLOT_H
#define DBUS_DATASLOT_H

#include <dbus/dbus-internals.h>

DBUS_BEGIN_DECLS

typedef struct DBusDataSlotAllocator DBusDataSlotAllocator;
typedef struct DBusDataSlotList DBusDataSlotList;

/** Opaque typedef for DBusDataSlot */
typedef struct DBusDataSlot DBusDataSlot;
/** DBusDataSlot is used to store application data on the connection
*/
struct DBusDataSlot
{
    void *data;                /**< The application data */
    DBusFreeFunction free_data_func; /**< Free the application data */
};

typedef struct DBusAllocatedSlot DBusAllocatedSlot;

/** An allocated slot for storing data
*/
struct DBusAllocatedSlot
{
    dbus_int32_t slot_id; /**< ID of this slot */
    int          refcount; /**< Number of uses of the slot */
};

/**
* An allocator that tracks a set of slot IDs.
*/
struct DBusDataSlotAllocator
{
    DBusAllocatedSlot *allocated_slots; /**< Allocated slots */
    int n_allocated_slots; /**< number of slots malloc'd */
    int n_used_slots;      /**< number of slots used */
    DBusRMutex **lock_loc; /**< location of thread lock */
};

```

```

};

/**
 * Data structure that stores the actual user data set at a given
 * slot.
 */
struct DBusDataSlotList
{
    DBusDataSlot *slots;    /**< Data slots */
    int          n_slots;   /**< Slots we have storage for in data_slots
*/
};

dbus_bool_t _dbus_data_slot_allocator_init (DBusDataSlotAllocator
*allocator);
dbus_bool_t _dbus_data_slot_allocator_alloc (DBusDataSlotAllocator
*allocator,
                                             DBusRMutex
                                             int
**mutex_loc,
*slot_id_p);
void _dbus_data_slot_allocator_free (DBusDataSlotAllocator
*allocator,
                                     int
*slot_id_p);
void _dbus_data_slot_list_init (DBusDataSlotList
*list);
dbus_bool_t _dbus_data_slot_list_set (DBusDataSlotAllocator
*allocator,
                                     DBusDataSlotList
*list,
                                     int
slot,
*data,
DBusFreeFunction
free_data_func,
DBusFreeFunction
*old_free_func,
void
**old_data);
void* _dbus_data_slot_list_get (DBusDataSlotAllocator
*allocator,
                                DBusDataSlotList
*list,
                                int
slot);
void _dbus_data_slot_list_clear (DBusDataSlotList
*list);
void _dbus_data_slot_list_free (DBusDataSlotList
*list);

```

```

DBUS_END_DECLS

#endif /* DBUS_DATASLOT_H */

File = dbus-env.bat.cmake

:: environment setting for dbus clients
@echo off

:: session bus address
set DBUS_SESSION_BUS_ADDRESS=@DBUS_SESSION_BUS_DEFAULT_ADDRESS@

:: system bus address
set DBUS_SYSTEM_BUS_DEFAULT_ADDRESS=@DBUS_SYSTEM_BUS_DEFAULT_ADDRESS@

File = dbus-errors.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-errors.c Error reporting
 *
 * Copyright (C) 2002, 2004 Red Hat Inc.
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

#include <config.h>
#include "dbus-errors.h"
#include "dbus-internals.h"
#include "dbus-string.h"

```

```

#include "dbus-protocol.h"
#include <stdarg.h>
#include <string.h>

/**
 * @defgroup DBusErrorInternals Error reporting internals
 * @ingroup DBusInternals
 * @brief Error reporting internals
 * @{
 */

/**
 * @def DBUS_ERROR_INIT
 *
 * Expands to a suitable initializer for a DBusError on the stack.
 * Declaring a DBusError with:
 *
 * @code
 * DBusError error = DBUS_ERROR_INIT;
 *
 * do_things_with (&error);
 * @endcode
 *
 * is a more concise form of:
 *
 * @code
 * DBusError error;
 *
 * dbus_error_init (&error);
 * do_things_with (&error);
 * @endcode
 */

/**
 * Internals of DBusError
 */
typedef struct
{
    char *name; /**< error name */
    char *message; /**< error message */

    unsigned int const_message : 1; /**< Message is not owned by
DBusError */

    unsigned int dummy2 : 1; /**< placeholder */
    unsigned int dummy3 : 1; /**< placeholder */
    unsigned int dummy4 : 1; /**< placeholder */
    unsigned int dummy5 : 1; /**< placeholder */

    void *padding1; /**< placeholder */
} DBusRealError;

```

```

_DBUS_STATIC_ASSERT (sizeof (DBusRealError) == sizeof (DBusError));

/**
 * Returns a longer message describing an error name.
 * If the error name is unknown, returns the name
 * itself.
 *
 * @param error the error to describe
 * @returns a constant string describing the error.
 */
static const char*
message_from_error (const char *error)
{
    if (strcmp (error, DBUS_ERROR_FAILED) == 0)
        return "Unknown error";
    else if (strcmp (error, DBUS_ERROR_NO_MEMORY) == 0)
        return "Not enough memory available";
    else if (strcmp (error, DBUS_ERROR_IO_ERROR) == 0)
        return "Error reading or writing data";
    else if (strcmp (error, DBUS_ERROR_BAD_ADDRESS) == 0)
        return "Could not parse address";
    else if (strcmp (error, DBUS_ERROR_NOT_SUPPORTED) == 0)
        return "Feature not supported";
    else if (strcmp (error, DBUS_ERROR_LIMITS_EXCEEDED) == 0)
        return "Resource limits exceeded";
    else if (strcmp (error, DBUS_ERROR_ACCESS_DENIED) == 0)
        return "Permission denied";
    else if (strcmp (error, DBUS_ERROR_AUTH_FAILED) == 0)
        return "Could not authenticate to server";
    else if (strcmp (error, DBUS_ERROR_NO_SERVER) == 0)
        return "No server available at address";
    else if (strcmp (error, DBUS_ERROR_TIMEOUT) == 0)
        return "Connection timed out";
    else if (strcmp (error, DBUS_ERROR_NO_NETWORK) == 0)
        return "Network unavailable";
    else if (strcmp (error, DBUS_ERROR_ADDRESS_IN_USE) == 0)
        return "Address already in use";
    else if (strcmp (error, DBUS_ERROR_DISCONNECTED) == 0)
        return "Disconnected.";
    else if (strcmp (error, DBUS_ERROR_INVALID_ARGS) == 0)
        return "Invalid arguments.";
    else if (strcmp (error, DBUS_ERROR_NO_REPLY) == 0)
        return "Did not get a reply message.";
    else if (strcmp (error, DBUS_ERROR_FILE_NOT_FOUND) == 0)
        return "File doesn't exist.";
    else if (strcmp (error, DBUS_ERROR_OBJECT_PATH_IN_USE) == 0)
        return "Object path already in use";
    else
        return error;
}

```

```

/** @} */ /* End of internals */

/**
 * @defgroup DBusErrors Error reporting
 * @ingroup DBus
 * @brief Error reporting
 *
 * Types and functions related to reporting errors.
 *
 * In essence D-Bus error reporting works as follows:
 *
 * @code
 * DBusError error;
 * dbus_error_init (&error);
 * dbus_some_function (arg1, arg2, &error);
 * if (dbus_error_is_set (&error))
 * {
 *     fprintf (stderr, "an error occurred: %s\n", error.message);
 *     dbus_error_free (&error);
 * }
 * @endcode
 *
 * By convention, all functions allow #NULL instead of a DBusError*,
 * so callers who don't care about the error can ignore it.
 *
 * There are some rules. An error passed to a D-Bus function must
 * always be unset; you can't pass in an error that's already set. If
 * a function has a return code indicating whether an error occurred,
 * and also a #DBusError parameter, then the error will always be set
 * if and only if the return code indicates an error occurred. i.e.
 * the return code and the error are never going to disagree.
 *
 * An error only needs to be freed if it's been set, not if
 * it's merely been initialized.
 *
 * You can check the specific error that occurred using
 * dbus_error_has_name().
 *
 * Errors will not be set for programming errors, such as passing
 * invalid arguments to the libdbus API. Instead, libdbus will print
 * warnings, exit on a failed assertion, or even crash in those cases
 * (in other words, incorrect use of the API results in undefined
 * behavior, possibly accompanied by helpful debugging output if
 * you're lucky).
 *
 * @{
 */

/**
 * Initializes a DBusError structure. Does not allocate any memory;
 * the error only needs to be freed if it is set at some point.

```

```

*
* @param error the DBusError.
*/
void
dbus_error_init (DBusError *error)
{
    DBusRealError *real;

    _dbus_return_if_fail (error != NULL);

    _dbus_assert (sizeof (DBusError) == sizeof (DBusRealError));

    real = (DBusRealError *)error;

    real->name = NULL;
    real->message = NULL;

    real->const_message = TRUE;
}

/**
 * Frees an error that's been set (or just initialized),
 * then reinitializes the error as in dbus_error_init().
 *
 * @param error memory where the error is stored.
 */
void
dbus_error_free (DBusError *error)
{
    DBusRealError *real;

    _dbus_return_if_fail (error != NULL);

    real = (DBusRealError *)error;

    if (!real->const_message)
    {
        dbus_free (real->name);
        dbus_free (real->message);
    }

    dbus_error_init (error);
}

/**
 * Assigns an error name and message to a DBusError. Does nothing if
 * error is #NULL. The message may be #NULL, which means a default
 * message will be deduced from the name. The default message will be
 * totally useless, though, so using a #NULL message is not
 * recommended.
 *
 * Because this function does not copy the error name or message, you

```

```

* must ensure the name and message are global data that won't be
* freed. You probably want dbus_set_error() instead, in most cases.
*
* @param error the error.or #NULL
* @param name the error name (not copied!!!)
* @param message the error message (not copied!!!)
*/
void
dbus_set_error_const (DBusError *error,
                    const char *name,
                    const char *message)
{
    DBusRealError *real;

    _dbus_return_if_error_is_set (error);
    _dbus_return_if_fail (name != NULL);

    if (error == NULL)
        return;

    _dbus_assert (error->name == NULL);
    _dbus_assert (error->message == NULL);

    if (message == NULL)
        message = message_from_error (name);

    real = (DBusRealError *)error;

    real->name = (char*) name;
    real->message = (char *)message;
    real->const_message = TRUE;
}

/**
* Moves an error src into dest, freeing src and
* overwriting dest. Both src and dest must be initialized.
* src is reinitialized to an empty error. dest may not
* contain an existing error. If the destination is
* #NULL, just frees and reinit's the source error.
*
* @param src the source error
* @param dest the destination error or #NULL
*/
void
dbus_move_error (DBusError *src,
                DBusError *dest)
{
    _dbus_return_if_error_is_set (dest);

    if (dest)
    {
        dbus_error_free (dest);
    }
}

```



```

        *dest = *src;
        dbus_error_init (src);
    }
    else
        dbus_error_free (src);
}

/**
 * Checks whether the error is set and has the given
 * name.
 * @param error the error
 * @param name the name
 * @returns #TRUE if the given named error occurred
 */
dbus_bool_t
dbus_error_has_name (const DBusError *error,
                    const char      *name)
{
    _dbus_return_val_if_fail (error != NULL, FALSE);
    _dbus_return_val_if_fail (name != NULL, FALSE);

    _dbus_assert ((error->name != NULL && error->message != NULL) ||
                  (error->name == NULL && error->message == NULL));

    if (error->name != NULL)
    {
        DBusString str1, str2;
        _dbus_string_init_const (&str1, error->name);
        _dbus_string_init_const (&str2, name);
        return _dbus_string_equal (&str1, &str2);
    }
    else
        return FALSE;
}

/**
 * Checks whether an error occurred (the error is set).
 *
 * @param error the error object
 * @returns #TRUE if an error occurred
 */
dbus_bool_t
dbus_error_is_set (const DBusError *error)
{
    _dbus_return_val_if_fail (error != NULL, FALSE);
    _dbus_assert ((error->name != NULL && error->message != NULL) ||
                  (error->name == NULL && error->message == NULL));
    return error->name != NULL;
}

/**
 * Assigns an error name and message to a DBusError.

```

```

* Does nothing if error is #NULL.
*
* The format may be #NULL, which means a (pretty much useless)
* default message will be deduced from the name. This is not a good
* idea, just go ahead and provide a useful error message. It won't
* hurt you.
*
* If no memory can be allocated for the error message,
* an out-of-memory error message will be set instead.
*
* @param error the error.or #NULL
* @param name the error name
* @param format printf-style format string.
*/
void
dbus_set_error (DBusError *error,
               const char *name,
               const char *format,
               ...)
{
    DBusRealError *real;
    DBusString str;
    va_list args;

    if (error == NULL)
        return;

    /* it's a bug to pile up errors */
    _dbus_return_if_error_is_set (error);
    _dbus_return_if_fail (name != NULL);

    _dbus_assert (error->name == NULL);
    _dbus_assert (error->message == NULL);

    if (!_dbus_string_init (&str))
        goto nomem;

    if (format == NULL)
    {
        if (!_dbus_string_append (&str,
                                message_from_error (name)))
        {
            _dbus_string_free (&str);
            va_end (args);
            goto nomem;
        }
    }
    else
    {
        va_start (args, format);
        if (!_dbus_string_append_printf_valist (&str, format, args))
        {

```

```

        _dbus_string_free (&str);
        va_end (args);
        goto nomem;
    }
    va_end (args);
}

real = (DBusRealError *)error;

if (!_dbus_string_steal_data (&str, &real->message))
{
    _dbus_string_free (&str);
    goto nomem;
}
_dbus_string_free (&str);

real->name = _dbus_strdup (name);
if (real->name == NULL)
{
    dbus_free (real->message);
    real->message = NULL;
    goto nomem;
}
real->const_message = FALSE;

return;

nomem:
    _DBUS_SET_OOM (error);
}

/** @} */ /* End public API */

```

File = dbus-errors.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-errors.h Error reporting
 *
 * Copyright (C) 2002 Red Hat Inc.
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 */

```

```

* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_ERROR_H
#define DBUS_ERROR_H

#include <dbus/dbus-macros.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-protocol.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusErrors
 * @{
 */

/** Mostly-opaque type representing an error that occurred */
typedef struct DBusError DBusError;

/**
 * Object representing an exception.
 */
struct DBusError
{
    const char *name;    /**< public error name field */
    const char *message; /**< public error message field */

    unsigned int dummy1 : 1; /**< placeholder */
    unsigned int dummy2 : 1; /**< placeholder */
    unsigned int dummy3 : 1; /**< placeholder */
    unsigned int dummy4 : 1; /**< placeholder */
    unsigned int dummy5 : 1; /**< placeholder */

    void *padding1; /**< placeholder */
};

#define DBUS_ERROR_INIT { NULL, NULL, TRUE, 0, 0, 0, 0, NULL }

```

```

DBUS_EXPORT
void      dbus_error_init      (DBusError      *error);
DBUS_EXPORT
void      dbus_error_free      (DBusError      *error);
DBUS_EXPORT
void      dbus_set_error       (DBusError      *error,
                               const char      *name,
                               const char      *message,
                               ...);

DBUS_EXPORT
void      dbus_set_error_const (DBusError      *error,
                               const char      *name,
                               const char      *message);

DBUS_EXPORT
void      dbus_move_error      (DBusError      *src,
                               DBusError      *dest);

DBUS_EXPORT
dbus_bool_t dbus_error_has_name (const DBusError *error,
                               const char      *name);

DBUS_EXPORT
dbus_bool_t dbus_error_is_set   (const DBusError *error);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_ERROR_H */

```

File = dbus-faq.html

```

<html><head><meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1"><title>D-Bus FAQ</title><meta name="generator"
content="DocBook XSL Stylesheets V1.76.1"></head><body bgcolor="white"
text="black" link="#0000FF" vlink="#840084" alink="#0000FF"><div
class="article" title="D-Bus FAQ"><div class="titlepage"><div><div><h2
class="title"><a name="index"></a>D-Bus FAQ</h2></div><div><div
class="authorgroup"><div class="author"><h3 class="author"><span
class="firstname">Havoc</span> <span
class="surname">Pennington</span></h3><div class="affiliation"><span
class="orgname">Red Hat, Inc.<br></span><div class="address"><p><br>
<code class="email">&lt;<a class="email"
href="mailto:hp@pobox.com">hp@pobox.com</a>&gt;</code><br>
</p></div></div></div><div class="author"><h3
class="author"><span class="firstname">David</span> <span
class="othername">A</span> <span
class="surname">Wheeler</span></h3></div></div></div><div><p
class="releaseinfo">Version 0.3</p></div></div><hr></div><div
class="qandaset" title="Frequently Asked Questions"><a
name="faq"></a><dl><dt>1. <a href="#idp48080">
What is D-Bus?

```

in

```
</a></dt><dt>2. <a href="#idp51984">
  Is D-Bus stable/finished?
</a></dt><dt>3. <a href="#idp55152">
  How is the reference implementation licensed? Can I use it
  proprietary applications?
</a></dt><dt>4. <a href="#idp59104">
  What is the difference between a bus name, and object path,
  and an interface?
</a></dt><dt>5. <a href="#service">
  What is a "service"?
</a></dt><dt>6. <a href="#components">
  Is D-Bus a "component system"?
</a></dt><dt>7. <a href="#speed">
  How fast is the D-Bus reference implementation?
</a></dt><dt>8. <a href="#size">
  How large is the D-Bus reference implementation?
</a></dt><dt>9. <a href="#other-ipc">
  How does D-Bus differ from other interprocess communication
  or networking protocols?
</a></dt><dt>10. <a href="#corba">
  How does D-Bus differ from CORBA?
</a></dt><dt>11. <a href="#xmlrpcsoap">
  How does D-Bus differ from XML-RPC and SOAP?
</a></dt><dt>12. <a href="#dce">
  How does D-Bus differ from DCE?
</a></dt><dt>13. <a href="#dcom">
  How does D-Bus differ from DCOM and COM?
</a></dt><dt>14. <a href="#internet-communications-engine">
  How does D-Bus differ from ZeroC's Internet Communications
  Engine (Ice)
</a></dt><dt>15. <a href="#inter-client-exchange">
  How does D-Bus differ from Inter-Client Exchange (ICE)?
</a></dt><dt>16. <a href="#dcop">
  How does D-Bus differ from DCOP?
</a></dt><dt>17. <a href="#yet-more-ipc">
  How does D-Bus differ from [yet more IPC mechanisms]?
</a></dt><dt>18. <a href="#which-ipc">
  Which IPC mechanism should I use?
</a></dt><dt>19. <a href="#idp4836432">
  How can I submit a bug or patch?
</a></dt></dl><table border="0" width="100%" summary="Q and A
Set"><col align="left" width="1%"><col><tbody><tr class="question"
title="1."><td align="left" valign="top"><a name="idp48080"></a><a
name="idp48336"></a><p><b>1.</b></p></td><td align="left"
valign="top"><p>
  What is D-Bus?
  </p></td></tr><tr class="answer"><td align="left"
valign="top"></td><td align="left" valign="top"><p>
  This is probably best answered by reading the D-Bus <a
class="ulink" href="dbus-tutorial.html" target="_top">tutorial</a> or
```

the introduction to the [specification](dbus-specification.html). In short, it is a system consisting of 1) a wire protocol for exposing a typical object-oriented language/framework to other applications; and 2) a bus daemon that allows applications to find and monitor one another.

Phrased differently, D-Bus is 1) an interprocess communication (IPC) system and 2) some higher-level structure (lifecycle tracking, service activation, security policy) provided by two bus daemons, one systemwide and one per-user-session.

2.

Is D-Bus stable/finished?

The low-level library "libdbus" and the protocol specification are considered

ABI stable. The <README>

file has a discussion of the API/ABI stability guarantees. Higher-level bindings (such as those for Qt, GLib, Python, Java, C#) each have their own release schedules and degree of maturity, not linked to the low-level library and bus daemon release. Check the project page for the binding you're considering to understand that project's policies.

3.

How is the reference implementation licensed? Can I use it in proprietary applications?

The short answer is yes, you can use it in proprietary applications.

You should read the <COPYING>

file, which offers you the choice of two licenses. These are the GPL and the

AFL. The GPL requires that your application be licensed under the GPL

as well. The AFL is an "X-style" or "BSD-style" license compatible

with proprietary licensing, but it does have some requirements; in particular it prohibits you from filing a lawsuit alleging that the D-Bus software infringes your patents while you continue to use D-Bus. If you're going to sue, you have to stop using the software. Read the licenses to determine their meaning, this FAQ entry is not intended to change the meaning or terms of the licenses.

</p></td></tr><tr class="question" title="4."><td align="left" valign="top"><p>4.</p></td><td align="left" valign="top"><p>

What is the difference between a bus name, and object path, and an interface?

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

If you imagine a C++ program that implements a network service, then

the bus name is the hostname of the computer running this C++ program,

the object path is a C++ object instance pointer, and an interface is

a C++ class (a pure virtual or abstract class, to be exact).

</p><p>

In Java terms, the object path is an object reference, and an interface is a Java interface.

</p><p>

People get confused because if they write an application with a single object instance and a single interface, then the bus name, object path, and interface look redundant. For example, you might have a text editor that uses the bus name <code

class="literal">org.freedesktop.TextEditor</code>,</p><p>

has a global singleton object called

<code class="literal">/org/freedesktop/TextEditor</code>,</p><p>

and

that singleton object could implement the interface

<code class="literal">org.freedesktop.TextEditor</code>.</p><p>

</p><p>

However, a text editor application could as easily own multiple bus

names (for example, <code

class="literal">org.kde.KWrite</code> in addition to

generic <code class="literal">TextEditor</code>), have

multiple objects (maybe

<code class="literal">/org/kde/documents/4352</code> where

the number changes

multiple according to the document), and each object could implement multiple interfaces, such as `org.freedesktop.DBus.Introspectable`, `org.freedesktop.BasicTextField`, `org.kde.RichTextDocument`.

5. [service](#)

What is a "service"?

A service is a program that can be launched by the bus daemon to provide some functionality to other programs. Services are normally launched according to the bus name they will have.

People often misuse the word "service" for any bus name, but this tends to be ambiguous and confusing so is discouraged.

In the D-Bus docs we try to use "service" only when talking about programs the bus knows how to launch, i.e. a service always has a .service file.

6. [components](#)

Is D-Bus a "component system"?

D-Bus is not a component system. "Component system" was originally defined by COM, and was essentially a workaround for the limitations of the C++ object system (adding introspection, runtime location of objects, ABI guarantees, and so forth). With the C# language and CLR,

Microsoft added these features to the primary object system, leaving COM obsolete. Similarly, Java has much less need for something like COM than C++ did. Even QObject (from Qt) and GObject (from GLib) offer some of the same features found in COM.

Component systems are not about GUI control embedding. Embedding a spreadsheet in a word processor document is a matter of defining some specific *interfaces* that objects can implement. These interfaces provide methods related to GUI controls. So an object implementing those interfaces can be embedded.

The word "component" just means "object with some fancy features" and in modern languages all objects are effectively "components."

So components are fancy objects, and some objects are GUI controls.

A third, unrelated feature is interprocess communication or IPC. D-Bus is an IPC system. Given an object (or "component" if you must), you can expose the functionality of that object over an IPC system. Examples of IPC systems are DCOM, CORBA, SOAP, XML-RPC, and D-Bus. You can use any of these IPC systems with any object/component system, though some of them are "tuned" for specific object systems. You can think of an IPC system primarily as a wire protocol.

If you combine an IPC system with a set of GUI control interfaces, then you can have an out-of-process or dynamically-loaded GUI control.

Another related concept is the *plugin* or *extension*. Generic plugin systems such as the [Eclipse](http://eclipse.org) system are not so different from component/object systems, though perhaps a "plugin" tends to be a

bundle of objects with a user-visible name and can be downloaded/packaged as a unit.

7.

How fast is the D-Bus reference implementation?

Of course it depends a bit on what you're doing.

<http://lists.freedesktop.org/pipermail/dbus/2004-November/001779.html>

This mail contains some benchmarking. At the time of that benchmark, D-Bus one-to-one communication was about 2.5x slower than simply pushing the data raw over a socket. After the recent rewrite of the marshaling code, D-Bus is slower than that because a lot of optimization work was lost. But it can probably be sped up again.

D-Bus communication with the intermediate bus daemon should be (and as last profiled, was) about twice as slow as one-to-one mode, because a round trip involves four socket reads/writes rather than two socket reads/writes.

The overhead comes from a couple of places; part of it is "abstraction penalty" (there are layers of code to support multiple main loops, multiple transport types, security, etc.).

Probably the largest part comes from data validation (because the reference implementation does not trust incoming data).

It would be simple to add a "no validation" mode, but probably not a good idea all things considered.

Raw bandwidth isn't the only concern; D-Bus is designed to enable asynchronous communication and avoid round trips. This is frequently a more important performance issue than throughput.

8.

How large is the D-Bus reference implementation?

A production build (with assertions, unit tests, and verbose logging disabled) is on the order of a 150K shared library.

A much, much smaller implementation would be possible by omitting out of memory handling, hardcoding a main loop (or always using blocking I/O), skipping validation, and otherwise simplifying things.

9.

How does D-Bus differ from other interprocess communication or networking protocols?

Keep in mind, it is not only an IPC system; it also includes lifecycle tracking, service activation, security policy, and other higher-level structure and assumptions.

The best place to start is to read the D-Bus [tutorial](#), so you have a concrete idea what D-Bus actually is. If you understand other protocols on a wire format level, you may also want to read the D-Bus [specification](#) to see what D-Bus looks like on a low level.

As the [tutorial](#) and [specification](#) both explain, D-Bus is tuned for some specific use cases. Thus, it probably isn't tuned for what you want to do, unless you are doing the things D-Bus was designed for. Don't make the mistake of thinking that any system involving "IPC" is the same thing.

The D-Bus authors would not recommend using D-Bus for applications where it doesn't make sense. The following questions compare D-Bus to some other protocols primarily to help you understand D-Bus and decide whether it's appropriate; D-Bus is neither intended nor claimed to be the right choice for every application.

It should be possible to bridge D-Bus to other IPC systems, just as D-Bus can be bridged to object systems.

Note: the D-Bus mailing list subscribers are *very much not* interested in debating which IPC system is the One True System. So if you want to discuss that, please use another forum.

[CORBA](#) [10.](#)

How does D-Bus differ from CORBA?

Start by reading [#other-ipc](#) [9.](#)Q: 9

[CORBA](http://www.omg.org) is designed to support object-oriented IPC between objects, automatically marshalling parameters as necessary. CORBA is strongly supported by the [Open Management Group \(OMG\)](http://www.omg.org), which produces various standards and supporting documents for CORBA and has the backing of many large organizations. There are many CORBA ORBs available, both proprietary ORBs and free / open source software ORBs (the latter include [ORBit](http://orbit-resource.sourceforge.net/), [MICO](http://www.mico.org/), and [The ACE Orb \(TAO\)](http://www.theaceorb.com/)). Many organizations continue to use CORBA ORBs for various kinds of IPC.

Both GNOME and KDE have used CORBA and then moved away from it. KDE had more success with a system called DCOP, and GNOME layered a system called Bonobo on top of CORBA. Without custom extensions, CORBA does not support many of the things one wants to do in a desktop environment with the GNOME/KDE architecture.

CORBA on the other hand has a number of features of interest for enterprise and web application development, though XML systems such as SOAP are the latest fad.

</p><p>
Like D-Bus, CORBA uses a fast binary protocol (IIOP). Both systems work in terms of objects and methods, and have concepts such as "oneway" calls. Only D-Bus has direct support for "signals" as in GLib/Qt (or Java listeners, or C# delegates).

</p><p>
D-Bus hardcodes and specifies a lot of things that CORBA leaves open-ended, because CORBA is more generic and D-Bus has two specific use-cases in mind. This makes D-Bus a bit simpler.

</p><p>
However, unlike CORBA D-Bus does not specify the API for the language bindings. Instead, "native" bindings adapted specifically to the conventions of a framework such as GObject, GObject, C#, Java, Python, etc. are encouraged. The libdbus reference implementation is designed to be a backend for bindings of this nature, rather than to be used directly. The rationale is that an IPC system API should not "leak" all over a program; it should come into play only just before data goes over the wire. As an aside, OMG is apparently working on a simpler C++ binding for CORBA.

</p><p>
Many CORBA implementations such as ORBit are faster than the libdbus reference implementation. One reason is that D-Bus considers data from the other end of the connection to be untrusted and extensively validates it. But generally speaking other priorities were placed ahead of raw speed in the libdbus implementation. A fast D-Bus implementation along the lines of ORBit should be possible, of course.

</p><p>
On a more trivial note, D-Bus involves substantially fewer acronyms than CORBA.

</p></td></tr><tr class="question" title="11."><td align="left" valign="top"><p>11.</p></td><td align="left" valign="top"><p>
How does D-Bus differ from XML-RPC and SOAP?
</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>
Start by reading Q: 9.
</p><p>
In SOAP and XML-RPC, RPC calls are transformed
into an XML-based format, then sent over the wire (typically using the HTTP protocol), where they are processed and returned. XML-RPC is the simple protocol (its spec is only a page or two), and SOAP is the full-featured protocol.
</p><p>
XML-RPC and SOAP impose XML parsing overhead that is normally irrelevant in the context of the Internet, but significant for constant fine-grained IPC among applications in a desktop session.
</p><p>
D-Bus offers persistent connections and with the bus daemon supports lifecycle tracking of other applications connected to the bus. With XML-RPC and SOAP, typically each method call exists in isolation and has its own HTTP connection.
</p></td></tr><tr class="question" title="12."><td align="left" valign="top"><p>12.</p></td><td align="left" valign="top"><p>
How does D-Bus differ from DCE?
</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>
Start by reading Q: 9.
</p><p>
Distributed Computing Environment (DCE) is an industry-standard vendor-neutral standard that includes an IPC mechanism. The Open Group
has released an implementation as open source software.
DCE
is quite capable, and includes a vast amount of functionality such as

a distributed time service. As the name implies, DCE is intended for use in a large, multi-computer distributed application. D-Bus would

not be well-suited for this.

</p></td></tr><tr class="question" title="13."><td align="left" valign="top"><p>13.</p></td><td align="left" valign="top"><p>

How does D-Bus differ from DCOM and COM?

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

Start by reading Q: 9.

</p><p>

Comparing D-Bus to COM is apples and oranges;

see Q: 6.

</p><p>

DCOM (distributed COM) is a Windows IPC system designed for use with

the COM object system. It's similar in some ways to DCE and CORBA.

</p></td></tr><tr class="question" title="14."><td align="left" valign="top"><p>14.</p></td><td align="left" valign="top"><p>

How does D-Bus differ from ZeroC's Internet Communications Engine (Ice)

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

Start by reading Q: 9.

</p><p>

The Internet

Communications Engine (Ice) is a powerful IPC mechanism more

on the level of SOAP or CORBA than D-Bus. Ice has a "dual-

license" business around it; i.e. you can use it under the GPL, or pay for a

proprietary license.

</p></td></tr><tr class="question" title="15."><td align="left" valign="top"><p>15.</p></td><td align="left" valign="top"><p>

How does D-Bus differ from Inter-Client Exchange (ICE)?

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

<a class="ulink"

href="http://www.x.org/X11R6.8.1/docs/ICE/ice.pdf" target="_top">ICE

was developed for the X Session Management protocol (XSMP), as part of the X Window System (X11R6.1). The idea was to allow desktop sessions

to contain nongraphical clients in addition to X clients.

ICE is a binary protocol designed for desktop use, and KDE's DCOP builds on ICE. ICE is substantially simpler than D-Bus (in contrast to most of the other IPC systems mentioned here, which are more complex). ICE doesn't really define a mapping to objects and methods (DCOP adds that layer). The reference implementation of ICE (libICE)

is often considered to be horrible (and horribly insecure).

DCOP and XSMP are the only two widely-used applications of ICE, and both could in principle be replaced by D-Bus. (Though whether

GNOME and KDE will bother is an open question.)

[16.](#)

How does D-Bus differ from DCOP?

Start by reading [Q: 9](#).

D-Bus is intentionally pretty similar to [DCOP](#),

and can be thought of as a "DCOP the next generation" suitable for

sharing between the various open source desktop projects.

D-Bus is a bit more complex than DCOP, though the Qt binding for D-Bus

should not be more complex for programmers. The additional complexity

of D-Bus arises from its separation of object references vs. bus names

vs. interfaces as distinct concepts, and its support for one-to-one

connections in addition to connections over the bus. The libdbus

reference implementation has a lot of API to support multiple bindings

and main loops, and performs data validation and out-of-memory handling

in order to support secure applications such as the systemwide bus.

</p><p>

D-Bus is probably somewhat slower than DCOP due to data validation

and more "layers" in the reference implementation. A comparison

hasn't been posted to the list though.

</p><p>

At this time, KDE has not committed to using D-Bus, but there have

been discussions of KDE bridging D-Bus and DCOP, or even changing

DCOP's implementation to use D-Bus internally (so that GNOME and KDE

would end up using exactly the same bus). See the KDE mailing list

archives for some of these discussions.

</p></td></tr><tr class="question" title="17."><td align="left" valign="top"><p>17.</p></td><td align="left" valign="top"><p>

How does D-Bus differ from [yet more IPC mechanisms]?

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

Start by reading Q: 9.

</p><p>

There are countless uses of network sockets in the world. MBUS, Sun ONC/RPC, Jabber/XMPP,

SIP, are some we can think of quickly.

</p></td></tr><tr class="question" title="18."><td align="left" valign="top"><p>18.</p></td><td align="left" valign="top"><p>

Which IPC mechanism should I use?

</p></td></tr><tr class="answer"><td align="left" valign="top"></td><td align="left" valign="top"><p>

Start by reading Q: 9.

</p><p>

If you're writing an Internet or Intranet application, XML-RPC or SOAP

work for many people. These are standard, available for most languages, simple to debug and easy to use.

</p><p>

If you're writing a desktop application for UNIX, then D-Bus is of course our recommendation for talking to other parts of the desktop session.

```

</p><p>
    D-Bus is also designed for communications between system
daemons and
    communications between the desktop and system daemons.
</p><p>
    If you're doing something complicated such as clustering,
    distributed swarms, peer-to-peer, or whatever then
    the authors of this FAQ don't have expertise in these
    areas and you should ask someone else or try a search
engine.
    D-Bus is most likely a poor choice but could be appropriate
    for some things.
</p><p>
    Note: the D-Bus mailing list is probably not the place to
    discuss which system is appropriate for your application,
    though you are welcome to ask specific questions about
    D-Bus <span class="emphasis"><em>after reading this FAQ, the
tutorial, and
        searching the list archives</em></span>. The best way
        to search the list archives is probably to use
        an Internet engine such as Google. On Google,
        include "site:freedesktop.org" in your search.
</p></td></tr><tr class="question" title="19."><td
align="left" valign="top"><a name="idp4836432"></a><a
name="idp4836688"></a><p><b>19.</b></p></td><td align="left"
valign="top"><p>
    How can I submit a bug or patch?
</p></td></tr><tr class="answer"><td align="left"
valign="top"></td><td align="left" valign="top"><p>
    The D-Bus <a class="ulink"
href="http://dbus.freedesktop.org" target="_top">web site</a>
    has a link to the bug tracker, which is the best place to
store
    patches. You can also post them to the list, especially if
you want
    to discuss the patch or get feedback.
</p></td></tr></tbody></table></div></div></body></html>

```

File = dbus-faq.xml

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
"http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd"
[
]>

<article id="index">
  <articleinfo>
    <title>D-Bus FAQ</title>
    <releaseinfo>Version 0.3</releaseinfo>

```

```
<date>17 November 2006</date>
<authorgroup>
  <author>
    <firstname>Havoc</firstname>
    <surname>Pennington</surname>
    <affiliation>
      <orgname>Red Hat, Inc.</orgname>
      <address>
        <email>hp@pobox.com</email>
      </address>
    </affiliation>
  </author>
  <author>
    <firstname>David</firstname>
    <othername role="mi">A</othername>
    <surname>Wheeler</surname>
  </author>
</authorgroup>
</articleinfo>
```

```
<qandaset id="faq">
```

```
<qandaentry>
  <question>
    <para>
      What is D-Bus?
    </para>
  </question>
  <answer>
    <para>
      This is probably best answered by reading the D-Bus <ulink
      url="dbus-tutorial.html">tutorial</ulink> or
      the introduction to the <ulink url="dbus-
      specification.html">specification</ulink>. In
      short, it is a system consisting of 1) a wire protocol for
      exposing a
      typical object-oriented language/framework to other
      applications; and
      2) a bus daemon that allows applications to find and monitor
      one another.
      Phrased differently, D-Bus is 1) an interprocess
      communication (IPC) system and 2) some higher-level
      structure (lifecycle tracking, service activation, security
      policy) provided by two bus daemons,
      one systemwide and one per-user-session.
    </para>
  </answer>
</qandaentry>

<qandaentry>
  <question>
    <para>
```

Is D-Bus stable/finished?

The low-level library "libdbus" and the protocol specification are considered ABI stable. The [README](#) file has a discussion of the API/ABI stability guarantees. Higher-level bindings (such as those for Qt, GLib, Python, Java, C#) each have their own release schedules and degree of maturity, not linked to the low-level library and bus daemon release. Check the project page for the binding you're considering to understand that project's policies.

How is the reference implementation licensed? Can I use it in proprietary applications?

The short answer is yes, you can use it in proprietary applications. You should read the [COPYING](#) file, which offers you the choice of two licenses. These are the GPL and the AFL. The GPL requires that your application be licensed under the GPL as well. The AFL is an "X-style" or "BSD-style" license compatible with proprietary licensing, but it does have some requirements; in particular it prohibits you from filing a lawsuit alleging that the D-Bus software infringes your patents *while you continue to use D-Bus*. If you're going to sue, you have to stop using the software. Read the licenses to determine their meaning, this FAQ

entry is not intended to change the meaning or terms of the licenses.

</para>
</answer>
</qandaentry>

<qandaentry>
<question>

<para>
What is the difference between a bus name, and object path, and an interface?

</para>
</question>

<answer>
<para>

If you imagine a C++ program that implements a network service, then the bus name is the hostname of the computer running this C++ program, the object path is a C++ object instance pointer, and an interface is a C++ class (a pure virtual or abstract class, to be exact).

</para>
<para>

In Java terms, the object path is an object reference, and an interface is a Java interface.

</para>
<para>

People get confused because if they write an application with a single object instance and a single interface, then the bus name, object path, and interface look redundant. For example, you might have a text editor that uses the bus name

<literal>org.freedesktop.TextEditor</literal>, has a global singleton object called <literal>/org/freedesktop/TextEditor</literal>, and that singleton object could implement the interface <literal>org.freedesktop.TextEditor</literal>.

</para>
<para>

However, a text editor application could as easily own multiple bus names (for example, <literal>org.kde.KWrite</literal> in addition to generic <literal>TextEditor</literal>), have multiple objects (maybe <literal>/org/kde/documents/4352</literal> where the number changes according to the document), and each object could implement multiple interfaces, such as

<literal>org.freedesktop.DBus.Introspectable</literal>,

```
        <literal>org.freedesktop.BasicTextField</literal>,
        <literal>org.kde.RichTextDocument</literal>.
    </para>
</answer>
</qandaentry>
```

```
<qandaentry id="service">
  <question>
    <para>
      What is a "service"?
    </para>
  </question>
  <answer>
    <para>
      A service is a program that can be launched by the bus
daemon
      to provide some functionality to other programs. Services
      are normally launched according to the bus name they will
      have.
    </para>
    <para>
      People often misuse the word "service" for any
      bus name, but this tends to be ambiguous and confusing so is
discouraged.
      In the D-Bus docs we try to use "service" only when talking
about
      programs the bus knows how to launch, i.e. a service always
has a
      .service file.
    </para>
  </answer>
</qandaentry>
```

```
<qandaentry id="components">
  <question>
    <para>
      Is D-Bus a "component system"?
    </para>
  </question>
  <answer>
    <para>
      It helps to keep these concepts separate in your mind:
    <orderedlist>
      <listitem>
        <para>
          Object/component system
        </para>
      </listitem>
      <listitem>
        <para>
          GUI control/widget embedding interfaces
        </para>
      </listitem>
    </orderedlist>
  </answer>
</qandaentry>
```

```
        </para>
    </listitem>
    <listitem>
        <para>
            Interprocess communication system or wire protocol
        </para>
    </listitem>
</orderedlist>
</para>
<para>
    D-Bus is not a component system. "Component system" was
originally defined by COM, and was essentially a workaround for the
limitations of the C++ object system (adding introspection, runtime
location of objects, ABI guarantees, and so forth). With the C# language
and CLR, Microsoft added these features to the primary object system,
leaving COM obsolete. Similarly, Java has much less need for
something like COM than C++ did. Even QObject (from Qt) and GObject (from
GLib) offer some of the same features found in COM.
</para>
<para>
    Component systems are not about GUI control embedding.
Embedding a spreadsheet in a word processor document is a matter of
defining some specific <emphasis>interfaces</emphasis> that objects
can implement. These interfaces provide methods related to
GUI controls. So an object implementing those interfaces
can be embedded.
</para>
<para>
    The word "component" just means "object with some fancy
features" and in modern languages all objects are effectively
"components."
</para>
<para>
    So components are fancy objects, and some objects are GUI
controls.
</para>
<para>
    A third, unrelated feature is interprocess communication or
IPC.
    D-Bus is an IPC system. Given an object (or "component" if
you must),
```


you can expose the functionality of that object over an IPC system.

Examples of IPC systems are DCOM, CORBA, SOAP, XML-RPC, and D-Bus.

You can use any of these IPC systems with any object/component system, though some of them are "tuned" for specific object systems. You can think of an IPC system primarily as a wire protocol.

If you combine an IPC system with a set of GUI control interfaces, then you can have an out-of-process or dynamically-loaded GUI control.

Another related concept is the `<firstterm>plugin</firstterm>` or `<firstterm>extension</firstterm>`. Generic plugin systems such as the

`<ulink url="http://eclipse.org">Eclipse</ulink>` system are not so different

from component/object systems, though perhaps a "plugin" tends to be a

bundle of objects with a user-visible name and can be downloaded/packaged as a unit.

`</para>`
`</answer>`
`</qandaentry>`

`<qandaentry id="speed">`

`<question>`

`<para>`

How fast is the D-Bus reference implementation?

`</para>`

`</question>`

`<answer>`

`<para>`

Of course it depends a bit on what you're doing.

`<ulink url="http://lists.freedesktop.org/pipermail/dbus/2004-November/001779.html">`

This mail`</ulink>` contains some benchmarking. At the time of that

benchmark, D-Bus one-to-one communication was about 2.5x slower than

simply pushing the data raw over a socket. After the recent rewrite of

the marshaling code, D-Bus is slower than that because a lot of

optimization work was lost. But it can probably be sped up again.

</para>
<para>
D-Bus communication with the intermediate bus daemon should
be
(and as last profiled, was) about twice as slow as one-to-
one
mode, because a round trip involves four socket reads/writes
rather
than two socket reads/writes.

</para>
<para>
The overhead comes from a couple of places; part of it is
simply
"abstraction penalty" (there are layers of code to support
multiple main loops, multiple transport types, security,
etc.).

Probably the largest part comes from data validation
(because the reference implementation does not trust
incoming data).
It would be simple to add a "no validation" mode, but
probably
not a good idea all things considered.

</para>
<para>
Raw bandwidth isn't the only concern; D-Bus is designed to
enable asynchronous communication and avoid round trips.
This is frequently a more important performance issue
than throughput.
</para>
</answer>
</qandaentry>

<qandaentry id="size">
<question>
<para>
How large is the D-Bus reference implementation?
</para>
</question>
<answer>
<para>
A production build (with assertions, unit tests, and verbose
logging
disabled) is on the order of a 150K shared library.
</para>
<para>
A much, much smaller implementation would be possible by
omitting out
of memory handling, hardcoding a main loop (or always using
blocking
I/O), skipping validation, and otherwise simplifying things.
</para>

</answer>
</qandaentry>

<qandaentry id="other-ipc">

<question>

<para>

How does D-Bus differ from other interprocess communication or networking protocols?

</para>

</question>

<answer>

<para>

Keep in mind, it is not only an IPC system; it also includes lifecycle tracking, service activation, security policy, and

other

higher-level structure and assumptions.

</para>

<para>

The best place to start is to read the D-Bus [tutorial](#), so you have a concrete idea what D-Bus actually is. If you understand other protocols on a wire format level, you may also want to read the D-Bus [specification](#) to see what D-Bus looks like on a low level.

</para>

<para>

As the [tutorial](#) and [specification](#) both explain, D-Bus is tuned

for some specific use cases. Thus, it probably isn't tuned for what you want to do, unless you are doing the things D-Bus was designed for. Don't make the mistake of thinking that any system involving "IPC" is the same thing.

</para>

<para>

The D-Bus authors would not recommend using D-Bus for applications where it doesn't make sense. The following questions compare D-Bus to some other protocols primarily to help you understand D-Bus and decide whether it's appropriate; D-Bus is neither

intended

nor claimed to be the right choice for every application.

</para>

<para>

It should be possible to bridge D-Bus to other IPC systems, just as D-Bus can be bridged to object systems.

</para>

<para>

Note: the D-Bus mailing list subscribers are *very* much not

interested

One True System. So if you want to discuss that, please use another forum.

</para>
</answer>
</qandaentry>

<qandaentry id="corba">

<question>

<para>

How does D-Bus differ from CORBA?

</para>

</question>

<answer>

<para>

Start by reading [other-ipc](#).

</para>

<para>

[CORBA](http://www.omg.org) is designed to support

object-oriented IPC between objects, automatically marshalling

parameters as necessary. CORBA is strongly supported by the

[Open Management Group \(OMG\)](http://www.omg.org), which

produces various standards and supporting documents for CORBA and has

the backing of many large organizations. There are many CORBA ORBs

available, both proprietary ORBs and free / open source software ORBs

(the latter include [ORBit](http://orbit-resource.sourceforge.net/),

[MICO](http://www.mico.org/), and [The ACE Orb \(TAO\)](http://www.theaceorb.com/)).

Many organizations continue to use CORBA ORBs for various kinds of

IPC.

</para>

<para>

Both GNOME and KDE have used CORBA and then moved away from it. KDE

had more success with a system called DCOP, and GNOME layered a system

called Bonobo on top of CORBA. Without custom extensions, CORBA does

not support many of the things one wants to do in a desktop environment with the GNOME/KDE architecture.

</para>
<para>
CORBA on the other hand has a number of features of interest
for
enterprise and web application development, though XML
systems such as
SOAP are the latest fad.
</para>
<para>
Like D-Bus, CORBA uses a fast binary protocol (IIOP). Both
systems
work in terms of objects and methods, and have concepts such
as
"oneway" calls. Only D-Bus has direct support for "signals"
as in
Glib/Qt (or Java listeners, or C# delegates).
</para>
<para>
D-Bus hardcodes and specifies a lot of things that CORBA
leaves open-ended,
because CORBA is more generic and D-Bus has two specific
use-cases in mind.
This makes D-Bus a bit simpler.
</para>
<para>
However, unlike CORBA D-Bus does <emphasis>not</emphasis>
specify the
API for the language bindings. Instead, "native" bindings
adapted
specifically to the conventions of a framework such as
QObject,
GObject, C#, Java, Python, etc. are encouraged. The libdbus
reference
implementation is designed to be a backend for bindings of
this
nature, rather than to be used directly. The rationale is
that an IPC
system API should not "leak" all over a program; it should
come into
play only just before data goes over the wire. As an aside,
OMG is
apparently working on a simpler C++ binding for CORBA.
</para>
<para>
Many CORBA implementations such as ORBit are faster than the
libdbus
reference implementation. One reason is that D-Bus
considers data
from the other end of the connection to be untrusted and
extensively
validates it. But generally speaking other priorities were
placed

ahead of raw speed in the libdbus implementation. A fast D-Bus implementation along the lines of ORBit should be possible, of course.

</para>
<para>
On a more trivial note, D-Bus involves substantially fewer acronyms than CORBA.

</para>
</answer>
</qandaentry>

<qandaentry id="xmlrpcsoap">

<question>

<para>

How does D-Bus differ from XML-RPC and SOAP?

</para>

</question>

<answer>

<para>

Start by reading <xref linkend="other-ipc"/>.

</para>

<para>

In <ulink url="http://www.w3.org/TR/SOAP/">SOAP</ulink> and <ulink url="http://www.xmlrpc.com">XML-RPC</ulink>, RPC calls are transformed into an XML-based format, then sent over the wire (typically using the HTTP protocol), where they are processed and returned. XML-RPC is the simple protocol (its spec is only a page or two), and SOAP is the full-featured protocol.

</para>

<para>

XML-RPC and SOAP impose XML parsing overhead that is normally irrelevant in the context of the Internet, but significant for constant fine-grained IPC among applications in a desktop session.

</para>

<para>

D-Bus offers persistent connections and with the bus daemon supports lifecycle tracking of other applications connected to the bus. With XML-RPC and SOAP, typically each method

call

exists in isolation and has its own HTTP connection.

</para>

</answer>
</qandaentry>

<qandaentry id="dce">

<question>

<para>

How does D-Bus differ from DCE?

</para>

</question>

<answer>

<para>

Start by reading <xref linkend="other-ipc"/>.

</para>

<para>

<ulink url="http://www.opengroup.org/dce/">Distributed
Computing Environment (DCE)</ulink> is an industry-standard vendor-
neutral standard that includes an IPC mechanism. <ulink
url="http://www.opengroup.org/comm/press/05-01-12.htm">The
Open Group has released an implementation as open source
software</ulink>. DCE is quite capable, and includes a vast amount of
functionality such as a distributed time service. As the name implies, DCE is
intended for use in a large, multi-computer distributed application. D-
Bus would not be well-suited for this.

</para>

</answer>

</qandaentry>

<qandaentry id="dcom">

<question>

<para>

How does D-Bus differ from DCOM and COM?

</para>

</question>

<answer>

<para>

Start by reading <xref linkend="other-ipc"/>.

</para>

<para>

Comparing D-Bus to COM is apples and oranges;
see <xref linkend="components"/>.

</para>

<para>

DCOM (distributed COM) is a Windows IPC system designed for
use with

the COM object system. It's similar in some ways to DCE and CORBA.

</para>
</answer>
</qandaentry>

<qandaentry id="internet-communications-engine">

<question>

<para>

How does D-Bus differ from ZeroC's Internet Communications Engine (Ice)

</para>

</question>

<answer>

<para>

Start by reading [other-ipc](#).

</para>

<para>

The [Internet Communications Engine \(Ice\)](http://www.zeroc.com/ice.html) is a powerful IPC

mechanism more

on the level of SOAP or CORBA than D-Bus. Ice has a "dual-
license"

business around it; i.e. you can use it under the GPL, or
pay for a

proprietary license.

</para>

</answer>

</qandaentry>

<qandaentry id="inter-client-exchange">

<question>

<para>

How does D-Bus differ from Inter-Client Exchange (ICE)?

</para>

</question>

<answer>

<para>

[ICE](http://www.x.org/X11R6.8.1/docs/ICE/ice.pdf)

was developed for the X Session Management protocol (XSMP),
as part of

the X Window System (X11R6.1). The idea was to allow desktop
sessions

to contain nongraphical clients in addition to X clients.

</para>

<para>

ICE is a binary protocol designed for desktop use, and KDE's

DCOP

builds on ICE. ICE is substantially simpler than D-Bus (in
contrast

more methods (libICE) to most of the other IPC systems mentioned here, which are complex). ICE doesn't really define a mapping to objects and (DCOP adds that layer). The reference implementation of ICE is often considered to be horrible (and horribly insecure).

DCOP and XSMP are the only two widely-used applications of ICE, and both could in principle be replaced by D-Bus. (Though GNOME and KDE will bother is an open question.)

<qandaentry id="dcop">

<question>

<para>

How does D-Bus differ from DCOP?

</para>

</question>

<answer>

<para>

Start by reading <xref linkend="other-ipc"/>.

</para>

<para>

D-Bus is intentionally pretty similar to <ulink

url="http://developer.kde.org/documentation/library/kdeqt/dcop.html">DCOP</ulink>,

and can be thought of as a "DCOP the next generation" suitable for

sharing between the various open source desktop projects.

</para>

<para>

D-Bus is a bit more complex than DCOP, though the Qt binding for D-Bus

should not be more complex for programmers. The additional complexity

of D-Bus arises from its separation of object references vs. bus names

vs. interfaces as distinct concepts, and its support for one-to-one

connections in addition to connections over the bus. The libdbus

reference implementation has a lot of API to support multiple bindings

and main loops, and performs data validation and out-of-memory handling

in order to support secure applications such as the systemwide bus.

</para>

<para>

D-Bus is probably somewhat slower than DCOP due to data validation

and more "layers" in the reference implementation. A comparison

hasn't been posted to the list though.

</para>

<para>

At this time, KDE has not committed to using D-Bus, but there have

been discussions of KDE bridging D-Bus and DCOP, or even changing

DCOP's implementation to use D-Bus internally (so that GNOME and KDE

would end up using exactly the same bus). See the KDE mailing list

archives for some of these discussions.

</para>

</answer>

</qandaentry>

<qandaentry id="yet-more-ipc">

<question>

<para>

How does D-Bus differ from [yet more IPC mechanisms]?

</para>

</question>

<answer>

<para>

Start by reading <xref linkend="other-ipc"/>.

</para>

<para>

There are countless uses of network sockets in the world.

<ulink

url="http://www.mbus.org/">MBUS</ulink>, Sun ONC/RPC, Jabber/XMPP,

SIP, are some we can think of quickly.

</para>

</answer>

</qandaentry>

<qandaentry id="which-ipc">

<question>

<para>

Which IPC mechanism should I use?

```
</para>
</question>
<answer>
  <para>
    Start by reading <xref linkend="other-ipc"/>.
  </para>
  <para>
    If you're writing an Internet or Intranet application, XML-
RPC or SOAP
    work for many people. These are standard, available for most
    languages, simple to debug and easy to use.
  </para>
  <para>
    If you're writing a desktop application for UNIX,
    then D-Bus is of course our recommendation for
    talking to other parts of the desktop session.
  </para>
  <para>
    D-Bus is also designed for communications between system
daemons and
    communications between the desktop and system daemons.
  </para>
  <para>
    If you're doing something complicated such as clustering,
    distributed swarms, peer-to-peer, or whatever then
    the authors of this FAQ don't have expertise in these
    areas and you should ask someone else or try a search
engine.
    D-Bus is most likely a poor choice but could be appropriate
    for some things.
  </para>
  <para>
    Note: the D-Bus mailing list is probably not the place to
    discuss which system is appropriate for your application,
    though you are welcome to ask specific questions about
    D-Bus <emphasis>after reading this FAQ, the tutorial, and
    searching the list archives</emphasis>. The best way
    to search the list archives is probably to use
    an Internet engine such as Google. On Google,
    include "site:freedesktop.org" in your search.
  </para>
</answer>
</qandaentry>

<qandaentry>
  <question>
    <para>
      How can I submit a bug or patch?
    </para>
  </question>
  <answer>
```

```
<para>
    The D-Bus <ulink url="http://dbus.freedesktop.org">web
site</ulink>
    has a link to the bug tracker, which is the best place to
store
    patches. You can also post them to the list, especially if
you want
    to discuss the patch or get feedback.
</para>
</answer>
</qandaentry>

</qandaset>

</article>
```

```
File = dbus-file-unix.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-file-unix.c unix related file implementation (internal to D-
Bus implementation)
*
* Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
* Copyright (C) 2003 CodeFactory AB
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>

#include "dbus-protocol.h"
#include "dbus-errors.h"
```

```

#include "dbus-file.h"
#include "dbus-internals.h"
#include "dbus-sysdeps.h"
#include "dbus-sysdeps-unix.h"

#include <sys/stat.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>

#ifdef O_BINARY
#define O_BINARY 0
#endif

/**
 * Appends the contents of the given file to the string,
 * returning error code. At the moment, won't open a file
 * more than a megabyte in size.
 *
 * @param str the string to append to
 * @param filename filename to load
 * @param error place to set an error
 * @returns #FALSE if error was set
 */
dbus_bool_t
_dbus_file_get_contents (DBusString      *str,
                        const DBusString *filename,
                        DBusError       *error)
{
    int fd;
    struct stat sb;
    int orig_len;
    int total;
    const char *filename_c;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);

    /* O_BINARY useful on Cygwin */
    fd = open (filename_c, O_RDONLY | O_BINARY);
    if (fd < 0)
    {
        dbus_set_error (error, _dbus_error_from_errno (errno),
                       "Failed to open \"%s\": %s",
                       filename_c,
                       _dbus_strerror (errno));
        return FALSE;
    }

    _dbus_verbose ("file fd %d opened\n", fd);

```

```

if (fstat (fd, &sb) < 0)
{
    dbus_set_error (error, _dbus_error_from_errno (errno),
                    "Failed to stat \"%s\": %s",
                    filename_c,
                    _dbus_strerror (errno));

    _dbus_verbose ("fstat() failed: %s",
                  _dbus_strerror (errno));

    _dbus_close (fd, NULL);

    return FALSE;
}

if (sb.st_size > _DBUS_ONE_MEGABYTE)
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                    "File size %lu of \"%s\" is too large.",
                    (unsigned long) sb.st_size, filename_c);
    _dbus_close (fd, NULL);
    return FALSE;
}

total = 0;
orig_len = _dbus_string_get_length (str);
if (sb.st_size > 0 && S_ISREG (sb.st_mode))
{
    int bytes_read;

    while (total < (int) sb.st_size)
    {
        bytes_read = _dbus_read (fd, str,
                                sb.st_size - total);
        if (bytes_read <= 0)
        {
            dbus_set_error (error, _dbus_error_from_errno (errno),
                            "Error reading \"%s\": %s",
                            filename_c,
                            _dbus_strerror (errno));

            _dbus_verbose ("read() failed: %s",
                          _dbus_strerror (errno));

            _dbus_close (fd, NULL);
            _dbus_string_set_length (str, orig_len);
            return FALSE;
        }
        else
            total += bytes_read;
    }
}

```

```

        _dbus_close (fd, NULL);
        return TRUE;
    }
    else if (sb.st_size != 0)
    {
        _dbus_verbose ("Can only open regular files at the moment.\n");
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "\'%s\' is not a regular file",
                       filename_c);
        _dbus_close (fd, NULL);
        return FALSE;
    }
    else
    {
        _dbus_close (fd, NULL);
        return TRUE;
    }
}

/**
 * Writes a string out to a file. If the file exists,
 * it will be atomically overwritten by the new data.
 *
 * @param str the string to write out
 * @param filename the file to save string to
 * @param world_readable If set, ensure the file is world readable
 * @param error to be filled in on failure
 * @returns #FALSE on failure
 */
dbus_bool_t
_dbus_string_save_to_file (const DBusString *str,
                          const DBusString *filename,
                          dbus_bool_t world_readable,
                          DBusError *error)
{
    int fd;
    int bytes_to_write;
    const char *filename_c;
    DBusString tmp_filename;
    const char *tmp_filename_c;
    int total;
    dbus_bool_t need_unlink;
    dbus_bool_t retval;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    fd = -1;
    retval = FALSE;
    need_unlink = FALSE;

    if (!_dbus_string_init (&tmp_filename))

```



```

        goto out;
    }
}

_dbus_verbose ("tmp file fd %d opened\n", fd);

need_unlink = TRUE;

total = 0;
bytes_to_write = _dbus_string_get_length (str);

while (total < bytes_to_write)
{
    int bytes_written;

    bytes_written = _dbus_write (fd, str, total,
                                bytes_to_write - total);

    if (bytes_written <= 0)
    {
        dbus_set_error (error, _dbus_error_from_errno (errno),
                        "Could not write to %s: %s", tmp_filename_c,
                        _dbus_strerror (errno));

        goto out;
    }

    total += bytes_written;
}

if (fsync(fd))
{
    dbus_set_error (error, _dbus_error_from_errno (errno),
                    "Could not synchronize file %s: %s",
                    tmp_filename_c, _dbus_strerror (errno));

    goto out;
}

if (!_dbus_close (fd, NULL))
{
    dbus_set_error (error, _dbus_error_from_errno (errno),
                    "Could not close file %s: %s",
                    tmp_filename_c, _dbus_strerror (errno));

    goto out;
}

fd = -1;

if (rename (tmp_filename_c, filename_c) < 0)
{

```

```

        dbus_set_error (error, _dbus_error_from_errno (errno),
                        "Could not rename %s to %s: %s",
                        tmp_filename_c, filename_c,
                        _dbus_strerror (errno));

        goto out;
    }

    need_unlink = FALSE;

    retval = TRUE;

out:
    /* close first, then unlink, to prevent ".nfs34234235" garbage
     * files
     */

    if (fd >= 0)
        _dbus_close (fd, NULL);

    if (need_unlink && unlink (tmp_filename_c) < 0)
        _dbus_verbose ("Failed to unlink temp file %s: %s\n",
                        tmp_filename_c, _dbus_strerror (errno));

    _dbus_string_free (&tmp_filename);

    if (!retval)
        _DBUS_ASSERT_ERROR_IS_SET (error);

    return retval;
}

/** Makes the file readable by every user in the system.
 *
 * @param filename the filename
 * @param error error location
 * @returns #TRUE if the file's permissions could be changed.
 */
dbus_bool_t
_dbus_make_file_world_readable(const DBusString *filename,
                               DBusError *error)
{
    const char *filename_c;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);
    if (chmod (filename_c, 0644) == -1)
    {
        dbus_set_error (error,
                        DBUS_ERROR_FAILED,
                        "Could not change permissions of file %s: %s\n",

```

```

        filename_c,
        _dbus_strerror (errno));
    return FALSE;
}
return TRUE;
}

/** Creates the given file, failing if the file already exists.
 *
 * @param filename the filename
 * @param error error location
 * @returns #TRUE if we created the file and it didn't exist
 */
dbus_bool_t
_dbus_create_file_exclusively (const DBusString *filename,
                               DBusError *error)
{
    int fd;
    const char *filename_c;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);

    fd = open (filename_c, O_WRONLY | O_BINARY | O_EXCL | O_CREAT,
              0600);
    if (fd < 0)
    {
        dbus_set_error (error,
                       DBUS_ERROR_FAILED,
                       "Could not create file %s: %s\n",
                       filename_c,
                       _dbus_strerror (errno));
        return FALSE;
    }

    _dbus_verbose ("exclusive file fd %d opened\n", fd);

    if (!_dbus_close (fd, NULL))
    {
        dbus_set_error (error,
                       DBUS_ERROR_FAILED,
                       "Could not close file %s: %s\n",
                       filename_c,
                       _dbus_strerror (errno));
        return FALSE;
    }

    return TRUE;
}

/**

```

```

* Deletes the given file.
*
* @param filename the filename
* @param error error location
*
* @returns #TRUE if unlink() succeeded
*/
dbus_bool_t
_dbus_delete_file (const DBusString *filename,
                  DBusError *error)
{
    const char *filename_c;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);

    if (unlink (filename_c) < 0)
    {
        dbus_set_error (error, DBUS_ERROR_FAILED,
                       "Failed to delete file %s: %s\n",
                       filename_c, _dbus_strerror (errno));

        return FALSE;
    }
    else
        return TRUE;
}

```

File = dbus-file-win.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-file-win.c windows related file implementation (internal to D-
Bus implementation)
*
* Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
* Copyright (C) 2003 CodeFactory AB
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

```

```

*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#include "dbus-protocol.h"
#include "dbus-string.h"
#include "dbus-internals.h"
#include "dbus-sysdeps-win.h"
#include "dbus-pipe.h"

#include <windows.h>

/**
 * Thin wrapper around the read() system call that appends
 * the data it reads to the DBusString buffer. It appends
 * up to the given count.
 *
 * @param hnd the HANDLE to read from
 * @param buffer the buffer to append data to
 * @param count the amount of data to read
 * @param error place to set an error
 * @returns the number of bytes read or -1
 */
static int
_dbus_file_read (HANDLE          hnd,
                 DBusString      *buffer,
                 int             count,
                 DBusError       *error)
{
    BOOL result;
    DWORD bytes_read;
    int start;
    char *data;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_assert (count >= 0);

    start = _dbus_string_get_length (buffer);

    if (!_dbus_string_lengthen (buffer, count))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return -1;
    }
}

```

```

data = _dbus_string_get_data_len (buffer, start, count);

result = ReadFile (hnd, data, count, &bytes_read, NULL);
if (result == 0)
{
    char *emsg = _dbus_win_error_string (GetLastError ());
    dbus_set_error (error, _dbus_win_error_from_last_error (),
                   "Failed to read from 0x%x: %s", hnd, emsg);
    _dbus_win_free_error_string (emsg);
    return -1;
}

if (bytes_read)
{
    /* put length back (doesn't actually realloc) */
    _dbus_string_set_length (buffer, start + bytes_read);
}

#ifdef 0
    if (bytes_read > 0)
        _dbus_verbose_bytes_of_string (buffer, start, bytes_read);
#endif
}

return bytes_read;
}

/**
 * Appends the contents of the given file to the string,
 * returning error code. At the moment, won't open a file
 * more than a megabyte in size.
 *
 * @param str the string to append to
 * @param filename filename to load
 * @param error place to set an error
 * @returns #FALSE if error was set
 */
dbus_bool_t
_dbus_file_get_contents (DBusString *str,
                        const DBusString *filename,
                        DBusError *error)
{
    HANDLE hnd;
    DWORD fsize;
    DWORD fsize_hi;
    int orig_len;
    unsigned int total;
    const char *filename_c;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);

```

```

hnd = CreateFileA (filename_c, GENERIC_READ,
                  FILE_SHARE_READ | FILE_SHARE_WRITE,
                  NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
if (hnd == INVALID_HANDLE_VALUE)
{
    char *emsg = _dbus_win_error_string (GetLastError ());
    dbus_set_error (error, _dbus_win_error_from_last_error (),
                  "Failed to open \"%s\": %s", filename_c, emsg);
    _dbus_win_free_error_string (emsg);
    return FALSE;
}

_dbus_verbose ("file %s hnd %p opened\n", filename_c, hnd);

fsize = GetFileSize (hnd, &fsize_hi);
if (fsize == 0xFFFFFFFF && GetLastError() != NO_ERROR)
{
    char *emsg = _dbus_win_error_string (GetLastError ());
    dbus_set_error (error, _dbus_win_error_from_last_error (),
                  "Failed to get file size for \"%s\": %s",
                  filename_c, emsg);
    _dbus_win_free_error_string (emsg);

    _dbus_verbose ("GetFileSize() failed: %s", emsg);

    CloseHandle (hnd);

    return FALSE;
}

if (fsize_hi != 0 || fsize > _DBUS_ONE_MEGABYTE)
{
    dbus_set_error (error, DBUS_ERROR_FAILED,
                  "File size %lu/%lu of \"%s\" is too large.",
                  (unsigned long) fsize_hi,
                  (unsigned long) fsize, filename_c);

    CloseHandle (hnd);
    return FALSE;
}

total = 0;
orig_len = _dbus_string_get_length (str);
if (fsize > 0)
{
    int bytes_read;

    while (total < fsize)
    {
        bytes_read = _dbus_file_read (hnd, str, fsize - total,
error);
        if (bytes_read <= 0)

```

```

        {
            if (bytes_read == 0)
            {
                dbus_set_error (error, DBUS_ERROR_FAILED,
                                "Premature EOF reading \"%s\"",
                                filename_c);
            }
            else
                _DBUS_ASSERT_ERROR_IS_SET (error);

            CloseHandle (hnd);
            _dbus_string_set_length (str, orig_len);
            return FALSE;
        }
        else
            total += bytes_read;
    }

    CloseHandle (hnd);
    return TRUE;
}
else
{
    CloseHandle (hnd);
    return TRUE;
}
}

/**
 * Writes a string out to a file. If the file exists,
 * it will be atomically overwritten by the new data.
 *
 * @param str the string to write out
 * @param filename the file to save string to
 * @param world_readable if true, ensure file is world readable
 * @param error to be filled in on failure
 * @returns #FALSE on failure
 */
dbus_bool_t
_dbus_string_save_to_file (const DBusString *str,
                          const DBusString *filename,
                          dbus_bool_t world_readable,
                          DBusError *error)
{
    HANDLE hnd;
    int bytes_to_write;
    const char *filename_c;
    DBusString tmp_filename;
    const char *tmp_filename_c;
    int total;
    const char *str_c;

```



```

dbus_bool_t need_unlink;
dbus_bool_t retval;

_DBUS_ASSERT_ERROR_IS_CLEAR (error);

hnd = INVALID_HANDLE_VALUE;
retval = FALSE;
need_unlink = FALSE;

if (!_dbus_string_init (&tmp_filename))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    return FALSE;
}

if (!_dbus_string_copy (filename, 0, &tmp_filename, 0))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&tmp_filename);
    return FALSE;
}

if (!_dbus_string_append (&tmp_filename, "."))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&tmp_filename);
    return FALSE;
}

#define N_TMP_FILENAME_RANDOM_BYTES 8
if (!_dbus_generate_random_ascii (&tmp_filename,
N_TMP_FILENAME_RANDOM_BYTES))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&tmp_filename);
    return FALSE;
}

filename_c = _dbus_string_get_const_data (filename);
tmp_filename_c = _dbus_string_get_const_data (&tmp_filename);

/* TODO - support world-readable in an atomic fashion */
hnd = CreateFileA (tmp_filename_c, GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL, CREATE_NEW, FILE_ATTRIBUTE_NORMAL,
INVALID_HANDLE_VALUE);
if (hnd == INVALID_HANDLE_VALUE)
{
    char *emsg = _dbus_win_error_string (GetLastError ());
    dbus_set_error (error, _dbus_win_error_from_last_error (),
"Could not create \"%s\": %s", filename_c,
emsg);
}

```

```

    _dbus_win_free_error_string (emsg);
    goto out;
}
if (world_readable)
{
    if (!_dbus_make_file_world_readable (&tmp_filename, error))
        goto out;
}

_dbus_verbose ("tmp file %s hnd %p opened\n", tmp_filename_c, hnd);

need_unlink = TRUE;

total = 0;
bytes_to_write = _dbus_string_get_length (str);
str_c = _dbus_string_get_const_data (str);

while (total < bytes_to_write)
{
    DWORD bytes_written;
    BOOL res;

    res = WriteFile (hnd, str_c + total, bytes_to_write - total,
                    &bytes_written, NULL);

    if (res == 0 || bytes_written <= 0)
    {
        char *emsg = _dbus_win_error_string (GetLastError ());
        dbus_set_error (error, _dbus_win_error_from_last_error (),
                       "Could not write to %s: %s",
tmp_filename_c, emsg);
        _dbus_win_free_error_string (emsg);
        goto out;
    }

    total += bytes_written;
}

if (CloseHandle (hnd) == 0)
{
    char *emsg = _dbus_win_error_string (GetLastError ());
    dbus_set_error (error, _dbus_win_error_from_last_error (),
                   "Could not close file %s: %s", tmp_filename_c,
emsg);
    _dbus_win_free_error_string (emsg);
    goto out;
}

hnd = INVALID_HANDLE_VALUE;

/* Unlike rename(), MoveFileEx() can replace existing files */

```

```

    if (!MoveFileExA (tmp_filename_c, filename_c,
MOVEFILE_REPLACE_EXISTING))
    {
        char *emsg = _dbus_win_error_string (GetLastError ());
        dbus_set_error (error, _dbus_win_error_from_last_error (),
            "Could not rename %s to %s: %s",
            tmp_filename_c, filename_c, emsg);
        _dbus_win_free_error_string (emsg);

        goto out;
    }

    need_unlink = FALSE;

    retval = TRUE;

out:
    /* close first, then unlink */

    if (hnd != INVALID_HANDLE_VALUE)
        CloseHandle (hnd);

    if (need_unlink && DeleteFileA (tmp_filename_c) == 0)
    {
        char *emsg = _dbus_win_error_string (GetLastError ());
        _dbus_verbose ("Failed to unlink temp file %s: %s",
tmp_filename_c,
            emsg);
        _dbus_win_free_error_string (emsg);
    }

    _dbus_string_free (&tmp_filename);

    if (!retval)
        _DBUS_ASSERT_ERROR_IS_SET (error);

    return retval;
}

/** Creates the given file, failing if the file already exists.
 *
 * @param filename the filename
 * @param error error location
 * @returns #TRUE if we created the file and it didn't exist
 */
dbus_bool_t
_dbus_create_file_exclusively (const DBusString *filename,
                                DBusError *error)
{
    HANDLE hnd;
    const char *filename_c;

```

```

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    filename_c = _dbus_string_get_const_data (filename);

    hnd = CreateFileA (filename_c, GENERIC_WRITE,
                      FILE_SHARE_READ | FILE_SHARE_WRITE,
                      NULL, CREATE_NEW, FILE_ATTRIBUTE_NORMAL,
                      INVALID_HANDLE_VALUE);
    if (hnd == INVALID_HANDLE_VALUE)
    {
        char *emsg = _dbus_win_error_string (GetLastError ());
        dbus_set_error (error, _dbus_win_error_from_last_error (),
                       "Could not create file %s: %s",
                       filename_c, emsg);
        _dbus_win_free_error_string (emsg);
        return FALSE;
    }

    _dbus_verbose ("exclusive file %s hnd %p opened\n", filename_c,
hnd);

    if (CloseHandle (hnd) == 0)
    {
        char *emsg = _dbus_win_error_string (GetLastError ());
        dbus_set_error (error, _dbus_win_error_from_last_error (),
                       "Could not close file %s: %s",
                       filename_c, emsg);
        _dbus_win_free_error_string (emsg);

        return FALSE;
    }

    return TRUE;
}

```

File = dbus-file.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-sysdeps-unix.c Wrappers around UNIX system/libc features
(internal to D-Bus implementation)
*
* Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
* Copyright (C) 2003 CodeFactory AB
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify

```

```
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/
```

```
#include <config.h>
```

```
#include "dbus-file.h"
```

```
File = dbus-file.h
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-file.h dbus file related stuff (internal to D-Bus
implementation)
*
* Copyright (C) 2002, 2003 Red Hat, Inc.
* Copyright (C) 2003 CodeFactory AB
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/
```

```

*/

#ifndef DBUS_FILE_H
#define DBUS_FILE_H

//#include <dbus/dbus-types.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-errors.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusFile
 * @{
 */

/**
 * File interface
 */
dbus_bool_t _dbus_file_exists (const char *file);
dbus_bool_t _dbus_file_get_contents (DBusString *str,
                                     const DBusString *filename,
                                     DBusError *error);
dbus_bool_t _dbus_string_save_to_file (const DBusString *str,
                                     const DBusString *filename,
                                     dbus_bool_t
world_readable,
                                     DBusError *error);

dbus_bool_t _dbus_make_file_world_readable (const DBusString
*filename,
                                     DBusError *error);

dbus_bool_t _dbus_create_file_exclusively (const DBusString
*filename,
                                     DBusError
*error);
dbus_bool_t _dbus_delete_file (const DBusString
*filename,
                                     DBusError
*error);

/** @} */

DBUS_END_DECLS

#endif

```

File = dbus-gconnection.sgml

```
<!-- ##### SECTION Title ##### -->
DBusGConnection

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->
<para>

</para>

<!-- ##### SECTION See_Also ##### -->
<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### STRUCT DBusGConnection ##### -->
<para>

</para>

<!-- ##### MACRO DBUS_TYPE_G_CONNECTION ##### -->
<para>

</para>

<!-- ##### FUNCTION dbus_g_bus_get ##### -->
<para>

</para>

@type:
@error:
@Returns:

<!-- ##### FUNCTION dbus_g_bus_get_private ##### -->
<para>

</para>
```

@type:
@context:
@error:
@Returns:

<!-- ##### FUNCTION dbus_g_thread_init ##### -->

<para>

</para>

@void:

<!-- ##### FUNCTION dbus_g_connection_open ##### -->

<para>

</para>

@address:

@error:

@Returns:

<!-- ##### FUNCTION dbus_g_connection_ref ##### -->

<para>

</para>

@connection:

@Returns:

<!-- ##### FUNCTION dbus_g_connection_unref ##### -->

<para>

</para>

@connection:

<!-- ##### FUNCTION dbus_g_connection_flush ##### -->

<para>

</para>

@connection:

<!-- ##### FUNCTION dbus_g_connection_get_connection ##### -->

<para>

</para>

@gconnection:
@Returns:

<!-- ##### FUNCTION dbus_g_connection_register_g_object ##### -->
<para>

</para>

@connection:
@at_path:
@object:

<!-- ##### FUNCTION dbus_g_connection_unregister_g_object ##### -->
<para>

</para>

@connection:
@object:

<!-- ##### FUNCTION dbus_g_connection_lookup_g_object ##### -->
<para>

</para>

@connection:
@at_path:
@Returns:

File = dbus-gerror.sgml

<!-- ##### SECTION Title ##### -->
DBusGError

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->
<para>

</para>

<!-- ##### SECTION See_Also ##### -->

<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### ENUM DBusGError ##### -->

<para>

</para>

@DBUS_GERROR_FAILED:

@DBUS_GERROR_NO_MEMORY:

@DBUS_GERROR_SERVICE_UNKNOWN:

@DBUS_GERROR_NAME_HAS_NO_OWNER:

@DBUS_GERROR_NO_REPLY:

@DBUS_GERROR_IO_ERROR:

@DBUS_GERROR_BAD_ADDRESS:

@DBUS_GERROR_NOT_SUPPORTED:

@DBUS_GERROR_LIMITS_EXCEEDED:

@DBUS_GERROR_ACCESS_DENIED:

@DBUS_GERROR_AUTH_FAILED:

@DBUS_GERROR_NO_SERVER:

@DBUS_GERROR_TIMEOUT:

@DBUS_GERROR_NO_NETWORK:

@DBUS_GERROR_ADDRESS_IN_USE:

@DBUS_GERROR_DISCONNECTED:

@DBUS_GERROR_INVALID_ARGS:

@DBUS_GERROR_FILE_NOT_FOUND:

@DBUS_GERROR_FILE_EXISTS:

@DBUS_GERROR_UNKNOWN_METHOD:

@DBUS_GERROR_TIMED_OUT:

@DBUS_GERROR_MATCH_RULE_NOT_FOUND:

@DBUS_GERROR_MATCH_RULE_INVALID:

@DBUS_GERROR_SPAWN_EXEC_FAILED:

@DBUS_GERROR_SPAWN_FORK_FAILED:

@DBUS_GERROR_SPAWN_CHILD_EXITED:

@DBUS_GERROR_SPAWN_CHILD_SIGNALED:

@DBUS_GERROR_SPAWN_FAILED:

@DBUS_GERROR_UNIX_PROCESS_ID_UNKNOWN:

@DBUS_GERROR_INVALID_SIGNATURE:

@DBUS_GERROR_INVALID_FILE_CONTENT:

@DBUS_GERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN:

@DBUS_GERROR_REMOTE_EXCEPTION:

<!-- ##### FUNCTION dbus_g_error_has_name ##### -->

<para>

</para>

@error:

@name:

@Returns:

<!-- ##### FUNCTION dbus_g_error_get_name ##### -->

<para>

</para>

@error:

@Returns:

<!-- ##### FUNCTION dbus_g_error_domain_register ##### -->

<para>

</para>

@domain:

@default_iface:

@code_enum:

<!-- ##### MACRO DBUS_GERROR ##### -->

<para>

</para>

File = dbus-gidl.c

/* -*- mode: C; c-file-style: "gnu" -*- */

/* dbus-gidl.c data structure describing an interface, to be generated
from IDL

* or something

*

* Copyright (C) 2003, 2005 Red Hat, Inc.

*

* Licensed under the Academic Free License version 2.1

*

* This program is free software; you can redistribute it and/or
modify

```
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/
```

```
#include <config.h>
```

```
#include "dbus-gidl.h"
```

```
struct BaseInfo
```

```
{
    unsigned int refcount : 28;
    unsigned int type      : 4;
    char *name;
};
```

```
struct NodeInfo
```

```
{
    BaseInfo base;
    GSList *interfaces;
    GSList *nodes;
};
```

```
struct InterfaceInfo
```

```
{
    BaseInfo base;
    GHashTable *annotations;
    /* Since we have BaseInfo now these could be one list */
    GSList *methods;
    GSList *signals;
    GSList *properties;
};
```

```
struct MethodInfo
```

```
{
    BaseInfo base;
    GHashTable *annotations;
    GSList *args;
};
```

```

struct SignalInfo
{
    BaseInfo base;
    GSList *args;
};

struct PropertyInfo
{
    BaseInfo base;
    char *type;
    PropertyAccessFlags access;
};

struct ArgInfo
{
    BaseInfo base;
    char *type;
    ArgDirection direction;
    GHashTable *annotations;
};

static void
get_hash_key (gpointer key, gpointer value, gpointer data)
{
    GSList **list = data;
    *list = g_slist_prepend (*list, key);
}

static GSList *
get_hash_keys (GHashTable *table)
{
    GSList *ret = NULL;

    g_hash_table_foreach (table, get_hash_key, &ret);

    return ret;
}

BaseInfo *
base_info_ref (BaseInfo *info)
{
    g_return_val_if_fail (info != NULL, NULL);
    g_return_val_if_fail (info->refcount > 0, NULL);

    info->refcount += 1;

    return info;
}

static void
base_info_free (void *ptr)
{

```

```

BaseInfo *info;

info = ptr;

g_free (info->name);
g_free (info);
}

void
base_info_unref (BaseInfo *info)
{
    g_return_if_fail (info != NULL);
    g_return_if_fail (info->refcount > 0);

    /* This is sort of bizarre, BaseInfo was tacked on later */

    switch (info->type)
    {
        case INFO_TYPE_NODE:
            node_info_unref ((NodeInfo*) info);
            break;
        case INFO_TYPE_INTERFACE:
            interface_info_unref ((InterfaceInfo*) info);
            break;
        case INFO_TYPE_SIGNAL:
            signal_info_unref ((SignalInfo*) info);
            break;
        case INFO_TYPE_METHOD:
            method_info_unref ((MethodInfo*) info);
            break;
        case INFO_TYPE_PROPERTY:
            property_info_unref ((PropertyInfo*) info);
            break;
        case INFO_TYPE_ARG:
            arg_info_unref ((ArgInfo*) info);
            break;
    }
}

InfoType
base_info_get_type (BaseInfo *info)
{
    return info->type;
}

const char*
base_info_get_name (BaseInfo *info)
{
    return info->name;
}

void

```

```

base_info_set_name (BaseInfo      *info,
                   const char     *name)
{
    char *old;

    old = info->name;
    info->name = g_strdup (name);
    g_free (old);
}

GType
base_info_get_gtype (void)
{
    static GType our_type = 0;

    if (our_type == 0)
        our_type = g_boxed_type_register_static ("BaseInfo",
                                                (GBoxedCopyFunc)
base_info_ref,
                                                (GBoxedFreeFunc)
base_info_unref);

    return our_type;
}

static void
free_interface_list (GSList **interfaces_p)
{
    GSList *tmp;
    tmp = *interfaces_p;
    while (tmp != NULL)
        {
            interface_info_unref (tmp->data);
            tmp = tmp->next;
        }
    g_slist_free (*interfaces_p);
    *interfaces_p = NULL;
}

static void
free_node_list (GSList **nodes_p)
{
    GSList *tmp;
    tmp = *nodes_p;
    while (tmp != NULL)
        {
            node_info_unref (tmp->data);
            tmp = tmp->next;
        }
    g_slist_free (*nodes_p);
    *nodes_p = NULL;
}

```

```

static void
free_method_list (GSList **methods_p)
{
    GSList *tmp;
    tmp = *methods_p;
    while (tmp != NULL)
        {
            method_info_unref (tmp->data);
            tmp = tmp->next;
        }
    g_slist_free (*methods_p);
    *methods_p = NULL;
}

```

```

static void
free_signal_list (GSList **signals_p)
{
    GSList *tmp;
    tmp = *signals_p;
    while (tmp != NULL)
        {
            signal_info_unref (tmp->data);
            tmp = tmp->next;
        }
    g_slist_free (*signals_p);
    *signals_p = NULL;
}

```

```

static void
free_property_list (GSList **props_p)
{
    GSList *tmp;
    tmp = *props_p;
    while (tmp != NULL)
        {
            property_info_unref (tmp->data);
            tmp = tmp->next;
        }
    g_slist_free (*props_p);
    *props_p = NULL;
}

```

```

NodeInfo*
node_info_new (const char *name)
{
    NodeInfo *info;

    /* name can be NULL */

    info = g_new0 (NodeInfo, 1);
    info->base.refcount = 1;
}

```



```

    info->base.name = g_strdup (name);
    info->base.type = INFO_TYPE_NODE;

    return info;
}

NodeInfo *
node_info_ref (NodeInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
node_info_unref (NodeInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)
        {
            free_interface_list (&info->interfaces);
            free_node_list (&info->nodes);
            base_info_free (info);
        }
}

const char*
node_info_get_name (NodeInfo *info)
{
    return info->base.name;
}

GSList*
node_info_get_interfaces (NodeInfo *info)
{
    return info->interfaces;
}

void
node_info_add_interface (NodeInfo *info,
                        InterfaceInfo *interface)
{
    interface_info_ref (interface);
    info->interfaces = g_slist_append (info->interfaces, interface);
}

GSList*
node_info_get_nodes (NodeInfo *info)
{
    return info->nodes;
}

```

```

void
node_info_add_node (NodeInfo *info,
                   NodeInfo *node)
{
    node_info_ref (node);
    info->nodes = g_slist_append (info->nodes, node);
}

void
node_info_replace_node (NodeInfo          *info,
                       NodeInfo          *old_child,
                       NodeInfo          *new_child)
{
    GSList *link;

    node_info_ref (new_child); /* before unref old_child in case they
are the same */
    link = g_slist_find (info->nodes, old_child);
    g_assert (link != NULL);
    node_info_unref (old_child);
    link->data = new_child;
}

InterfaceInfo*
interface_info_new (const char *name)
{
    InterfaceInfo *info;

    info = g_new0 (InterfaceInfo, 1);
    info->base.refcount = 1;
    info->base.name = g_strdup (name);
    info->base.type = INFO_TYPE_INTERFACE;
    info->annotations = g_hash_table_new_full (g_str_hash, g_str_equal,
                                              (GDestroyNotify) g_free,
                                              (GDestroyNotify) g_free);

    return info;
}

InterfaceInfo *
interface_info_ref (InterfaceInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
interface_info_unref (InterfaceInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)

```

```

    {
        g_hash_table_destroy (info->annotations);
        free_method_list (&info->methods);
        free_signal_list (&info->signals);
        free_property_list (&info->properties);
        base_info_free (info);
    }
}

const char*
interface_info_get_name (InterfaceInfo *info)
{
    return info->base.name;
}

GSList *
interface_info_get_annotations (InterfaceInfo *info)
{
    return get_hash_keys (info->annotations);
}

const char*
interface_info_get_annotation (InterfaceInfo *info,
                               const char *name)
{
    return g_hash_table_lookup (info->annotations, name);
}

GSList*
interface_info_get_methods (InterfaceInfo *info)
{
    return info->methods;
}

GSList*
interface_info_get_signals (InterfaceInfo *info)
{
    return info->signals;
}

GSList*
interface_info_get_properties (InterfaceInfo *info)
{
    return info->properties;
}

void
interface_info_add_annotation (InterfaceInfo *info,
                              const char *name,
                              const char *value)
{
    g_hash_table_insert (info->annotations,

```

```

        g_strdup (name),
        g_strdup (value));
}

void
interface_info_add_method (InterfaceInfo *info,
                           MethodInfo     *method)
{
    method_info_ref (method);
    info->methods = g_slist_append (info->methods, method);
}

void
interface_info_add_signal (InterfaceInfo *info,
                           SignalInfo   *signal)
{
    signal_info_ref (signal);
    info->signals = g_slist_append (info->signals, signal);
}

void
interface_info_add_property (InterfaceInfo *info,
                             PropertyInfo *property)
{
    property_info_ref (property);
    info->properties = g_slist_append (info->properties, property);
}

static void
free_arg_list (GSLIST **args_p)
{
    GSLIST *tmp;
    tmp = *args_p;
    while (tmp != NULL)
    {
        ArgInfo *ai = tmp->data;
        g_assert (ai->base.type == INFO_TYPE_ARG);
        arg_info_unref (tmp->data);
        tmp = tmp->next;
    }
    g_slist_free (*args_p);
    *args_p = NULL;
}

MethodInfo*
method_info_new (const char *name)
{
    MethodInfo *info;

    info = g_new0 (MethodInfo, 1);
    info->base.refcount = 1;
    info->base.name = g_strdup (name);
}

```

```

info->base.type = INFO_TYPE_METHOD;
info->annotations = g_hash_table_new_full (g_str_hash, g_str_equal,
                                           (GDestroyNotify) g_free,
                                           (GDestroyNotify) g_free);

return info;
}

MethodInfo *
method_info_ref (MethodInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
method_info_unref (MethodInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)
        {
            g_hash_table_destroy (info->annotations);
            free_arg_list (&info->args);
            base_info_free (info);
        }
}

const char*
method_info_get_name (MethodInfo *info)
{
    return info->base.name;
}

GSLIST *
method_info_get_annotations (MethodInfo *info)
{
    return get_hash_keys (info->annotations);
}

const char*
method_info_get_annotation (MethodInfo *info,
                           const char *name)
{
    return g_hash_table_lookup (info->annotations, name);
}

GSLIST*
method_info_get_args (MethodInfo *info)
{
    return info->args;
}

```

```

int
method_info_get_n_args (MethodInfo *info)
{
    return g_slist_length (info->args);
}

static int
args_sort_by_direction (const void *a,
                        const void *b)
{
    const ArgInfo *arg_a = a;
    const ArgInfo *arg_b = b;

    if (arg_a->direction == arg_b->direction)
        return 0;
    else if (arg_a->direction == ARG_IN)
        return -1; /* in is less than out */
    else
        return 1;
}

void
method_info_add_annotation (MethodInfo *info,
                            const char *name,
                            const char *value)
{
    g_hash_table_insert (info->annotations,
                        g_strdup (name),
                        g_strdup (value));
}

void
method_info_add_arg (MethodInfo *info,
                    ArgInfo *arg)
{
    arg_info_ref (arg);
    info->args = g_slist_append (info->args, arg);

    /* Keep "in" args sorted before "out" and otherwise maintain
     * stable order (g_slist_sort is stable, at least in sufficiently
     * new glib)
     */
    info->args = g_slist_sort (info->args, args_sort_by_direction);
}

SignalInfo*
signal_info_new (const char *name)
{
    SignalInfo *info;

    info = g_new0 (SignalInfo, 1);
}

```

```

    info->base.refcount = 1;
    info->base.name = g_strdup (name);
    info->base.type = INFO_TYPE_SIGNAL;

    return info;
}

SignalInfo *
signal_info_ref (SignalInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
signal_info_unref (SignalInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)
        {
            free_arg_list (&info->args);
            base_info_free (info);
        }
}

const char*
signal_info_get_name (SignalInfo *info)
{
    return info->base.name;
}

GSList*
signal_info_get_args (SignalInfo *info)
{
    return info->args;
}

int
signal_info_get_n_args (SignalInfo *info)
{
    return g_slist_length (info->args);
}

void
signal_info_add_arg (SignalInfo    *info,
                    ArgInfo       *arg)
{
    g_assert (arg->direction == ARG_OUT);

    arg_info_ref (arg);
    info->args = g_slist_append (info->args, arg);
}

```

```

    /* signal args don't need sorting since only "out" is allowed */
}

PropertyInfo*
property_info_new (const char          *name,
                  const char          *type,
                  PropertyAccessFlags access)
{
    PropertyInfo *info;

    info = g_new0 (PropertyInfo, 1);
    info->base.refcount = 1;
    info->base.name = g_strdup (name);
    info->base.type = INFO_TYPE_PROPERTY;

    info->type = g_strdup (type);
    info->access = access;

    return info;
}

PropertyInfo*
property_info_ref (PropertyInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
property_info_unref (PropertyInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)
    {
        g_free (info->type);
        base_info_free (info);
    }
}

const char*
property_info_get_name (PropertyInfo *info)
{
    return info->base.name;
}

const char *
property_info_get_type (PropertyInfo *info)
{
    return info->type;
}

```



```

PropertyAccessFlags
property_info_get_access (PropertyInfo *info)
{
    return info->access;
}

ArgInfo*
arg_info_new (const char *name,
              ArgDirection direction,
              const char *type)
{
    ArgInfo *info;

    info = g_new0 (ArgInfo, 1);
    info->base.refcount = 1;
    info->base.type = INFO_TYPE_ARG;

    /* name can be NULL */
    info->base.name = g_strdup (name);
    info->direction = direction;
    info->type = g_strdup (type);
    info->annotations = g_hash_table_new_full (g_str_hash, g_str_equal,
                                              (GDestroyNotify) g_free,
                                              (GDestroyNotify) g_free);

    return info;
}

ArgInfo *
arg_info_ref (ArgInfo *info)
{
    info->base.refcount += 1;

    return info;
}

void
arg_info_unref (ArgInfo *info)
{
    info->base.refcount -= 1;
    if (info->base.refcount == 0)
    {
        g_hash_table_destroy (info->annotations);
        g_free (info->type);
        base_info_free (info);
    }
}

const char*
arg_info_get_name (ArgInfo *info)
{

```

```

    return info->base.name;
}

const char *
arg_info_get_type (ArgInfo *info)
{
    return info->type;
}

ArgDirection
arg_info_get_direction (ArgInfo *info)
{
    return info->direction;
}

GSList*
arg_info_get_annotations (ArgInfo *info)
{
    return get_hash_keys (info->annotations);
}

const char*
arg_info_get_annotation (ArgInfo *info,
                        const char *annotation)
{
    return g_hash_table_lookup (info->annotations, annotation);
}

void
arg_info_add_annotation (ArgInfo *info,
                        const char *name,
                        const char *value)
{
    g_hash_table_insert (info->annotations,
                        g_strdup (name),
                        g_strdup (value));
}

File = dbus-gidl.h

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gidl.h data structure describing an interface, to be generated
from IDL
*           or something
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
*/

```

```

* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_GLIB_IDL_H
#define DBUS_GLIB_IDL_H

#include <dbus/dbus.h>
#include <glib-object.h>

G_BEGIN_DECLS

typedef struct BaseInfo      BaseInfo;
typedef struct NodeInfo     NodeInfo;
typedef struct InterfaceInfo InterfaceInfo;
typedef struct MethodInfo   MethodInfo;
typedef struct SignalInfo   SignalInfo;
typedef struct PropertyInfo PropertyInfo;
typedef struct ArgInfo      ArgInfo;

typedef enum
{
    ARG_INVALID = -1,
    ARG_IN,
    ARG_OUT
} ArgDirection;

typedef enum
{
    PROPERTY_READ  = 1 << 0,
    PROPERTY_WRITE = 1 << 1
} PropertyAccessFlags;

typedef enum
{
    INFO_TYPE_NODE,
    INFO_TYPE_INTERFACE,
    INFO_TYPE_METHOD,

```

```

    INFO_TYPE_SIGNAL,
    INFO_TYPE_ARG,
    INFO_TYPE_PROPERTY
} InfoType;

BaseInfo*      base_info_ref      (BaseInfo      *info);
void           base_info_unref    (BaseInfo      *info);
InfoType       base_info_get_type (BaseInfo      *info);
const char*    base_info_get_name (BaseInfo      *info);
void           base_info_set_name (BaseInfo      *info,
                                   const char      *name);

GType          base_info_get_gtype (void);
#define        BASE_INFO_TYPE      (base_info_get_gtype ())

NodeInfo*      node_info_new      (const char
                                   *name);
NodeInfo*      node_info_ref      (NodeInfo
                                   *info);
void           node_info_unref    (NodeInfo
                                   *info);
const char*    node_info_get_name (NodeInfo
                                   *info);
GSLList*       node_info_get_interfaces (NodeInfo
                                   *info);
GSLList*       node_info_get_nodes   (NodeInfo
                                   *info);
void           node_info_add_interface (NodeInfo
                                   *info,
                                   InterfaceInfo
                                   *iface);
void           node_info_add_node    (NodeInfo
                                   *info,
                                   NodeInfo
                                   *child);
void           node_info_replace_node (NodeInfo
                                   *info,
                                   NodeInfo
                                   *old_child,
                                   NodeInfo
                                   *new_child);
InterfaceInfo* interface_info_new  (const char
                                   *name);
InterfaceInfo* interface_info_ref  (InterfaceInfo
                                   *info);
void           interface_info_unref (InterfaceInfo
                                   *info);
const char*    interface_info_get_name (InterfaceInfo
                                   *info);
GSLList*       interface_info_get_annotations (InterfaceInfo
                                   *info);

```

```

const char*          interface_info_get_annotation (InterfaceInfo*info,
                                                    const char
                                                    *annotation);
GSLList*            interface_info_get_methods   (InterfaceInfo
*info);
GSLList*            interface_info_get_signals   (InterfaceInfo
*info);
GSLList*            interface_info_get_properties (InterfaceInfo
*info);
void                interface_info_add_annotation (InterfaceInfo
*info,
                                                    const char
                                                    *name,
                                                    const char
                                                    *value);
void                interface_info_add_method   (InterfaceInfo
*info,
                                                    MethodInfo
*method);
void                interface_info_add_signal   (InterfaceInfo
*info,
                                                    SignalInfo
*signal);
void                interface_info_add_property (InterfaceInfo
*info,
                                                    PropertyInfo
*property);
MethodInfo*         method_info_new            (const char
*name);
MethodInfo*         method_info_ref           (MethodInfo
*info);
void                method_info_unref         (MethodInfo
*info);
const char*         method_info_get_name      (MethodInfo
*info);
GSLList*            method_info_get_annotations (MethodInfo
*info);
const char*         method_info_get_annotation (MethodInfo
*info,
                                                    const char
                                                    *annotation);
void                method_info_add_annotation (MethodInfo
*info,
                                                    const char
                                                    *name,
                                                    const char
                                                    *value);
GSLList*            method_info_get_args      (MethodInfo
*info);
void                method_info_add_arg       (MethodInfo
*info,
                                                    ArgInfo
*arg);
int                 method_info_get_n_args    (MethodInfo
*info);
SignalInfo*         signal_info_new          (const char
*name);

```



```
void          arg_info_add_annotation      (ArgInfo
*info,
                                         const char      *name,
                                         const char      *value);
```

```
G_END_DECLS
```

```
#endif /* DBUS_GLIB_IDL_H */
```

```
File = dbus-glib-1-uninstalled.pc
```

```
prefix=/usr
exec_prefix=/usr
libdir=/usr/lib
includedir=/usr/include
```

```
Name: dbus-glib
Description: GLib integration for the free desktop message bus
Version: 0.100.2
Requires: dbus-1 glib-2.0 gobject-2.0
Libs: ${pc_top_builddir}/${pcfiledir}/dbus/libdbus-glib-1.la
Cflags: -I${pc_top_builddir}/${pcfiledir}
```

```
File = dbus-glib-1-uninstalled.pc.in
```

```
prefix=@prefix@
exec_prefix=@exec_prefix@
libdir=@libdir@
includedir=@includedir@
```

```
Name: dbus-glib
Description: GLib integration for the free desktop message bus
Version: @VERSION@
Requires: dbus-1 glib-2.0 gobject-2.0
Libs: ${pc_top_builddir}/${pcfiledir}/dbus/libdbus-glib-1.la
Cflags: -I${pc_top_builddir}/${pcfiledir}
```

```
File = dbus-glib-1.pc
```

```
prefix=/usr
exec_prefix=/usr
libdir=/usr/lib
includedir=/usr/include
```

Name: dbus-glib
Description: GLib integration for the free desktop message bus
Version: 0.100.2
Requires: dbus-1 glib-2.0 gobject-2.0
Libs: -L\${libdir} -ldbus-glib-1
Cflags: -I\${includedir}/dbus-1.0

File = dbus-glib-1.pc.in

```
prefix=@prefix@  
exec_prefix=@exec_prefix@  
libdir=@libdir@  
includedir=@includedir@
```

Name: dbus-glib
Description: GLib integration for the free desktop message bus
Version: @VERSION@
Requires: dbus-1 glib-2.0 gobject-2.0
Libs: -L\${libdir} -ldbus-glib-1
Cflags: -I\${includedir}/dbus-1.0

File = dbus-glib-DBus-GLib-low-level.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>DBus GLib low level</title>  
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">  
<link rel="home" href="index.html" title="D-Bus GLib bindings -  
Reference Manual">  
<link rel="up" href="ch02.html" title="API Reference">  
<link rel="prev" href="dbus-glib-Specializable-GType-System.html"  
title="Specializable GType System">  
<link rel="next" href="ch03.html" title="Tools Reference">  
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">  
<link rel="stylesheet" href="style.css" type="text/css">  
</head>  
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"  
alink="#0000FF">  
<table class="navigation" id="top" width="100%" summary="Navigation  
header" cellpadding="2" cellspacing="2">  
<tr valign="middle">
```



```

<td><a accesskey="p" href="dbus-glib-Specializable-GType-System.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference Manual</th>
<td><a accesskey="n" href="ch03.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBus-GLib-low-level.synopsis" class="shortcut">Top</a>
      Â |Â
      <a href="#dbus-glib-DBus-GLib-low-level.description" class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBus-GLib-low-level"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-DBus-GLib-low-level.top_of_page"></a>DBus GLib low level</span></h2>
<p>DBus GLib low level â€” DBus lower level functions</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBus-GLib-low-level.stability-level"></a><h2>Stability Level</h2>
Unstable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBus-GLib-low-level.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib-lowlevel.h&gt;

<span class="returnvalue">void</span>                               <a class="link" href="dbus-glib-DBus-GLib-low-level.html#dbus-set-g-error" title="dbus_set_g_error ()">dbus_set_g_error</a>
(<em class="parameter"><code><a href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>
**gerror</code></em>,
                                     <em
class="parameter"><code><span class="type">DBusError</span></code>
*derror</code></em>);

```

```

<span class="returnvalue">void</span>                <a class="link"
href="dbus-glib-DBus-GLib-low-level.html#dbus-connection-setup-with-g-
main" title="dbus_connection_setup_with_g_main
()">dbus_connection_setup_with_g_main</a>    (<em
class="parameter"><code><span class="type">DBusConnection</span>
*connection</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-
Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>);
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> *    <a class="link"
href="dbus-glib-DBus-GLib-low-level.html#dbus-connection-get-g-
connection" title="dbus_connection_get_g_connection
()">dbus_connection_get_g_connection</a>    (<em
class="parameter"><code><span class="type">DBusConnection</span>
*connection</code></em>);
<span class="returnvalue">void</span>                <a class="link"
href="dbus-glib-DBus-GLib-low-level.html#dbus-server-setup-with-g-
main" title="dbus_server_setup_with_g_main
()">dbus_server_setup_with_g_main</a>    (<em
class="parameter"><code><span class="type">DBusServer</span>
*server</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-
Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>);
#define                <a class="link" href="dbus-glib-DBus-GLib-low-
level.html#DBUS-TYPE-CONNECTION:CAPS"
title="DBUS_TYPE_CONNECTION">DBUS_TYPE_CONNECTION</a>
#define                <a class="link" href="dbus-glib-DBus-GLib-low-
level.html#DBUS-TYPE-MESSAGE:CAPS"
title="DBUS_TYPE_MESSAGE">DBUS_TYPE_MESSAGE</a>
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-DBus-GLib-low-
level.description"></a><h2>Description</h2>
<p>
These functions can be used to access lower level of Dbus.
</p>
</div>
<div class="refsect1">
<a name="dbus-glib-DBus-GLib-low-level.details"></a><h2>Details</h2>
<div class="refsect2">
<a name="dbus-set-g-error"></a><h3>dbus_set_g_error ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_set_g_error                (<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-

```

```
Reporting.html#GError"><span class="type">GError</span></a>  
**gerror</code></em>,</pre>
```

```
<em  
class="parameter"><code><span class="type">DBusError</span>  
*derror</code></em>);</pre>
```

Store the information from a Dbus method error return into a GError. For the normal case of an arbitrary remote process, the error code will be DBUS_GERROR_REMOTE_EXCEPTION. Now, Dbus errors have two components; a message and a "name". The former is an arbitrary (normally American English) string. The second is a string like com.example.FooFailure which programs can use as a conditional source. Because a GError only has one string, we use a hack to encode both values:

```
</p>  
&lt;human readable string&gt;&lt;null&gt;&lt;error  
name&gt;&lt;null&gt;
```

You can use the following code to retrieve both values:

```
</p>  
<div class="informalexample">  
  <table class="listing_frame" border="0" cellpadding="0"  
    cellspacing="0">  
    <tbody>  
      <tr>  
        <td class="listing_lines" align="right"><pre>1  
2  
3</pre></td>  
        <td class="listing_code"><pre class="programlisting"><span  
class="gtkdoc kwb">const char</span> <span class="gtkdoc  
opt">*</span>msg <span class="gtkdoc opt">=</span> error<span  
class="gtkdoc opt">-&gt;</span>message<span class="gtkdoc  
opt">;</span>  
<span class="gtkdoc kwb">size_t</span> len <span class="gtkdoc  
opt">=</span> <span class="function">strlen</span><span class="gtkdoc  
opt">(</span>msg<span class="gtkdoc opt">)</span>;</span>  
<span class="gtkdoc kwb">const char</span> <span class="gtkdoc  
opt">*</span>error_name <span class="gtkdoc opt">=</span> msg<span  
class="gtkdoc opt">+</span>len<span class="gtkdoc opt">+</span><span  
class="number">1</span><span class="gtkdoc opt">;</span></pre></td>  
      </tr>  
    </tbody>  
  </table>  
</div>
```

```
<p>  
</p>  
<div class="variablelist"><table border="0">
```

```

<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gerror</code></em>Â :</span></p></td>
<td>an error</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>derror</code></em>Â :</span></p></td>
<td>a <span class="type">DBusError</span>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-connection-setup-with-g-
main"></a><h3>dbus_connection_setup_with_g_main ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_connection_setup_with_g_main (<em class="parameter"><code><span
class="type">DBusConnection</span> *connection</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-
Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>);</pre>
<p>
Sets the watch and timeout functions of a <span
class="type">DBusConnection</span>
to integrate the connection with the GLib main loop.
Pass in <a href="http://library.gnome.org/devel/glib/unstable/glib-
Standard-Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>
for the <a href="http://library.gnome.org/devel/glib/unstable/glib-
The-Main-Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> unless you're
doing something specialized.
</p>
<p>
If called twice for the same context, does nothing the second
time. If called once with context A and once with context B,
context B replaces context A as the context monitoring the
connection.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the connection</td>

```

```

</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em>Â :</span></p></td>
<td>the <a href="http://library.gnome.org/devel/glib/unstable/glib-
The-Main-Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> or <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#NULL:CAPS"><code class="literal">NULL</code></a> for
default context</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-connection-get-g-
connection"></a><h3>dbus_connection_get_g_connection ()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> *
dbus_connection_get_g_connection (<em class="parameter"><code><span
class="type">DBusConnection</span> *connection</code></em>);</pre>
<p>
Get the <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> corresponding to this <span
class="type">DBusConnection</span>. This only
makes sense if the <span class="type">DBusConnection</span> was
originally a <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> that was
registered with the GLib main loop. The return value does not have
its
refcount incremented.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>a <span class="type">DBusConnection</span>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a></td>
</tr>

```

```

</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-server-setup-with-g-
main"></a><h3>dbus_server_setup_with_g_main ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_server_setup_with_g_main      (<em class="parameter"><code><span
class="type">DBusServer</span> *server</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-
Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>);</pre>
<p>
Sets the watch and timeout functions of a <span
class="type">DBusServer</span>
to integrate the server with the GLib main loop.
In most cases the context argument should be <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>.
</p>
<p>
If called twice for the same context, does nothing the second
time. If called once with context A and once with context B,
context B replaces context A as the context monitoring the
connection.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>server</code></em>Â :</span></p></td>
<td>the server</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em>Â :</span></p></td>
<td>the <a href="http://library.gnome.org/devel/glib/unstable/glib-
The-Main-Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> or <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#NULL:CAPS"><code class="literal">NULL</code></a> for
default</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">

```

<h3>DBUS_TYPE_CONNECTION</h3>

```
class="programlisting">#define DBUS_TYPE_CONNECTION
(dbus_connection_get_g_type ())
</pre>
```

<p>

Expands to a function call returning a boxed GType representing a

DBusConnection pointer from libdbus. Not to be confused with

<code class="literal">DBUS_TYPE_G_CONNECTION</code>, which you should usually use instead.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody><tr>

<td><p>ReturnsÂ :</p></td>

<td>the GLib type</td>

</tr></tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<h3>DBUS_TYPE_MESSAGE</h3>

```
class="programlisting">#define DBUS_TYPE_MESSAGE
(dbus_message_get_g_type ())
</pre>
```

<p>

Expands to a function call returning a boxed GType representing a

DBusMessage pointer from libdbus. Not to be confused with

<code class="literal">DBUS_TYPE_G_MESSAGE</code>, which you should usually use instead.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody><tr>

<td><p>ReturnsÂ :</p></td>

<td>the GLib type</td>

</tr></tbody>

</table></div>

</div>

```

</div>
</div>
<div class="footer">
<hr>
        Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBus-GObject-related-functions.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBus GObject related functions</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBusGConnection.html"
title="DBusGConnection">
<link rel="next" href="dbus-glib-DBusGMessage.html"
title="DBusGMessage">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBusGConnection.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBusGMessage.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBus-GObject-related-functions.synopsis"
class="shortcut">Top</a>
        Â |Â
        <a href="#dbus-glib-DBus-GObject-related-
functions.description" class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">

```



```

<a name="dbus-glib-DBus-GObject-related-functions"></a><div
class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-DBus-GObject-
related-functions.top_of_page"></a>DBus GObject related
functions</span></h2>
<p>DBus GObject related functions â€” Exporting a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
remotely</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBus-GObject-related-functions.stability-
level"></a><h2>Stability Level</h2>
Stable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBus-GObject-related-
functions.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h&gt;

struct          <a class="link" href="dbus-glib-DBus-GObject-
related-functions.html#DBusGObjectInfo" title="struct
DBusGObjectInfo">DBusGObjectInfo</a>;
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
type-install-info" title="dbus_g_object_type_install_info
()">dbus_g_object_type_install_info</a>          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
object_type</code></em>,
<em
class="parameter"><code>const <a class="link" href="dbus-glib-DBus-
GObject-related-functions.html#DBusGObjectInfo" title="struct
DBusGObjectInfo"><span class="type">DBusGObjectInfo</span></a>
*info</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
type-register-shadow-property"
title="dbus_g_object_type_register_shadow_property
()">dbus_g_object_type_register_shadow_property</a>
          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
iface_type</code></em>,

```

```

class="parameter"><code>const <span class="type">char</span>
*dbus_prop_name</code></em>,
class="parameter"><code>const <span class="type">char</span>
*shadow_prop_name</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-DBus-GObject-related-
functions.html#dbus-g-object-path-get-g-type"
title="dbus_g_object_path_get_g_type
()">dbus_g_object_path_get_g_type</a> (<em
class="parameter"><code><span class="type">void</span></code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
register-marshaller" title="dbus_g_object_register_marshall
er()">dbus_g_object_register_marshall
er</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureMarshal"><span
class="type">GClosureMarshal</span></a> marshaller</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
rettype</code></em>,
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
register-marshaller-array"
title="dbus_g_object_register_marshall
er_array()">dbus_g_object_register_marshall
er_array</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureMarshal"><span
class="type">GClosureMarshal</span></a> marshaller</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
rettype</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
n_types</code></em>,
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-

```

```

Information.html#GType"><span class="type">GType</span></a>
*types</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-glib-global-
set-disable-legacy-property-access"
title="dbus_glib_global_set_disable_legacy_property_access
()">dbus_glib_global_set_disable_legacy_property_access</a>
          (<em
class="parameter"><code><span class="type">void</span></code></em>);
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-DBus-GObject-related-
functions.description"></a><h2>Description</h2>
<p>
FIXME
</p>
</div>
<div class="refsect1">
<a name="dbus-glib-DBus-GObject-related-
functions.details"></a><h2>Details</h2>
<div class="refsect2">
<a name="DBusGObjectInfo"></a><h3>struct DBusGObjectInfo</h3>
<pre class="programlisting">struct DBusGObjectInfo {
    int    format_version;

    const DBusGMethodInfo *method_infos;
    int    n_method_infos;
    const char *data;
    const char *exported_signals;
    const char *exported_properties;
};
</pre>
<p>
Introspection data for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>,
normally autogenerated by
a tool such as <a class="link" href="dbus-binding-tool.html"
title="dbus-binding-tool"><span class="type">dbus-binding-
tool</span></a>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>Ã <em
class="structfield"><code><a name="DBusGObjectInfo.format-
version"></a>format_version</code></em>;</span></p></td>
<td>Allows us to change the rest of this struct

```

```

by adding DBusGObjectInfo2, DBusGObjectInfo3, etc.</td>
</tr>
<tr>
<td><p><span class="term">constÂ <a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInfo" title="struct DBusGMethodInfo"><span
class="type">DBusGMethodInfo</span></a>Â *<em
class="structfield"><code><a name="DBusGObjectInfo.method-
infos"></a>method_infos</code></em>;</span></p></td>
<td>Array of method pointers</td>
</tr>
<tr>
<td><p><span class="term"><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>Â <em
class="structfield"><code><a name="DBusGObjectInfo.n-method-
infos"></a>n_method_infos</code></em>;</span></p></td>
<td>Length of the infos array</td>
</tr>
<tr>
<td><p><span class="term">constÂ <span class="type">char</span>Â *<em
class="structfield"><code><a name="DBusGObjectInfo.data"></a>data</code></em>;</span></p></td>
<td>Introspection data</td>
</tr>
<tr>
<td><p><span class="term">constÂ <span class="type">char</span>Â *<em
class="structfield"><code><a name="DBusGObjectInfo.exported-
signals"></a>exported_signals</code></em>;</span></p></td>
<td>Exported signals</td>
</tr>
<tr>
<td><p><span class="term">constÂ <span class="type">char</span>Â *<em
class="structfield"><code><a name="DBusGObjectInfo.exported-
properties"></a>exported_properties</code></em>;</span></p></td>
<td>Exported properties</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-object-type-install-
info"></a><h3>dbus_g_object_type_install_info ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_object_type_install_info      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
object_type</code></em>,
                                     <em
class="parameter"><code>const <a class="link" href="dbus-glib-DBus-
GObject-related-functions.html#DBusGObjectInfo" title="struct

```

```

DBusGObjectInfo"><span class="type">DBusGObjectInfo</span></a>
*info</code></em>);</pre>
<p>
Install introspection information about the given object <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
sufficient to allow methods on the object to be invoked by name.
The introspection information is normally generated by
dbus-glib-tool, then this function is called in the
<code class="function">class_init()</code> for the object class.
</p>
<p>
Once introspection information has been installed, instances of the
object registered with <a class="link" href="dbus-glib-
DBusGConnection.html#dbus-g-connection-register-g-object"
title="dbus_g_connection_register_g_object ()"><code
class="function">dbus_g_connection_register_g_object()</code></a> can
have
their methods invoked remotely.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>object_type</code></em>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for the
object</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>info</code></em>Â :</span></p></td>
<td>introspection data generated by <span class="type">dbus-glib-
tool</span>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-object-type-register-shadow-
property"></a><h3>dbus_g_object_type_register_shadow_property ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_object_type_register_shadow_property
(
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
iface_type</code></em>,

```

```
class="parameter"><code>const <span class="type">char</span>
*dbus_prop_name</code></em>,</pre>
```

```
class="parameter"><code>const <span class="type">char</span>
*shadow_prop_name</code></em>);</pre>
```

<p>

Registers a new property name <em

class="parameter"><code>shadow_prop_name</code> that overrides
the

<em class="parameter"><code>dbus_prop_name</code> in D-Bus

property get/set requests. Since all properties for

all interfaces implemented by a GObject exist in the same namespace,
this

allows implementations to use the same property name in two or more D-
Bus

interfaces implemented by the same GObject, as long as one of those D-
Bus

interface properties is registered with a shadow property name.

</p>

<p>

For example, if both org.foobar.Baz.InterfaceA and

org.foobar.Baz.InterfaceB

have a D-Bus property called "Bork", the developer assigns a shadow
property

name to the conflicting property name in one or both of these
GInterfaces to

resolve the conflict. Assume the GInterface implementing

org.foobar.Baz.InterfaceA registers a shadow property called "a-bork",
while

the GInterface implementing org.foobar.Baz.InterfaceB registers a
shadow

property called "b-bork". The GObject implementing both these
GInterfaces

would then use #<a

href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#g-object-class-override-property"><code

class="function">g_object_class_override_property()</code> to
implement both

"a-bork" and "b-bork" and D-Bus requests for "Bork" on either D-Bus
interface

will not conflict.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>iface_type</code>Â :</p></td>

<td>

```

<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for the
<span class="type">GInterface</span>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>dbus_prop_name</code></em>Â :</span></p></td>
<td>D-Bus property name (as specified in the introspection data)
to override with the shadow property name (as specified in the GType's
initialization function, ie glib-style)</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>shadow_prop_name</code></em>Â :</span></p></td>
<td>property name which should override the shadow property</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-object-path-get-g-
type"></a><h3>dbus_g_object_path_get_g_type ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_object_path_get_g_type      (<em class="parameter"><code><span
class="type">void</span></code></em>);</pre>
<p>
</p>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-object-register-
marshaller"></a><h3>dbus_g_object_register_marshalller ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_object_register_marshalller  (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureMarshal"><span
class="type">GClosureMarshal</span></a> marshaller</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
rettype</code></em>,
<em
class="parameter"><code>...</code></em>);</pre>
<p>
Register a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-

```

Closures.html#GClosureMarshal">GClosureMarshal to be used for signal invocations, giving its return type and a list of parameter types, followed by <code class="literal">G_TYPE_INVALID</code>.

</p>

<p>This function will not be needed once GLib includes libffi.</p>

<http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType>>GType
rettype</code>,</p></div>

class="parameter"><code>guint
n_types</code>,</p></div>

class="parameter"><code>const GType
*types</code>);</pre></div>

<p>

Register a <span
class="type">GClosureMarshal to be used for signal
invocations.</p></div>

<em class="parameter"><code>see_also</code> <a class="link"
href="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
register-marshaller" title="dbus_g_object_register_marshall
()"><code>

class="function">dbus_g_object_register_marshall()</code>
</p></div>

<div class="variablelist"><table border="0">
<col align="left" valign="top">

<tbody>

<tr>

<td><p><em
class="parameter"><code>marshaller</code>Â :</p></td>

<td>a <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureMarshal"><span
class="type">GClosureMarshal to be used for invocation</td>
</tr>

<tr>

<td><p><em
class="parameter"><code>rettype</code>Â :</p></td>

<td>a <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType">GType for the
return type of the function</td>

</tr>

<tr>

<td><p><em
class="parameter"><code>n_types</code>Â :</p></td>

<td>number of function parameters</td>

</tr>

<tr>

<td><p><em
class="parameter"><code>types</code>Â :</p></td>

```

<td>a C array of GTypes values</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-glib-global-set-disable-legacy-property-
access"></a><h3>dbus_glib_global_set_disable_legacy_property_access
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_glib_global_set_disable_legacy_property_access
                                (<em
class="parameter"><code><span
class="type">void</span></code></em>);</pre>
<p>
For historical reasons, DBus-GLib will allow read-only
access to every GObject property of an object exported
to the bus, regardless of whether or not the property
is listed in the type info installed with
<a class="link" href="dbus-glib-DBus-GObject-related-
functions.html#dbus-g-object-type-install-info"
title="dbus_g_object_type_install_info ()"><code
class="function">dbus_g_object_type_install_info()</code></a>. (Write
access is
denied however).
</p>
<p>
If you wish to restrict even read-only access, you
can call this method to globally change the behavior
for the entire process.
</p>
<p class="since">Since 0.88</p>
</div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBus-GObject-related-functions.see-
also"></a><h2>See Also</h2>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBusGConnection.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBusGConnection</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="ch02.html" title="API Reference">
<link rel="next" href="dbus-glib-DBus-GObject-related-functions.html"
title="DBus GObject related functions">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="ch02.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBus-GObject-related-
functions.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBusGConnection.synopsis" class="shortcut">Top</a>
      Â |Â
      <a href="#dbus-glib-DBusGConnection.description"
class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBusGConnection"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-
DBusGConnection.top_of_page"></a>DBusGConnection</span></h2>
<p>DBusGConnection â€” D-Bus Connection</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBusGConnection.stability-level"></a><h2>Stability
Level</h2>
Stable, unless otherwise indicated

```

```

</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBusGConnection.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h&gt;

        <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection"
title="DBusGConnection">DBusGConnection</a>;
#define          <a class="link" href="dbus-glib-
DBusGConnection.html#DBUS-TYPE-G-CONNECTION:CAPS"
title="DBUS_TYPE_G_CONNECTION">DBUS_TYPE_G_CONNECTION</a>
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-bus-get"
title="dbus_g_bus_get ()">dbus_g_bus_get</a>          (<em
class="parameter"><code><span class="type">DBusBusType</span>
type</code></em>,

                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-bus-get-private"
title="dbus_g_bus_get_private ()">dbus_g_bus_get_private</a>
(<em class="parameter"><code><span class="type">DBusBusType</span>
type</code></em>,

                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-
Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>,

                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-thread-init"
title="dbus_g_thread_init ()">dbus_g_thread_init</a>
(<em class="parameter"><code><span
class="type">void</span></code></em>);
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-open"
title="dbus_g_connection_open ()">dbus_g_connection_open</a>
(<em class="parameter"><code>const <a

```

```

href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gchar"><span class="type">gchar</span></a>
*address</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-ref"
title="dbus_g_connection_ref ()">dbus_g_connection_ref</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-unref"
title="dbus_g_connection_unref ()">dbus_g_connection_unref</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-flush"
title="dbus_g_connection_flush ()">dbus_g_connection_flush</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>);
<span class="returnvalue">DBusConnection</span> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-get-connection"
title="dbus_g_connection_get_connection
() ">dbus_g_connection_get_connection</a> (<em
class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *gconnection</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-register-g-
object" title="dbus_g_connection_register_g_object
() ">dbus_g_connection_register_g_object</a> (<em
class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*at_path</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
*object</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-unregister-g-

```

```

object" title="dbus_g_connection_unregister_g_object
()">dbus_g_connection_unregister_g_object</a>
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
*object</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span
class="returnvalue">GObject</span></a> * <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-lookup-g-
object" title="dbus_g_connection_lookup_g_object
()">dbus_g_connection_lookup_g_object</a> <em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*at_path</code></em>;
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-
DBusGConnection.description"></a><h2>Description</h2>
<p>
A <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> is a boxed type abstracting a
DBusConnection.
</p>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGConnection.details"></a><h2>Details</h2>
<div class="refsect2">
<a name="DBusGConnection"></a><h3>DBusGConnection</h3>
<pre class="programlisting">typedef struct _DBusGConnection
DBusGConnection;</pre>
<p>
A <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> is a boxed type abstracting a
DBusConnection from
libdbus.
</p>
</div>
<hr>
<div class="refsect2">

```

```

<a name="DBUS-TYPE-G-CONNECTION:CAPS"></a><h3>DBUS_TYPE_G_CONNECTION</h3>
<pre class="programlisting">#define DBUS_TYPE_G_CONNECTION
(dbus_g_connection_get_g_type ())
</pre>
<p>
Expands to a function call returning the boxed <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of a <a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>&nbsp;:</span></p></td>
<td>the GLib type</td>
</tr></tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-bus-get"></a><h3>dbus_g_bus_get ()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> * dbus_g_bus_get
(<em class="parameter"><code><span class="type">DBusBusType</span>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);</pre>
<p>
Returns a connection to the given bus. The connection is a global
variable
shared with other callers of this function.
</p>
<p>
(Internally, calls <code class="function">dbus_bus_get()</code> then
calls
<a class="link" href="dbus-glib-DBus-GLib-low-level.html#dbus-
connection-setup-with-g-main" title="dbus_connection_setup_with_g_main
() "><code
class="function">dbus_connection_setup_with_g_main()</code></a> on the
result.)
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>

```

<p> class="parameter"><code>type</code>Â : </p>	<p> bus type </p>
<p> class="parameter"><code>error</code>Â : </p>	<p> address where an error can be returned. </p>
<p> class="emphasis">ReturnsÂ : </p>	<p> DBusConnection </p>

[dbus-g-bus-get-private](#)

dbus_g_bus_get_private()

```

class="programlisting"><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span class="returnvalue">DBusGConnection</span></a> *
dbus_g_bus_get_private          (<em class="parameter"><code><span class="type">DBusBusType</span> type</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-The-Main-Event-Loop.html#GMainContext"><span
class="type">GMainContext</span></a> *context</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);</pre>

```

Returns a connection to the given bus. The connection will be a private non-shared connection and should be closed when usage is complete.

Internally this function calls `dbus_bus_get_private()` then calls [dbus_connection_setup_with_g_main\(\)](#) on the result; see the documentation of the former function for more information on private connections.


```

<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em>Â :</span></p></td>
<td>bus type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em>Â :</span></p></td>
<td>Mainloop context to attach to</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>error</code></em>Â :</span></p></td>
<td>address where an error can be returned.</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>a DBusConnection</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-thread-init"></a><h3>dbus_g_thread_init ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_thread_init
      (<em class="parameter"><code><span
class="type">void</span></code></em>);</pre>
<p>
Initializes the D-BUS thread system.
This function may only be called
once and must be called prior to calling any
other function in the D-BUS API.
</p>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-open"></a><h3>dbus_g_connection_open
()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> *
dbus_g_connection_open
      (<em class="parameter"><code>const
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gchar"><span class="type">gchar</span></a>
*address</code></em>,

```

<em

```

class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-

```

```

Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);</pre>
<p>
Returns a connection to the given address.
</p>
<p>
(Internally, calls <code
class="function">dbus_connection_open()</code> then calls
<a class="link" href="dbus-glib-DBus-GLib-low-level.html#dbus-
connection-setup-with-g-main" title="dbus_connection_setup_with_g_main
() "><code
class="function">dbus_connection_setup_with_g_main()</code></a> on the
result.)
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>address</code></em>Â :</span></p></td>
<td>address of the connection to open</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>error</code></em>Â :</span></p></td>
<td>address where an error can be returned.</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>a DBusConnection</td>
</tr>
</tbody>
</table></div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-ref"></a><h3>dbus_g_connection_ref ()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="returnvalue">DBusGConnection</span></a> *
dbus_g_connection_ref
      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>);</pre>
<p>
Increment refcount on a <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">

```

```

<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> to ref</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the connection that was ref'd</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-unref"></a><h3>dbus_g_connection_unref
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_connection_unref          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>);</pre>
<p>
Decrement refcount on a <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the connection to unref</td>
</tr></tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-flush"></a><h3>dbus_g_connection_flush
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_connection_flush          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>);</pre>
<p>
Blocks until outgoing calls and signal emissions have been sent.
</p>
<div class="variablelist"><table border="0">

```

```

<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> to flush</td>
</tr></tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-get-
connection"></a><h3>dbus_g_connection_get_connection ()</h3>
<pre class="programlisting"><span
class="returnvalue">DBusConnection</span> *
dbus_g_connection_get_connection (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*gconnection</code></em>);</pre>
<p>
Get the <span class="type">DBusConnection</span> corresponding to this
<a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>.
The return value does not have its refcount incremented.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gconnection</code></em>Â :</span></p></td>
<td>a <a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td><span class="type">DBusConnection</span></td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-register-g-
object"></a><h3>dbus_g_connection_register_g_object ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_connection_register_g_object (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"

```

```
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>,
```

```
                                <em>
class="parameter"><code>const <span class="type">char</span>
*at_path</code></em>,&#x27E9;
```

```
                                <em>
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
*object</code></em>);</pre>
```

```
<p>
Registers a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
at the given path. Properties, methods, and signals
of the object can then be accessed remotely. Methods are only
available
if method introspection data has been added to the object's class
with <a class="link" href="dbus-glib-DBus-GObject-related-
functions.html#dbus-g-object-type-install-info"
title="dbus_g_object_type_install_info ()"><code
class="function">dbus_g_object_type_install_info()</code></a>.
</p>
```

```
<p>
The registration will be cancelled if either the <span
class="type">DBusConnection</span> or
the <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
The-Base-Object-Type.html#GObject"><span
class="type">GObject</span></a> gets finalized, or if <a class="link"
href="dbus-glib-DBusGConnection.html#dbus-g-connection-unregister-g-
object" title="dbus_g_connection_unregister_g_object ()"><code
class="function">dbus_g_connection_unregister_g_object()</code></a>
is used.
</p>
```

```
<p>
Note: If an object is registered multiple times, the first
registration
takes priority for cases such as turning an object into an object
path.
</p>
```

```
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the D-BUS connection</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>at_path</code></em>Â :</span></p></td>
<td>the path where the object will live (the object's name)</td>
```

```

</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>object</code></em>Â :</span></p></td>
<td>the object</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-unregister-g-
object"></a><h3>dbus_g_connection_unregister_g_object ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_connection_unregister_g_object
                                (<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span class="type">GObject</span></a>
*object</code></em>);</pre>
<p>
Removes <em class="parameter"><code>object</code></em> from any object
paths at which it is exported on
<em class="parameter"><code>connection</code></em>. Properties,
methods, and signals
of the object can no longer be accessed remotely.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the D-BUS connection</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>object</code></em>Â :</span></p></td>
<td>the object</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-connection-lookup-g-
object"></a><h3>dbus_g_connection_lookup_g_object ()</h3>

```

```

<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-The-
Base-Object-Type.html#GObject"><span
class="returnvalue">GObject</span></a> *
dbus_g_connection_lookup_g_object    (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*at_path</code></em>);</pre>
<p>
FIXME
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>at_path</code></em>Â :</span></p></td>
<td>path</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the object at path <em class="parameter"><code>at_path</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGConnection.see-also"></a><h2>See Also</h2>
<span class="type">DBusConnection</span>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBusGError.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBusGError</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBusGMethod.html"
title="DBusGMethod">
<link rel="next" href="dbus-glib-DBusGProxy.html" title="DBusGProxy">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBusGMethod.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBusGProxy.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBusGError.synopsis" class="shortcut">Top</a>
      Â |Â
      <a href="#dbus-glib-DBusGError.description"
class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBusGError"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-
DBusGError.top_of_page"></a>DBusGError</span></h2>
<p>DBusGError â€” DBus GError</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
```



```

<a name="dbus-glib-DBusGError.stability-level"></a><h2>Stability
Level</h2>
Stable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBusGError.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h&gt;

enum          <a class="link" href="dbus-glib-
DBusGError.html#DBusGError" title="enum DBusGError">DBusGError</a>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-DBusGError.html#dbus-g-error-has-name"
title="dbus_g_error_has_name ()">dbus_g_error_has_name</a>
(<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
*error</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*name</code></em>);
const <span class="returnvalue">char</span> *          <a class="link"
href="dbus-glib-DBusGError.html#dbus-g-error-get-name"
title="dbus_g_error_get_name ()">dbus_g_error_get_name</a>
(<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
*error</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBusGError.html#dbus-g-error-domain-register"
title="dbus_g_error_domain_register
()>dbus_g_error_domain_register</a>          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Quarks.html#GQuark"><span class="type">GQuark</span></a>
domain</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*default_iface</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
code_enum</code></em>);
#define          <a class="link" href="dbus-glib-
DBusGError.html#DBUS-GERROR:CAPS" title="DBUS_GERROR">DBUS_GERROR</a>
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGError.description"></a><h2>Description</h2>
<p>

```

<dbus-glib-DBusGError.html#DBusGError> DBusGError is the <http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError> GError used by DBus.

</p>

</div>

<div class="refsect1">

[dbus-glib-DBusGError.details](#) <h2>Details</h2>

<div class="refsect2">

[DBusGError](#) <h3>enum DBusGError</h3>

```
<pre class="programlisting">typedef enum {
DBus_GERROR_FAILED,
DBus_GERROR_NO_MEMORY,
DBus_GERROR_SERVICE_UNKNOWN,
DBus_GERROR_NAME_HAS_NO_OWNER,
DBus_GERROR_NO_REPLY,
DBus_GERROR_IO_ERROR,
DBus_GERROR_BAD_ADDRESS,
DBus_GERROR_NOT_SUPPORTED,
DBus_GERROR_LIMITS_EXCEEDED,
DBus_GERROR_ACCESS_DENIED,
DBus_GERROR_AUTH_FAILED,
DBus_GERROR_NO_SERVER,
DBus_GERROR_TIMEOUT,
DBus_GERROR_NO_NETWORK,
DBus_GERROR_ADDRESS_IN_USE,
DBus_GERROR_DISCONNECTED,
DBus_GERROR_INVALID_ARGS,
DBus_GERROR_FILE_NOT_FOUND,
DBus_GERROR_FILE_EXISTS,
DBus_GERROR_UNKNOWN_METHOD,
DBus_GERROR_TIMED_OUT,
DBus_GERROR_MATCH_RULE_NOT_FOUND,
DBus_GERROR_MATCH_RULE_INVALID,
DBus_GERROR_SPAWN_EXEC_FAILED,
DBus_GERROR_SPAWN_FORK_FAILED,
DBus_GERROR_SPAWN_CHILD_EXITED,
DBus_GERROR_SPAWN_CHILD_SIGNALED,
DBus_GERROR_SPAWN_FAILED,
DBus_GERROR_UNIX_PROCESS_ID_UNKNOWN,
DBus_GERROR_INVALID_SIGNATURE,
DBus_GERROR_INVALID_FILE_CONTENT,
DBus_GERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN,
DBus_GERROR_REMOTE_EXCEPTION
} DBusGError;
```

</pre>

<p>

A <http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError> GError enumeration for the domain

[DBusGError.html#DBUS-GERROR:CAPS](#) title="DBUS_GERROR"><code class="literal">DBUS_GERROR</code>. The values' meanings can be found by looking at the comments for the corresponding constants

in dbus-protocol.h.

</p>

</div>

<hr>

<div class="refsect2">

<h3>dbus_g_error_has_name ()</h3>

<pre class="programlisting"><a

href="http://library.gnome.org/devel/glib/unstable/glib-Basic-

Types.html#gboolean">gboolean

dbus_g_error_has_name (<em class="parameter"><code><a

href="http://library.gnome.org/devel/glib/unstable/glib-Error-

Reporting.html#GError">GError

*error</code>,

<em

class="parameter"><code>const char

*name</code>);</pre>

<p>

Determine whether D-BUS error name for a remote exception matches the given name. This function is intended to be invoked on a

GError returned from an invocation of a remote method, e.g. via

<a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-proxy-end-call

title="dbus_g_proxy_end_call ()"><code

class="function">dbus_g_proxy_end_call()</code>. It will silently return <a href="http://library.gnome.org/devel/glib/unstable/glib-

Standard-Macros.html#FALSE:CAPS"><code

class="literal">FALSE</code> for errors

which are not remote D-BUS exceptions (i.e. with a domain other than <a class="link" href="dbus-glib-DBusGError.html#DBUS-GERROR:CAPS"

title="DBUS_GERROR"><code class="literal">DBUS_GERROR</code> or a code other than

<code class="literal">DBUS_GERROR_REMOTE_EXCEPTION</code>).

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>error</code> :</p></td>

<td>the GError given from the remote method</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>name</code> :</p></td>

<td>the D-BUS error name</td>

</tr>

<tr>

<td>

Returns \hat{A} :

[TRUE](http://library.gnome.org/devel/glib/unstable/glib-Standard-Macros.html#TRUE:CAPS) if and only if the remote error has the given name

dbus_g_error_get_name ()

```

const char *          dbus_g_error_get_name
(<a href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError">GError</a>
 *error);
  
```

This function may only be invoked on a [GError](http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError) returned from an invocation of a remote method, e.g. via [dbus_g_proxy_end_call \(\)](dbus-glib-DBusGProxy.html#dbus-g-proxy-end-call). Moreover, you must ensure that the error's domain is [DBUS_GERROR](dbus-glib-DBusGError.html#DBUS-GERROR:CAPS), and the code is `DBUS_GERROR_REMOTE_EXCEPTION`.

<code>error</code> \hat{A} :
the GError given from the remote method
Returns \hat{A} :
the D-BUS name for a remote exception.

```

<hr>
<div class="refsect2">
<a name="dbus-g-error-domain-
register"></a><h3>dbus_g_error_domain_register ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_error_domain_register      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Quarks.html#GQuark"><span class="type">GQuark</span></a>
domain</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*default_iface</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
code_enum</code></em>);</pre>
<p>
Register a <a href="http://library.gnome.org/devel/glib/unstable/glib-
Error-Reporting.html#GError"><span class="type">GError</span></a>
domain and set of codes with D-Bus. When an object
raises a <a href="http://library.gnome.org/devel/glib/unstable/glib-
Error-Reporting.html#GError"><span class="type">GError</span></a> in
the domain <em class="parameter"><code>domain</code></em> from one of
its D-Bus methods,
the D-Bus error name used will be <em
class="parameter"><code>default_iface</code></em>, followed by a dot,
followed by the <span class="type">GEnumValue.value_nick</span>
corresponding to the <span class="type">GError.code</span>.
For D-Bus, it's conventional to use an error name (value_nick) that is
in CamelCase.
</p>
<p>
(For instance, if a D-Bus method <code
class="code">com.example.MyObject.GetThings</code>
can raise a <a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a> with
domain <code class="code">MY_ERROR</code> and code
<code class="code">MY_ERROR_NOT_HAPPY</code>, you could call
<code class="code">dbus_g_error_domain_register (MY_ERROR,
"com.example.MyError",
MY_TYPE_ERROR)</code>, and set up the value_nick for
<code class="code">MY_ERROR_NOT_HAPPY</code> to be <code
class="code">NotHappy</code>,
resulting in the D-Bus error string
<code class="code">com.example.MyError.NotHappy</code>.)
</p>
<p>
If <em class="parameter"><code>default_iface</code></em> is <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-

```

Macros.html#NULL:CAPS"><code class="literal">NULL</code>, the D-Bus interface of the method that failed will be used.

</p>

<p>

(For instance, if the above example had called <code class="code">dbus_g_error_domain_register (MY_ERROR, NULL, MY_TYPE_ERROR)</code>

instead, then the D-Bus error string would be

<code class="code">com.example.MyObject.NotHappy</code>.)

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>domain</code>Â :</p></td>

<td>the GError

domain</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>default_iface</code>Â :</p></td>

<td>the prefix used for error values, or <code class="literal">NULL</code>

</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>code_enum</code>Â :</p></td>

<td>a GType for a

GEnum of the error codes</td>

</tr>

</tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<h3>DBUS_GERROR</h3>

<pre class="programlisting">#define DBUS_G_ERROR dbus_g_error_quark ()</pre>

<p>

Expands to a function call returning the error domain quark for DBusGError,&br/>for use with <a

href="http://library.gnome.org/devel/glib/unstable/glib-Error-

Reporting.html#GError">GError.

```

</p>
</div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGError.see-also"></a><h2>See Also</h2>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBusGMessage.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBusGMessage</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings - Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBus-GObject-related-functions.html" title="DBus GObject related functions">
<link rel="next" href="dbus-glib-DBusGMethod.html" title="DBusGMethod">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBus-GObject-related-functions.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference Manual</th>
<td><a accesskey="n" href="dbus-glib-DBusGMethod.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">

```

```

<a href="#dbus-glib-DBusGMessage.synopsis" class="shortcut">Top</a>
    Â |Â
    <a href="#dbus-glib-DBusGMessage.description"
class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBusGMessage"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-
DBusGMessage.top_of_page"></a>DBusGMessage</span></h2>
<p>DBusGMessage "DBus Message"</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBusGMessage.stability-level"></a><h2>Stability
Level</h2>
Stable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBusGMessage.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h>;

    <a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage">DBusGMessage</a>;
#define    <a class="link" href="dbus-glib-
DBusGMessage.html#DBUS-TYPE-G-MESSAGE:CAPS"
title="DBUS_TYPE_G_MESSAGE">DBUS_TYPE_G_MESSAGE</a>
<a class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"
title="DBusGMessage"><span class="returnvalue">DBusGMessage</span></a>
*    <a class="link" href="dbus-glib-DBusGMessage.html#dbus-g-
message-ref" title="dbus_g_message_ref ()">dbus_g_message_ref</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage"><span
class="type">DBusGMessage</span></a> *message</code></em>);
<span class="returnvalue">void</span>    <a class="link"
href="dbus-glib-DBusGMessage.html#dbus-g-message-unref"
title="dbus_g_message_unref ()">dbus_g_message_unref</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage"><span
class="type">DBusGMessage</span></a> *message</code></em>);
<span class="returnvalue">DBusMessage</span> *    <a class="link"
href="dbus-glib-DBusGMessage.html#dbus-g-message-get-message"
title="dbus_g_message_get_message ()">dbus_g_message_get_message</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage"><span
class="type">DBusGMessage</span></a> *gmessage</code></em>);
</pre>
</div>

```


[Description](#)

A [DBusGMessage](#) is a boxed type abstracting a `DBusMessage`.

[Details](#)

```
typedef struct _DBusGMessage
DBusGMessage;
```

A [DBusGMessage](#) is a boxed type abstracting a `DBusMessage` from `libdbus`.

[DBUS_TYPE_G_MESSAGE](#)

```
#define DBUS_TYPE_G_MESSAGE
(dbus_g_message_get_g_type ())
```

Expands to a function call returning the boxed [GType](#) of a [DBusGConnection](#).

<p><code>dbus_g_message_get_g_type</code> : the <code>GLib</code> type</p>
--

```
dbus_g_message_ref ()
DBusGMessage * dbus_g_message_ref
(DBusGMessage DBusGMessage *message);
```

```

<p>
Increment refcount on a <a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage"><span
class="type">DBusGMessage</span></a>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>message</code></em>Â :</span></p></td>
<td>the message to ref</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the message that was ref'd</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-message-unref"></a><h3>dbus_g_message_unref ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_message_unref
      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"
title="DBusGMessage"><span class="type">DBusGMessage</span></a>
*message</code></em>);</pre>
<p>
Decrement refcount on a <a class="link" href="dbus-glib-
DBusGMessage.html#DBusGMessage" title="DBusGMessage"><span
class="type">DBusGMessage</span></a>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><em
class="parameter"><code>message</code></em>Â :</span></p></td>
<td>the message to unref</td>
</tr></tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-message-get-
message"></a><h3>dbus_g_message_get_message ()</h3>
<pre class="programlisting"><span
class="returnvalue">DBusMessage</span> *
dbus_g_message_get_message
      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"

```

```

title="DBusGMessage"><span class="type">DBusGMessage</span></a>
*gmessage</code></em>);</pre>
<p>
Get the <span class="type">DBusMessage</span> corresponding to this <a
class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"
title="DBusGMessage"><span class="type">DBusGMessage</span></a>.
The return value does not have its refcount incremented.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gmessage</code></em>Â :</span></p></td>
<td><a <a class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"
title="DBusGMessage"><span class="type">DBusGMessage</span></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td><span class="type">DBusMessage</span></td>
</tr>
</tbody>
</table></div>
</div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGMessage.see-also"></a><h2>See Also</h2>
<span class="type">DBusMessage</span>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBusGMethod.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBusGMethod</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBusGMessage.html"
title="DBusGMessage">

```

```

<link rel="next" href="dbus-glib-DBusGError.html" title="DBusGError">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBusGMessage.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBusGError.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBusGMethod.synopsis" class="shortcut">Top</a>
      Â |Â
      <a href="#dbus-glib-DBusGMethod.description"
class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBusGMethod"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-
DBusGMethod.top_of_page"></a>DBusGMethod</span></h2>
<p>DBusGMethod â€” GMethod Info &amp; Invocation</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBusGMethod.stability-level"></a><h2>Stability
Level</h2>
Stable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-DBusGMethod.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h&gt;

struct          <a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInfo" title="struct
DBusGMethodInfo">DBusGMethodInfo</a>;

```

```

        <a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation">DBusGMethodInvocation</a>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gchar"><span class="returnvalue">gchar</span></a> *
<a class="link" href="dbus-glib-DBusGMethod.html#dbus-g-method-get-
sender" title="dbus_g_method_get_sender
()">dbus_g_method_get_sender</a>
        (<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>);
<span class="returnvalue">DBusMessage</span> *
        <a class="link"
href="dbus-glib-DBusGMethod.html#dbus-g-method-get-reply"
title="dbus_g_method_get_reply ()">dbus_g_method_get_reply</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>);
<span class="returnvalue">void</span>
        <a class="link"
href="dbus-glib-DBusGMethod.html#dbus-g-method-send-reply"
title="dbus_g_method_send_reply ()">dbus_g_method_send_reply</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>,
        <em
class="parameter"><code><span class="type">DBusMessage</span>
*reply</code></em>);
<span class="returnvalue">void</span>
        <a class="link"
href="dbus-glib-DBusGMethod.html#dbus-g-method-return"
title="dbus_g_method_return ()">dbus_g_method_return</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>,
        <em
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span>
        <a class="link"
href="dbus-glib-DBusGMethod.html#dbus-g-method-return-error"
title="dbus_g_method_return_error ()">dbus_g_method_return_error</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>,
        <em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
*error</code></em>);
</pre>
</div>

```

<div class="refsect1">
<h2>Description</h2>
<p>

These types are used to call methods on GObject objects.

</p>
</div>

<div class="refsect1">
<h2>Details</h2>

<div class="refsect2">

<h3>struct DBusGMethodInfo</h3>

```
<pre class="programlisting">struct DBusGMethodInfo {  
    GCallback                function;  
    GClosureMarshal          marshaller;  
    int                       data_offset;  
};
```

</pre>
<p>

Object typically generated by dbus-binding-tool that

stores a mapping from introspection data to a function pointer for a C method to be invoked.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p>GCallbackÂ

<td>C method to invoke</td>

</tr>

<tr>

<td><p>GClosureMarshalÂ

<td>Marshaller to invoke method</td>

</tr>

<tr>

<td><p>intÂ

```

class="structfield"><code><a name="DBusGMethodInfo.data-
offset"></a>data_offset</code></em>;</span></p></td>
<td>Offset into the introspection data</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="DBusGMethodInvocation"></a><h3>DBusGMethodInvocation</h3>
<pre class="programlisting">typedef struct _DBusGMethodInvocation
DBusGMethodInvocation;</pre>
<p>
The context of an asynchronous method call. See <a class="link"
href="dbus-glib-DBusGMethod.html#dbus-g-method-return"
title="dbus_g_method_return ()"><code>
class="function">dbus_g_method_return()</code></a> and
<a class="link" href="dbus-glib-DBusGMethod.html#dbus-g-method-return-
error" title="dbus_g_method_return_error ()"><code>
class="function">dbus_g_method_return_error()</code></a>.
</p>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-method-get-sender"></a><h3>dbus_g_method_get_sender
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gchar"><span class="returnvalue">gchar</span></a> *
dbus_g_method_get_sender          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a>
*context</code></em>);</pre>
<p>
Get the sender of a message so we can send a
"reply" later (i.e. send a message directly
to a service which invoked the method at a
later time).
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em>Â :</span></p></td>
<td>the method context</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the unique name of the sender. It

```

```

is up to the caller to free the returned string.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-method-get-reply"></a><h3>dbus_g_method_get_reply
()</h3>
<pre class="programlisting"><span
class="returnvalue">DBusMessage</span> *          dbus_g_method_get_reply
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a>
*context</code></em>);</pre>
<p>
Get the reply message to append reply values
Used as a sidedoor when you can't generate dbus values
of the correct type due to glib binding limitations
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em> :</span></p></td>
<td>the method context</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>a <span class="type">DBusMessage</span> with the reply</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-method-send-reply"></a><h3>dbus_g_method_send_reply
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_method_send_reply          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>,
<em
class="parameter"><code><span class="type">DBusMessage</span>
*reply</code></em>);</pre>
<p>
Send a manually created reply message.
</p>

```


<p>

Used as a sidedoor when you can't generate dbus values of the correct type due to glib binding limitations

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em class="parameter"><code>context</code>Â :</p></td>

<td>the method context</td>

</tr>

<tr>

<td><p><em class="parameter"><code>reply</code>Â :</p></td>

<td>the reply message, will be unreffed</td>

</tr>

</tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<h3>dbus_g_method_return ()</h3>

<pre class="programlisting">void

dbus_g_method_return (<em class="parameter"><code><span

class="type">DBusGMethodInvocation *context</code>, <em

class="parameter"><code>...</code>);</pre>

<p>

Send a return message for a given method invocation, with arguments. This function also frees the sending context.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em class="parameter"><code>context</code>Â :</p></td>

<td>the method context</td>

</tr>

<tr>

<td><p><em class="parameter"><code>...</code>Â :</p></td>

<td>zero or more values to return from the method, with their number and types given by its <a class="link" href="dbus-glib-DBusGObject-related-functions.html#DBusGObjectInfo" title="struct

DBusGObjectInfo">DBusGObjectInfo

</td>

</tr>

</tbody>

```

</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-method-return-
error"></a><h3>dbus_g_method_return_error ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_method_return_error      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGMethod.html#DBusGMethodInvocation"
title="DBusGMethodInvocation"><span
class="type">DBusGMethodInvocation</span></a> *context</code></em>,
                                <em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
*error</code></em>);</pre>
<p>
Send a error message for a given method invocation.
This function also frees the sending context.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>context</code></em> :</span></p></td>
<td>the method context</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>error</code></em> :</span></p></td>
<td>the error to send</td>
</tr>
</tbody>
</table></div>
</div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGMethod.see-also"></a><h2>See Also</h2>
<a class="link" href="dbus-glib-DBusGMessage.html#DBusGMessage"
title="DBusGMessage"><span class="type">DBusGMessage</span></a>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-DBusGProxy.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DBusGProxy</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBusGError.html" title="DBusGError">
<link rel="next" href="dbus-glib-Specializable-GType-System.html"
title="Specializable GType System">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBusGError.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-Specializable-GType-
System.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">
<a href="#dbus-glib-DBusGProxy.synopsis" class="shortcut">Top</a>
      Â |Â
      <a href="#dbus-glib-DBusGProxy.description"
class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-DBusGProxy"></a><div class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-
DBusGProxy.top_of_page"></a>DBusGProxy</span></h2>
<p>DBusGProxy â€” DBus Proxy</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-DBusGProxy.stability-level"></a><h2>Stability
Level</h2>

```

Stable, unless otherwise indicated

</div>

<div class="refsynopsisdiv">

<h2>Synopsis</h2>

<pre class="synopsis">

#include <dbus/dbus-glib.h>

struct [DBusGProxy](dbus-glib-DBusGProxy.html#DBusGProxy);

[DBusGProxyCall](dbus-glib-DBusGProxy.html#DBusGProxyCall)

title="DBusGProxyCall">DBusGProxyCall;

void ([DBusGProxyCallNotify](dbus-glib-DBusGProxy.html#DBusGProxyCallNotify)

title="DBusGProxyCallNotify ()">*DBusGProxyCallNotify

(*<code>DBusGProxy *proxy</code>*

*<code>DBusGProxyCall *call_id</code>*

<code>void

*user_data</code>);

<span

class="returnvalue">DBusGProxy * [dbus_g_proxy_new_for_name](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name)

title="dbus_g_proxy_new_for_name ()">dbus_g_proxy_new_for_name

(*<code>DBusGConnection *connection</code>*

<code>const char

*name</code>

<code>const char

*path</code>);

<span

class="returnvalue">DBusGProxy * [dbus_g_proxy_new_for_name_owner](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name-owner)

title="dbus_g_proxy_new_for_name_owner

(*<code>DBusGConnection *connection</code>*

<code>const char

```

class="parameter"><code>const <span class="type">char</span>
*name</code></em>,
class="parameter"><code>const <span class="type">char</span>
*path</code></em>,
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);
<a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy"
title="struct DBusGProxy"><span
class="returnvalue">DBusGProxy</span></a> * <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-from-proxy"
title="dbus_g_proxy_new_from_proxy ()">dbus_g_proxy_new_from_proxy</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>,
class="parameter"><code>const <span class="type">char</span>
*path</code></em>);
<a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy"
title="struct DBusGProxy"><span
class="returnvalue">DBusGProxy</span></a> * <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-peer"
title="dbus_g_proxy_new_for_peer ()">dbus_g_proxy_new_for_peer</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span
class="type">DBusGConnection</span></a> *connection</code></em>,
class="parameter"><code>const <span class="type">char</span>
*path</code></em>,
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-set-interface"
title="dbus_g_proxy_set_interface ()">dbus_g_proxy_set_interface</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
class="parameter"><code>const <span class="type">char</span>
*interface_name</code></em>);

```

```

const <span class="returnvalue">char</span> *      <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-path"
title="dbus_g_proxy_get_path ()">dbus_g_proxy_get_path</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>);
const <span class="returnvalue">char</span> *      <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-bus-name"
title="dbus_g_proxy_get_bus_name ()">dbus_g_proxy_get_bus_name</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>);
const <span class="returnvalue">char</span> *      <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-interface"
title="dbus_g_proxy_get_interface ()">dbus_g_proxy_get_interface</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>);
<span class="returnvalue">void</span>              <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-add-signal"
title="dbus_g_proxy_add_signal ()">dbus_g_proxy_add_signal</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*signal_name</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span>              <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-connect-signal"
title="dbus_g_proxy_connect_signal ()">dbus_g_proxy_connect_signal</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*signal_name</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GCallback"><span class="type">GCallback</span></a>
handler</code></em>,
<em
class="parameter"><code><span class="type">void</span>
*data</code></em>,

```

```

class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureNotify"><span
class="type">GClosureNotify</span></a> free_data_func</code></em>;
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-disconnect-signal"
title="dbus_g_proxy_disconnect_signal
()">dbus_g_proxy_disconnect_signal</a> (<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*signal_name</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GCallback"><span class="type">GCallback</span></a>
handler</code></em>,
<em
class="parameter"><code><span class="type">void</span>
*data</code></em>;
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-send"
title="dbus_g_proxy_send ()">dbus_g_proxy_send</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code><span class="type">DBusMessage</span>
*message</code></em>,
<em
class="parameter"><code><span class="type">dbus_uint32_t</span>
*client_serial</code></em>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-proxy-call"
title="dbus_g_proxy_call ()">dbus_g_proxy_call</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>,
<em
class="parameter"><code><a

```

```

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-proxy-call-
with-timeout" title="dbus_g_proxy_call_with_timeout
()">dbus_g_proxy_call_with_timeout</a> (<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-call-no-reply"
title="dbus_g_proxy_call_no_reply ()">dbus_g_proxy_call_no_reply</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxyCall"
title="DBusGProxyCall"><span
class="returnvalue">DBusGProxyCall</span></a> * <a class="link"

```



```

href="dbus-glib-DBusGProxy.html#dbus-g-proxy-begin-call"
title="dbus_g_proxy_begin_call ()">dbus_g_proxy_begin_call</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCallNotify" title="DBusGProxyCallNotify
()"><span class="type">DBusGProxyCallNotify</span></a>
notify</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Datasets.html#GDestroyNotify"><span
class="type">GDestroyNotify</span></a> destroy</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxyCall"
title="DBusGProxyCall"><span
class="returnvalue">DBusGProxyCall</span></a> * <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-begin-call-with-timeout"
title="dbus_g_proxy_begin_call_with_timeout
()">dbus_g_proxy_begin_call_with_timeout</a>
(<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCallNotify" title="DBusGProxyCallNotify
()"><span class="type">DBusGProxyCallNotify</span></a>
notify</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-

```

```

Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Datasets.html#GDestroyNotify"><span
class="type">GDestroyNotify</span></a> destroy</code></em>,
<em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-proxy-end-call"
title="dbus_g_proxy_end_call ()">dbus_g_proxy_end_call</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="type">DBusGProxyCall</span></a> *call</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-cancel-call"
title="dbus_g_proxy_cancel_call ()">dbus_g_proxy_cancel_call</a>
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="type">DBusGProxyCall</span></a> *call</code></em>);

```

```

<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-DBusGProxy.html#dbus-g-proxy-set-default-timeout"
title="dbus_g_proxy_set_default_timeout
()">dbus_g_proxy_set_default_timeout</a>    (<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>);
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGProxy.description"></a><h2>Description</h2>
<p>
A <a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy"
title="struct DBusGProxy"><span class="type">DBusGProxy</span></a> is
a <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
The-Base-Object-Type.html#GObject"><span
class="type">GObject</span></a> representing a remote object in a D-
Bus
service.
</p>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGProxy.details"></a><h2>Details</h2>
<div class="refsect2">
<a name="DBusGProxy"></a><h3>struct DBusGProxy</h3>
<pre class="programlisting">struct DBusGProxy {
};
</pre>
<p>
A <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
The-Base-Object-Type.html#GObject"><span
class="type">GObject</span></a> representing a remote object in a D-
Bus service.
</p>
</div>
<hr>
<div class="refsect2">
<a name="DBusGProxyCall"></a><h3>DBusGProxyCall</h3>
<pre class="programlisting">typedef struct _DBusGProxyCall
DBusGProxyCall;</pre>
<p>
An opaque pointer representing an asynchronous call in progress.
</p>
</div>
<hr>
<div class="refsect2">
<a name="DBusGProxyCallNotify"></a><h3>DBusGProxyCallNotify ()</h3>

```

```

<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGProxyCallNotify)      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                <em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="type">DBusGProxyCall</span></a> *call_id</code></em>,
                                <em
class="parameter"><code><span class="type">void</span>
*user_data</code></em>);</pre>

```

Called when a reply to the call represented by `call_id` arrives. Use `dbus_g_proxy_end_call ()` to see whether `call_id` succeeded or failed, and get the arguments returned (if any) on success.

| |
|---|
| <p> <code>proxy</code> : the proxy on which the method was called </p> |
| <p> <code>call_id</code> : the call in progress </p> |
| <p> <code>user_data</code> : data passed to <code>dbus_g_proxy_begin_call ()</code> or similar </p> |

[dbus_g_proxy_new_for_name \(\)](#)

```

<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="returnvalue">DBusGProxy</span></a> *

```

```
dbus_g_proxy_new_for_name      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>,>
```

```
                                <em
class="parameter"><code>const <span class="type">char</span>
*name</code></em>,>
```

```
                                <em
class="parameter"><code>const <span class="type">char</span>
*path</code></em>,>
```

```
                                <em
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>);</pre>
```

<p>

Creates a new proxy for a remote interface exported by a connection on a message bus. Method calls and signal connections over this proxy will go to the name owner; the name's owner is expected to support the given interface name. THE NAME OWNER MAY CHANGE OVER TIME, for example between two different method calls, unless the name is a unique name. If you need a fixed owner, you need to request the current owner and bind a proxy to its unique name rather than to the generic name; see

<code class="function">dbus_g_proxy_new_for_name_owner()</code>.

</p>

<p>

A name-associated proxy only makes sense with a message bus, not for app-to-app direct dbus connections.

</p>

<p>

This proxy will only emit the "destroy" signal if the DBusConnection is disconnected, the proxy has no remaining references, or the name is a unique name and its owner disappears. If a well-known name changes owner, the proxy will still be alive.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em class="parameter"><code>connection</code>Â :</p></td>

<td>the connection to the remote bus</td>

</tr>

<tr>

<td><p><em class="parameter"><code>name</code>Â :</p></td>

<td>any name on the message bus</td>

</tr>

<tr>

| | |
|--|---|
| <p> class="parameter"><code>path</code> : </p> | <p> name of the object instance to call methods on </p> |
| <p> class="parameter"><code>iface</code> : </p> | <p> name of the interface to call methods on </p> |
| <p> Returns : </p> | <p> new proxy object </p> |

[dbus_g_proxy_new_for_name_owner \(\)](#)

```

class="programlisting"><a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span class="returnvalue">DBusGProxy</span></a> *
dbus_g_proxy_new_for_name_owner (<em class="parameter"><code><a class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection" title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>,
<em class="parameter"><code>const <span class="type">char</span>
*name</code></em>,
<em class="parameter"><code>const <span class="type">char</span>
*path</code></em>,
<em class="parameter"><code>const <span class="type">char</span>
*iface</code></em>,
<em class="parameter"><code><a href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>);</pre>


Similar to dbus\_g\_proxy\_new\_for\_name \(\), but makes a round-trip request to the message bus to get the current name owner, then binds the proxy to the unique name of the current owner, rather than to the well-known name. As a result, the name owner will not change over time, and the proxy will emit the "destroy" signal when the owner disappears from the message bus.


```

</p>

<p>

An example of the difference between [dbus-g-proxy-new-for-name](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name)

`dbus_g_proxy_new_for_name ()`

`dbus_g_proxy_new_for_name ()`

and [dbus-g-proxy-new-for-name-owner](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name-owner)

`dbus_g_proxy_new_for_name_owner ()`

`dbus_g_proxy_new_for_name_owner ()`: if you

provide the well-known name

`"org.freedesktop.Database"` [dbus-g-proxy-new-for-name](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name)

`dbus_g_proxy_new_for_name ()`

`dbus_g_proxy_new_for_name ()` remains bound

to that name as it changes owner. [dbus-g-proxy-new-for-name-owner](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name-owner)

`dbus_g_proxy_new_for_name_owner ()`

`dbus_g_proxy_new_for_name_owner ()`

will fail if the name has no owner. If the name has an owner,

[dbus-g-proxy-new-for-name-owner](dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name-owner)

`dbus_g_proxy_new_for_name_owner ()`

`dbus_g_proxy_new_for_name_owner ()` will

bind to the unique name

of that owner rather than the generic name.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>connection</code>Â :</p></td>

<td>the connection to the remote bus</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>name</code>Â :</p></td>

<td>any name on the message bus</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>path</code>Â :</p></td>

<td>name of the object inside the service to call methods on</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>iface</code>Â :</p></td>

<td>name of the interface to call methods on</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>error</code>Â :</p></td>

<td>return location for an error</td>

```

</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>new proxy object, or <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#NULL:CAPS"><code class="literal">NULL</code></a> on
error</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-new-from-
proxy"></a><h3>dbus_g_proxy_new_from_proxy ()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="returnvalue">DBusGProxy</span></a> *
dbus_g_proxy_new_from_proxy          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*path</code></em>);</pre>
<p>
Creates a proxy using an existing proxy as a template, substituting
the specified interface and path. Either or both may be NULL.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em>Â :</span></p></td>
<td>the proxy to use as a template</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iface</code></em>Â :</span></p></td>
<td>name of the interface to call methods on</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>path</code></em>Â :</span></p></td>
<td>of the object inside the peer to call methods on</td>
</tr>
<tr>

```



```

<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>new proxy object</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-new-for-peer"></a><h3>dbus_g_proxy_new_for_peer
()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="returnvalue">DBusGProxy</span></a> *
dbus_g_proxy_new_for_peer          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGConnection.html#DBusGConnection"
title="DBusGConnection"><span class="type">DBusGConnection</span></a>
*connection</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*path</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*iface</code></em>);</pre>
<p>
Creates a proxy for an object in peer application (one
we're directly connected to). That is, this function is
intended for use when there's no message bus involved,
we're doing a simple 1-to-1 communication between two
applications.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>connection</code></em>Â :</span></p></td>
<td>the connection to the peer</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>path</code></em>Â :</span></p></td>
<td>name of the object inside the peer to call methods on</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iface</code></em>Â :</span></p></td>
<td>name of the interface to call methods on</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>

```

```

<td>new proxy object</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-set-
interface"></a><h3>dbus_g_proxy_set_interface ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_set_interface          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*interface_name</code></em>);</pre>
<p>
Sets the object interface proxy is bound to
</p>
<p>
It is an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em> :</span></p></td>
<td>the proxy</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>interface_name</code></em> :</span></p></td>
<td>an object interface</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-get-path"></a><h3>dbus_g_proxy_get_path ()</h3>
<pre class="programlisting">const <span
class="returnvalue">char</span> *          dbus_g_proxy_get_path
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a>
*proxy</code></em>);</pre>
<p>
Gets the path this proxy is bound to
</p>
<p>

```

It is an error to call this method on a proxy that has emitted the `destroy` signal.

```
const class="returnvalue">char * dbus_g_proxy_get_bus_name
```

```
(class="parameter"><a class="link" href="dbus-glib-
```

```
class="parameter">DBusGProxy.html#DBusGProxy class="type">DBusGProxycode></em>);
```

Gets the bus name a proxy is bound to (may be `NULL` in some cases).

If you created the proxy with `dbus_g_proxy_new_for_name()`,

If you created it with `dbus_g_proxy_new_for_name_owner()`,

unique connection name will be returned. If you created it

It is an error to call this method on a proxy that has emitted the `destroy` signal.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em> :</span></p></td>
<td>the proxy</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>the bus name the proxy sends messages to</td>
</tr>
</tbody>
</table></div>
</div>
```

```
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-get-
interface"></a><h3>dbus_g_proxy_get_interface ()</h3>
<pre class="programlisting">const <span
class="returnvalue">char</span> *      dbus_g_proxy_get_interface
(<em class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>);</pre>
<p>
Gets the object interface proxy is bound to (may be <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#NULL:CAPS"><code class="literal">NULL</code></a> in some
cases).
</p>
```

It is an error to call this method on a proxy that has emitted the `destroy` signal.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em> :</span></p></td>
<td>the proxy</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>an object interface</td>
</tr>
</tbody>
```

```

</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-add-signal"></a><h3>dbus_g_proxy_add_signal
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_add_signal          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                <em
class="parameter"><code>const <span class="type">char</span>
*signal_name</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_type</code></em>,
                                <em
class="parameter"><code>...</code></em>);</pre>
<p>
Specifies the argument signature of a D-Bus signal. When the signal is
emitted by the remote object, if the GTypes corresponding to its
arguments'
types do not match the types given here, the signal will be ignored.
</p>
<p>
It is an error to add the same <em
class="parameter"><code>signal_name</code></em> to the same <em
class="parameter"><code>proxy</code></em> more than
once, even if the argument types given are the same.
</p>
<p>
It is also an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em> :</span></p></td>
<td>the proxy for a remote interface</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>signal_name</code></em> :</span></p></td>
<td>the name of the signal</td>
</tr>
<tr>

```

```

<td><p><span class="term"><em
class="parameter"><code>first_type</code></em>Â :</span></p></td>
<td>the first argument type, or <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> if none</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>any subsequent argument types, followed by <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-connect-
signal"></a><h3>dbus_g_proxy_connect_signal ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_connect_signal
      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
      <em
class="parameter"><code>const <span class="type">char</span>
*signal_name</code></em>,
      <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GCallback"><span class="type">GCallback</span></a>
handler</code></em>,
      <em
class="parameter"><code><span class="type">void</span>
*data</code></em>,
      <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Closures.html#GClosureNotify"><span
class="type">GClosureNotify</span></a>
free_data_func</code></em>);</pre>
<p>
Connect a signal handler to a proxy for a remote interface. When
the remote interface emits the specified signal, the proxy will
emit a corresponding GLib signal.
</p>
<p>
It is an error to call this method on a proxy that has emitted

```

the `destroy` signal.

<code>proxy</code>	a proxy for a remote interface
<code>signal_name</code>	the DBus signal name to listen for
<code>handler</code>	the handler to connect
<code>data</code>	data to pass to handler
<code>free_data_func</code>	callback function to destroy data

dbus_g_proxy_disconnect_signal

```
dbus_g_proxy_disconnect_signal (parameter <a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct DBusGProxy">DBusGProxy</a> *proxy, parameter const char *signal_name, parameter <a href="http://library.gnome.org/devel/gobject/unstable/gobject-Closures.html#GCallback">GCallback</a> handler)
```

```

class="parameter"><code><span class="type">void</span>
*data</code></em>);</pre>
<p>
Disconnect all signal handlers from a proxy that match the given
criteria.
</p>
<p>
It is an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em>Â :</span></p></td>
<td>a proxy for a remote interface</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>signal_name</code></em>Â :</span></p></td>
<td>the DBus signal name to disconnect</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>handler</code></em>Â :</span></p></td>
<td>the handler to disconnect</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>data</code></em>Â :</span></p></td>
<td>the data that was registered with handler</td>
</tr>
</tbody>
</table></div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-send"></a><h3>dbus_g_proxy_send ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_send
(

```


Sends a message to the interface we're proxying for. Does not block or wait for a reply. The message is only actually written out when you return to the main loop or block in

[`dbus_g_connection_flush\(\)`](dbus-glib-DBusGConnection.html#dbus-g-connection-flush "dbus_g_connection_flush ()")

The message is modified to be addressed to the target interface. That is, a destination name field or whatever is needed will be added to the message. The basic point of this function is to add the necessary header fields, otherwise it's equivalent to `dbus_connection_send()`.

This function adds a reference to the message, so the caller still owns its original reference.

It is an error to call this method on a proxy that has emitted the `destroy` signal.

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

--

```

dbus_g_proxy_call          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                                                    <em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
                                                                    <em
class="parameter"><code>...</code></em>);</pre>

```

Function for synchronously invoking a method and receiving reply values. This function is equivalent to `dbus_g_proxy_begin_call` followed by `dbus_g_proxy_end_call`. All of the input arguments are specified first, followed by `G_TYPE_INVALID`, followed by all of the output values, followed by a second `G_TYPE_INVALID`. Note that this means you must always specify `G_TYPE_INVALID` twice.

It is an error to call this method on a proxy that has emitted the `"destroy"` signal.

<pre> <div class="variablelist"><table border="0"> <col align="left" valign="top"> <tbody> <tr> <td><p><em class="parameter"><code>proxy</code> :</p></td> <td>a proxy for a remote interface</td> </tr> <tr> <td><p><em class="parameter"><code>method</code> :</p></td> <td>method to invoke</td> </tr> <tr> <td><p><em class="parameter"><code>error</code> :</p></td> <td>return location for an error</td> </tr> <tr> <td><p><em class="parameter"><code>first_arg_type</code> :</p></td> </pre>	<pre> </pre>
---	--------------

```

<td>type of first "in" argument, or <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> if none</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>value of first "in" argument, any further type/value pairs,
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a>, type/location pairs for
"out" arguments,
and <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Type-Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> again</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> if the
method succeeds, <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#FALSE:CAPS"><code class="literal">FALSE</code></a> if it
fails</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-call-with-
timeout"></a><h3>dbus_g_proxy_call_with_timeout ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_proxy_call_with_timeout      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
                                     <em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>,
                                     <em
class="parameter"><code><a

```

```
href="http://library.gnome.org/devel/glib/unstable/glib-Error-Reporting.html#GError"><span class="type">GError</span></a>  
**error</code></em>,  
                                <em
```

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType"><span class="type">GType</span></a>  
first_arg_type</code></em>,  
                                <em
```

```
class="parameter"><code>...</code></em>);</pre>  
<p>
```

Function for synchronously invoking a method and receiving reply values. This function is equivalent to `dbus_g_proxy_begin_call` followed by `dbus_g_proxy_end_call`. All of the input arguments are specified first, followed by `G_TYPE_INVALID`, followed by all of the output values, followed by a second `G_TYPE_INVALID`. Note that this means you must always specify `G_TYPE_INVALID` twice.

</p>

<p>

It is an error to call this method on a proxy that has emitted the `"destroy"` signal.

</p>

```
<div class="variablelist"><table border="0">
```

```
<col align="left" valign="top">
```

```
<tbody>
```

```
<tr>
```

```
<td><p><span class="term"><em  
class="parameter"><code>proxy</code></em> :</span></p></td>
```

```
<td>a proxy for a remote interface</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em  
class="parameter"><code>method</code></em> :</span></p></td>
```

```
<td>method to invoke</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em  
class="parameter"><code>timeout</code></em> :</span></p></td>
```

```
<td>the timeout in milliseconds, or -1 to use a default</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em  
class="parameter"><code>error</code></em> :</span></p></td>
```

```
<td>return location for an error</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em  
class="parameter"><code>first_arg_type</code></em> :</span></p></td>
```

```
<td>type of first "in" argument</td>
```

```
</tr>
```

```
<tr>
```

```

<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>as for <a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-
proxy-call" title="dbus_g_proxy_call ()"><code
class="function">dbus_g_proxy_call()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> if the
method succeeds, <a
href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#FALSE:CAPS"><code class="literal">FALSE</code></a> if it
fails</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-call-no-
reply"></a><h3>dbus_g_proxy_call_no_reply ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_call_no_reply      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
                                     <em
class="parameter"><code>...</code></em>);</pre>
<p>
Sends a method call message as with <a class="link" href="dbus-glib-
DBusGProxy.html#dbus-g-proxy-begin-call"
title="dbus_g_proxy_begin_call ()"><code
class="function">dbus_g_proxy_begin_call()</code></a>, but
does not ask for a reply or allow you to receive one.
</p>
<p>
It is an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<p>

```

TODO: this particular function shouldn't die on out of memory, since you should be able to do a call with large arguments.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em>Â :</span></p></td>
<td>a proxy for a remote interface</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>method</code></em>Â :</span></p></td>
<td>the name of the method to invoke</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>first_arg_type</code></em>Â :</span></p></td>
<td>type of the first argument, or <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> to call
the method without arguments</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>the first argument and any remaining type/argument pairs, followed
by
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> to terminate the list</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-begin-call"></a><h3>dbus_g_proxy_begin_call
()</h3>
<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="returnvalue">DBusGProxyCall</span></a> *
dbus_g_proxy_begin_call          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,>
```

```
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCallNotify" title="DBusGProxyCallNotify
()"><span class="type">DBusGProxyCallNotify</span></a>
notify</code></em>,</pre>
```

```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>,</pre>
```

```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Datasets.html#GDestroyNotify"><span
class="type">GDestroyNotify</span></a> destroy</code></em>,</pre>
```

```
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,</pre>
```

```
class="parameter"><code>...</code></em>);</pre><p>
```

Asynchronously invokes a method on a remote interface. The method call will not be sent over the wire until the application returns to the main loop, or blocks in <code>dbus_g_connection_flush()</code> to write out pending data. The call will be completed after a timeout, or when a reply is received. When the call returns, the callback specified will be invoked; you can then collect the results of the call (which may be an error, or a reply), use <code>dbus_g_proxy_end_call()</code>.

It is an error to call this method on a proxy that has emitted the "destroy" signal.

TODO this particular function shouldn't die on out of memory, since you should be able to do a call with large arguments.

```
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em>
class="parameter"><code>proxy</code></em> :</span></p></td>
<td>a proxy for a remote interface</td>
```

```

</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>method</code></em>Â :</span></p></td>
<td>the name of the method to invoke</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>notify</code></em>Â :</span></p></td>
<td>callback to be invoked when method returns</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
<td>user data passed to callback</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>destroy</code></em>Â :</span></p></td>
<td>function called to destroy user_data</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>first_arg_type</code></em>Â :</span></p></td>
<td>type of the first argument, or <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> if there
are no arguments</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>first argument, followed by any further type/value pairs, followed
by <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Type-Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>call identifier.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-begin-call-with-
timeout"></a><h3>dbus_g_proxy_begin_call_with_timeout ()</h3>

```



```

<pre class="programlisting"><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="returnvalue">DBusGProxyCall</span></a> *
dbus_g_proxy_begin_call_with_timeout
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxy" title="struct DBusGProxy"><span
class="type">DBusGProxy</span></a> *proxy</code></em>,
<em
class="parameter"><code>const <span class="type">char</span>
*method</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCallNotify" title="DBusGProxyCallNotify
()"><span class="type">DBusGProxyCallNotify</span></a>
notify</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
Datasets.html#GDestroyNotify"><span
class="type">GDestroyNotify</span></a> destroy</code></em>,
<em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
<em
class="parameter"><code>...</code></em>);</pre>
<p>
Asynchronously invokes a method on a remote interface. The method
call will not be sent over the wire until the application returns
to the main loop, or blocks in <a class="link" href="dbus-glib-
DBusGConnection.html#dbus-g-connection-flush"
title="dbus_g_connection_flush ()"><code
class="function">dbus_g_connection_flush()</code></a> to write out
pending data. The call will be completed after a timeout, or when
a reply is received. When the call returns, the callback specified
will be invoked; you can then collect the results of the call
(which may be an error, or a reply), use <a class="link" href="dbus-
glib-DBusGProxy.html#dbus-g-proxy-end-call"
title="dbus_g_proxy_end_call ()"><code
class="function">dbus_g_proxy_end_call()</code></a>.

```

</p>

<p>

It is an error to call this method on a proxy that has emitted the "destroy" signal.

</p>

<p>

TODO this particular function shouldn't die on out of memory, since you should be able to do a call with large arguments.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>proxy</code>Â :</p></td>

<td>a proxy for a remote interface</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>method</code>Â :</p></td>

<td>the name of the method to invoke</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>notify</code>Â :</p></td>

<td>callback to be invoked when method returns</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>user_data</code>Â :</p></td>

<td>user data passed to callback</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>destroy</code>Â :</p></td>

<td>function called to destroy user_data</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>timeout</code>Â :</p></td>

<td>the timeout in milliseconds, or -1 to use a default</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>first_arg_type</code>Â :</p></td>

<td>type of the first argument, or <code

class="literal">G_TYPE_INVALID</code> if there

are no arguments</td>

</tr>

<tr>

```

<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>first argument, followed by any further type/value pairs, followed
by <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Type-Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>call identifier.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-end-call"></a><h3>dbus_g_proxy_end_call ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_proxy_end_call          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="type">DBusGProxyCall</span></a> *call</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Error-
Reporting.html#GError"><span class="type">GError</span></a>
**error</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_arg_type</code></em>,
                                     <em
class="parameter"><code>...</code></em>)</pre>
<p>
Collects the results of a method call. The method call was normally
initiated with <a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-
proxy-end-call" title="dbus_g_proxy_end_call ()"><code
class="function">dbus_g_proxy_end_call()</code></a>. You may use this
function
outside of the callback given to dbus_g_proxy_begin_call; in that
case this function will block if the results haven't yet been
received.
</p>

```

<p>
All D-Bus method calls can fail with a remote error. If this occurs,
the `error` will be set and this
function will return [`FALSE`](http://library.gnome.org/devel/glib/unstable/glib-Standard-Macros.html#FALSE:CAPS).
</p>

<p>
Otherwise, the "out" parameters and return value of the
method are stored in the provided varargs list.
The list should be terminated with `G_TYPE_INVALID`.
</p>

<p><p> A proxy for a remote interface </p></p>	<p><code>proxy</code> : A proxy for a remote interface</p>
<p><p> The pending call ID from <code>dbus_g_proxy_begin_call</code> </p></p>	<p><code>call</code> : the pending call ID from <code>dbus_g_proxy_begin_call</code> <code>dbus_g_proxy_begin_call()</code></p>
<p><p> Return location for an error </p></p>	<p><code>error</code> : return location for an error</p>
<p><p> Type of first "out" argument, or <code>G_TYPE_INVALID</code> if there are no "out" arguments </p></p>	<p><code>first_arg_type</code> : type of first "out" argument, or <code>G_TYPE_INVALID</code> if there are no "out" arguments</p>
<p><p> Return location for first "out" argument, followed by any further type/location pairs, followed by <code>G_TYPE_INVALID</code> </p></p>	<p><code>...</code> : return location for first "out" argument, followed by any further type/location pairs, followed by <code>G_TYPE_INVALID</code></p>

```

</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> on
success</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-cancel-call"></a><h3>dbus_g_proxy_cancel_call
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_cancel_call          (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
                                     <em
class="parameter"><code><a class="link" href="dbus-glib-
DBusGProxy.html#DBusGProxyCall" title="DBusGProxyCall"><span
class="type">DBusGProxyCall</span></a> *call</code></em>);</pre>
<p>
Cancels a pending method call. The method call was normally
initiated with <a class="link" href="dbus-glib-DBusGProxy.html#dbus-g-
proxy-begin-call" title="dbus_g_proxy_begin_call ()"><code
class="function">dbus_g_proxy_begin_call()</code></a>. This function
may not be used on pending calls that have already been
ended with dbus_g_proxy_end_call.
</p>
<p>
It is an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em>Â :</span></p></td>
<td>a proxy for a remote interface</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>call</code></em>Â :</span></p></td>
<td>the pending call ID from <a class="link" href="dbus-glib-
DBusGProxy.html#dbus-g-proxy-begin-call"
title="dbus_g_proxy_begin_call ()"><code
class="function">dbus_g_proxy_begin_call()</code></a>

```

```

</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-proxy-set-default-
timeout"></a><h3>dbus_g_proxy_set_default_timeout ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_proxy_set_default_timeout      (<em class="parameter"><code><a
class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy" title="struct
DBusGProxy"><span class="type">DBusGProxy</span></a>
*proxy</code></em>,
<em
class="parameter"><code><a
href="https://www.gnu.org/software/gnutls/reference/gnutls-
crypto.html#int"><span class="type">int</span></a>
timeout</code></em>);</pre>
<p>
Sets the default timeout to use for a proxy. This timeout will be
used in calls where the timeout is not specified, or is specified to
be -1.
If this timeout is also set to -1, libdbus will use a reasonable
default
value.
</p>
<p>
This is useful for long-running operations that takes longer than
the default timeout (which is a on the order of magnitude of tens
of seconds). For some applications, consider using a pattern where
the method returns once the operation is underway
(e.g. immediately) and emits a signal when the operation terminates
(though beware of leaking information with/in the signal return
value).
</p>
<p>
It is an error to call this method on a proxy that has emitted
the <span class="type">"destroy"</span> signal.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>proxy</code></em>Â :</span></p></td>
<td>a proxy for a remote interface</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>timeout</code></em>Â :</span></p></td>

```

```

<td>the timeout in milliseconds, or -1 to reset to the libdbus
default</td>
</tr>
</tbody>
</table></div>
<p class="since">Since 0.75</p>
</div>
</div>
<div class="refsect1">
<a name="dbus-glib-DBusGProxy.see-also"></a><h2>See Also</h2>
<a class="link" href="dbus-glib-DBusGProxy.html#DBusGProxy"
title="struct DBusGProxy"><span class="type">DBusGProxy</span></a>
</div>
</div>
<div class="footer">
<hr>
Generated by GTK-Doc V1.18</div>
</body>
</html>

```

File = dbus-glib-docs.sgml

```

<?xml version="1.0"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
"http://www.oasis-
open.org/docbook/xml/4.1.2/docbookx.dtd" [
<!ENTITY dbus-binding-tool SYSTEM "dbus-binding-tool.xml">
<!ENTITY version SYSTEM "version.xml">
]>
<book id="index" xmlns:xi="http://www.w3.org/2003/XInclude">
  <bookinfo>
    <title>D-Bus GLib bindings - Reference Manual</title>
    <releaseinfo>for version &version;</releaseinfo>
  </bookinfo>

  <chapter>
    <title>Introduction</title>

    <para>D-Bus is a message bus system, a simple way for applications
to talk to one another.</para>

    <para>D-Bus supplies both a system daemon (for events such as "new
hardware device added" or "printer queue changed") and a
per-user-login-session daemon (for general IPC needs among user
applications). Also, the message bus is built on top of a general
one-to-one message passing framework, which can be used by any two
apps to communicate directly (without going through the message
bus daemon). Currently the communicating applications are on one
computer, but TCP/IP option is available and remote support
planned.</para>

```

<para>The D-Bus API isn't finished yet, and the design is by no means set in stone. One of our main goals is for lots of projects to use it, so if you wouldn't use it, by all means mail us and say why - design, licensing, indentation style, we would rather know than not know.</para>

<para>D-Bus' only <emphasis>required</emphasis> dependency is an XML parser (either libxml or expat). D-Bus includes many language bindings to the D-Bus system, if you want to build those. The list of projects using D-Bus is growing and they provide a wealth of examples of using the various APIs to learn from.</para>

<para>Trying out D-Bus in sample applications is encouraged - we want to get it widely used and tested. It should be working pretty well these days, and when it isn't bug reports are very welcome.</para>

<para>For more information on D-Bus, see
<ulink url="http://www.freedesktop.org/wiki/Software/dbus">
http://www.freedesktop.org/wiki/Software/dbus</ulink>.&br/></para>

</chapter>

<chapter>

<title>API Reference</title>

<sect1 id="api-overview">

<title>API for using D-BUS with GLib</title>

<para>libdbus proper is a low-level API, these GLib bindings wrap libdbus with a much higher-level approach. The higher level approach is possible because GLib defines a main loop, an object/type system, and an out-of-memory handling policy (it exits the program). See <ulink url="http://www.gtk.org">http://www.gtk.org</ulink> for GLib information.</para>

<para>To manipulate remote objects, use DBusGProxy. </para>

</sect1>

<xi:include href="xml/dbus-gconnection.xml"/>

<xi:include href="xml/dbus-gobject.xml"/>

<xi:include href="xml/dbus-gmessage.xml"/>

<xi:include href="xml/dbus-gmethod.xml"/>

<xi:include href="xml/dbus-gerror.xml"/>

<xi:include href="xml/dbus-gproxy.xml"/>

<xi:include href="xml/dbus-gtype-specialized.xml"/>

<xi:include href="xml/dbus-glib-lowlevel.xml"/>

</chapter>

<chapter>

<title>Tools Reference</title>


```
&dbus-binding-tool;
```

```
</chapter>
```

```
</book>
```

```
File = dbus-glib-lowlevel.h
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-glib-lowlevel.h GLib integration details that require
dbus/dbus.h
*
* Copyright (C) 2002, 2003 CodeFactory AB
* Copyright (C) 2003, 2004 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_GLIB_LOWLEVEL_H
#define DBUS_GLIB_LOWLEVEL_H

#include <dbus/dbus-glib.h>
#include <dbus/dbus.h>

G_BEGIN_DECLS

void dbus_set_g_error (GError **gerror,
                      DBusError *derror);

#define DBUS_TYPE_CONNECTION (dbus_connection_get_g_type ())
#define DBUS_TYPE_MESSAGE (dbus_message_get_g_type ())
GType dbus_connection_get_g_type (void) G_GNUC_CONST;
```

```

GType dbus_message_get_g_type      (void) G_GNUC_CONST;

void          dbus_connection_setup_with_g_main (DBusConnection
*connection,
                                                    GMainContext
*context);
void          dbus_server_setup_with_g_main   (DBusServer
*server,
                                                    GMainContext
*context);

void dbus_g_proxy_send (DBusGProxy   *proxy,
                        DBusMessage  *message,
                        dbus_uint32_t *client_serial);

DBusConnection* dbus_g_connection_get_connection (DBusGConnection
*gconnection);
DBusGConnection* dbus_connection_get_g_connection (DBusConnection
*connection);
DBusMessage*      dbus_g_message_get_message     (DBusGMessage
*gmessage);

gchar*           dbus_g_method_get_sender       (DBusGMethodInvocation
*context);

DBusMessage*     dbus_g_method_get_reply        (DBusGMethodInvocation
*context);

void             dbus_g_method_send_reply       (DBusGMethodInvocation
*context,
                                                DBusMessage *reply);

G_END_DECLS

#endif /* DBUS_GLIB_LOWLEVEL_H */

```

File = dbus-glib-lowlevel.sgml

```

<!-- ##### SECTION Title ##### -->
DBus GLib low level

```

```

<!-- ##### SECTION Short_Description ##### -->

```

```

<!-- ##### SECTION Long_Description ##### -->
<para>

```

</para>

<!-- ##### SECTION See_Also ##### -->

<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### FUNCTION dbus_set_g_error ##### -->

<para>

</para>

@gerror:

@derror:

<!-- ##### FUNCTION dbus_connection_setup_with_g_main ##### -->

<para>

</para>

@connection:

@context:

<!-- ##### FUNCTION dbus_connection_get_g_connection ##### -->

<para>

</para>

@connection:

@Returns:

<!-- ##### FUNCTION dbus_server_setup_with_g_main ##### -->

<para>

</para>

@server:

@context:

<!-- ##### MACRO DBUS_TYPE_CONNECTION ##### -->

<para>

</para>

<!-- ##### MACRO DBUS_TYPE_MESSAGE ##### -->

<para>

</para>

File = dbus-glib-overrides.txt

File = dbus-glib-sections.txt

<SECTION>

<FILE>dbus-gconnection</FILE>

<TITLE>DBusGConnection</TITLE>

<INCLUDE>dbus/dbus-glib.h</INCLUDE>

DBusGConnection

DBUS_TYPE_G_CONNECTION

dbus_g_bus_get

dbus_g_bus_get_private

dbus_g_thread_init

dbus_g_connection_open

dbus_g_connection_ref

dbus_g_connection_unref

dbus_g_connection_flush

dbus_g_connection_get_connection

dbus_g_connection_register_g_object

dbus_g_connection_unregister_g_object

dbus_g_connection_lookup_g_object

<SUBSECTION Standard>

dbus_g_connection_get_g_type

</SECTION>

<SECTION>

<FILE>dbus-gobject</FILE>

<TITLE>DBus GObject related functions</TITLE>

<INCLUDE>dbus/dbus-glib.h</INCLUDE>

DBusGObjectInfo

dbus_g_object_type_install_info

dbus_g_object_type_register_shadow_property

dbus_g_object_path_get_g_type

```
dbus_g_object_register_marshalller
dbus_g_object_register_marshalller_array
dbus_glib_global_set_disable_legacy_property_access
</SECTION>
```

```
<SECTION>
<FILE>dbus-gmessage</FILE>
<TITLE>DBusGMessage</TITLE>
<INCLUDE>dbus/dbus-glib.h</INCLUDE>
DBusGMessage
DBUS_TYPE_G_MESSAGE
dbus_g_message_ref
dbus_g_message_unref
dbus_g_message_get_message
<SUBSECTION Standard>
dbus_g_message_get_g_type
</SECTION>
```

```
<SECTION>
<FILE>dbus-gmethod</FILE>
<TITLE>DBusGMethod</TITLE>
<INCLUDE>dbus/dbus-glib.h</INCLUDE>
DBusGMethodInfo
DBusGMethodInvocation
dbus_g_method_get_sender
dbus_g_method_get_reply
dbus_g_method_send_reply
dbus_g_method_return
dbus_g_method_return_error
</SECTION>
```

```
<SECTION>
<FILE>dbus-gerror</FILE>
<TITLE>DBusGError</TITLE>
<INCLUDE>dbus/dbus-glib.h</INCLUDE>
DBusGError
dbus_g_error_has_name
dbus_g_error_get_name
dbus_g_error_domain_register
DBUS_GERROR
<SUBSECTION Standard>
dbus_g_error_quark
</SECTION>
```

```
<SECTION>
<FILE>dbus-gproxy</FILE>
<TITLE>DBusGProxy</TITLE>
<INCLUDE>dbus/dbus-glib.h</INCLUDE>
DBusGProxy
DBusGProxyCall
DBusGProxyCallNotify
dbus_g_proxy_new_for_name
```

```

dbus_g_proxy_new_for_name_owner
dbus_g_proxy_new_from_proxy
dbus_g_proxy_new_for_peer
dbus_g_proxy_set_interface
dbus_g_proxy_get_path
dbus_g_proxy_get_bus_name
dbus_g_proxy_get_interface
dbus_g_proxy_add_signal
dbus_g_proxy_connect_signal
dbus_g_proxy_disconnect_signal
dbus_g_proxy_send
dbus_g_proxy_call
dbus_g_proxy_call_with_timeout
dbus_g_proxy_call_no_reply
dbus_g_proxy_begin_call
dbus_g_proxy_begin_call_with_timeout
dbus_g_proxy_end_call
dbus_g_proxy_cancel_call
dbus_g_proxy_set_default_timeout
<SUBSECTION Standard>
DBUS_G_PROXY
DBUS_IS_G_PROXY
DBUS_TYPE_G_PROXY
dbus_g_proxy_get_type
DBUS_G_PROXY_CLASS
DBUS_IS_G_PROXY_CLASS
DBUS_G_PROXY_GET_CLASS
<SUBSECTION Private>
DBusGAsyncData
</SECTION>

<SECTION>
<FILE>dbus-glib-lowlevel</FILE>
<TITLE>DBus GLib low level</TITLE>
<INCLUDE>dbus/dbus-glib-lowlevel.h</INCLUDE>
dbus_set_g_error
dbus_connection_setup_with_g_main
dbus_connection_get_g_connection
dbus_server_setup_with_g_main
DBUS_TYPE_CONNECTION
DBUS_TYPE_MESSAGE
<SUBSECTION Standard>
dbus_connection_get_g_type
dbus_message_get_g_type
<SUBSECTION Private>
DBUS_INSIDE_DBUS_GLIB_H
</SECTION>

<SECTION>
<FILE>dbus-gtype-specialized</FILE>
<TITLE>Specializable GType System</TITLE>
<INCLUDE>dbus/dbus-glib.h</INCLUDE>

```

DBusGTypeSpecializedCollectionIterator
DBusGTypeSpecializedMapIterator
DBusGTypeSpecializedAppendContext
DBusGTypeSpecializedConstructor
DBusGTypeSpecializedFreeFunc
DBusGTypeSpecializedCopyFunc
DBusGTypeSpecializedVtable
DBusGTypeSpecializedCollectionFixedAccessorFunc
DBusGTypeSpecializedCollectionIteratorFunc
DBusGTypeSpecializedCollectionAppendFunc
DBusGTypeSpecializedCollectionEndAppendFunc
DBusGTypeSpecializedCollectionVtable
DBusGTypeSpecializedMapIteratorFunc
DBusGTypeSpecializedMapAppendFunc
DBusGTypeSpecializedMapVtable
DBusGTypeSpecializedStructGetMember
DBusGTypeSpecializedStructSetMember
DBusGTypeSpecializedStructVtable
dbus_g_type_get_collection
dbus_g_type_get_map
dbus_g_type_get_structv
dbus_g_type_get_struct
dbus_g_type_is_collection
dbus_g_type_is_map
dbus_g_type_is_struct
dbus_g_type_get_collection_specialization
dbus_g_type_get_map_key_specialization
dbus_g_type_get_map_value_specialization
dbus_g_type_get_struct_member_type
dbus_g_type_get_struct_size
dbus_g_type_specialized_construct
dbus_g_type_specialized_init_append
dbus_g_type_specialized_collection_append
dbus_g_type_specialized_collection_end_append
dbus_g_type_specialized_map_append
dbus_g_type_collection_get_fixed
dbus_g_type_collection_value_iterate
dbus_g_type_map_value_iterate
dbus_g_type_struct_get_member
dbus_g_type_struct_set_member
dbus_g_type_struct_get
dbus_g_type_struct_set
dbus_g_type_specialized_init
dbus_g_type_register_collection
dbus_g_type_register_map
dbus_g_type_map_peek_vtable
dbus_g_type_collection_peek_vtable
dbus_g_type_struct_peek_vtable
dbus_g_type_register_struct
dbus_g_value_build_g_variant
dbus_g_value_parse_g_variant
DBUS_TYPE_G_BOOLEAN_ARRAY

```

DBUS_TYPE_G_UCHAR_ARRAY
DBUS_TYPE_G_UINT_ARRAY
DBUS_TYPE_G_INT_ARRAY
DBUS_TYPE_G_UINT64_ARRAY
DBUS_TYPE_G_INT64_ARRAY
DBUS_TYPE_G_OBJECT_ARRAY
DBUS_TYPE_G_STRING_STRING_HASHTABLE
DBusGSignature
DBUS_TYPE_G_SIGNATURE
DBusGObjectPath
DBUS_TYPE_G_OBJECT_PATH
<SUBSECTION Private>
dbus_g_object_path_get_g_type
dbus_g_signature_get_g_type
</SECTION>

```

File = dbus-glib-Specializable-GType-System.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Specializable GType System</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.76.1">
<link rel="home" href="index.html" title="D-Bus GLib bindings -
Reference Manual">
<link rel="up" href="ch02.html" title="API Reference">
<link rel="prev" href="dbus-glib-DBusGProxy.html" title="DBusGProxy">
<link rel="next" href="dbus-glib-DBus-GLib-low-level.html" title="DBus
GLib low level">
<meta name="generator" content="GTK-Doc V1.18 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" vlink="#840084"
alink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation
header" cellpadding="2" cellspacing="2">
<tr valign="middle">
<td><a accesskey="p" href="dbus-glib-DBusGProxy.html"></a></td>
<td><a accesskey="u" href="ch02.html"></a></td>
<td><a accesskey="h" href="index.html"></a></td>
<th width="100%" align="center">D-Bus GLib bindings - Reference
Manual</th>
<td><a accesskey="n" href="dbus-glib-DBus-GLib-low-level.html"></a></td>
</tr>
<tr><td colspan="5" class="shortcuts">

```



```

<a href="#dbus-glib-Specializable-GType-System.synopsis"
class="shortcut">Top</a>
    Â |Â
    <a href="#dbus-glib-Specializable-GType-
System.description" class="shortcut">Description</a>
</td></tr>
</table>
<div class="refentry">
<a name="dbus-glib-Specializable-GType-System"></a><div
class="titlepage"></div>
<div class="refnamediv"><table width="100%"><tr>
<td valign="top">
<h2><span class="refentrytitle"><a name="dbus-glib-Specializable-
GType-System.top_of_page"></a>Specializable GType System</span></h2>
<p>Specializable GType System â€” Specialized GTypes</p>
</td>
<td valign="top" align="right"></td>
</tr></table></div>
<div class="refsect1">
<a name="dbus-glib-Specializable-GType-System.stability-
level"></a><h2>Stability Level</h2>
Unstable, unless otherwise indicated
</div>
<div class="refsynopsisdiv">
<a name="dbus-glib-Specializable-GType-
System.synopsis"></a><h2>Synopsis</h2>
<pre class="synopsis">
#include &lt;dbus/dbus-glib.h&gt;

<span class="returnvalue">void</span>                (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCollectionIterator"
title="DBusGTypeSpecializedCollectionIterator
()">*DBusGTypeSpecializedCollectionIterator</a>)
                                                    (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);
<span class="returnvalue">void</span>                (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapIterator"
title="DBusGTypeSpecializedMapIterator
()">*DBusGTypeSpecializedMapIterator</a>) (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-

```

```

values.html#GValue"><span class="type">GValue</span></a>
*key_val</code></em>,
<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a>
*value_val</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>;
<a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext">DBusGTypeSpecializedAppendCo
ntext</a>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
(<a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedConstructor"
title="DBusGTypeSpecializedConstructor
()">*DBusGTypeSpecializedConstructor</a>) (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>);
<span class="returnvalue">void</span> (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedFreeFunc"
title="DBusGTypeSpecializedFreeFunc
()">*DBusGTypeSpecializedFreeFunc</a>) (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
val</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
(<a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCopyFunc"
title="DBusGTypeSpecializedCopyFunc
()">*DBusGTypeSpecializedCopyFunc</a>) (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a

```

```
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
src</code></em>);
    <a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedVtable"
title="DBusGTypeSpecializedVtable">DBusGTypeSpecializedVtable</a>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
(<a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionFixedAccessorFunc"
title="DBusGTypeSpecializedCollectionFixedAccessorFunc"
() ">*DBusGTypeSpecializedCollectionFixedAccessorFunc</a>)
    (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
*values</code></em>,
    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
*len</code></em>);
<span class="returnvalue">void</span>    (<a class="link"
href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionIteratorFunc"
title="DBusGTypeSpecializedCollectionIteratorFunc"
() ">*DBusGTypeSpecializedCollectionIteratorFunc</a>)
    (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
    <em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionIterator"
title="DBusGTypeSpecializedCollectionIterator" () "><span
```

```

class="type">DBusGTypeSpecializedCollectionIterator</span></a>
iterator</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);
<span class="returnvalue">void</span> (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCollectionAppendFunc"
title="DBusGTypeSpecializedCollectionAppendFunc
()">*DBusGTypeSpecializedCollectionAppendFunc</a>)
(<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);
<span class="returnvalue">void</span> (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCollectionEndAppendFunc"
title="DBusGTypeSpecializedCollectionEndAppendFunc
()">*DBusGTypeSpecializedCollectionEndAppendFunc</a>)
(<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);
<a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedCollectionVtable"
title="DBusGTypeSpecializedCollectionVtable">DBusGTypeSpecializedColle
ctionVtable</a>;
<span class="returnvalue">void</span> (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapIteratorFunc"
title="DBusGTypeSpecializedMapIteratorFunc
()">*DBusGTypeSpecializedMapIteratorFunc</a>)
(<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-

```

```

Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedMapIterator"
title="DBusGTypeSpecializedMapIterator ()"><span
class="type">DBusGTypeSpecializedMapIterator</span></a>
iterator</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);
<span class="returnvalue">void</span> (<a class="link"
href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapAppendFunc"
title="DBusGTypeSpecializedMapAppendFunc
()">*DBusGTypeSpecializedMapAppendFunc</a>)
(<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*key</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);
<a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedMapVtable"
title="DBusGTypeSpecializedMapVtable">DBusGTypeSpecializedMapVtable</a>
>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
(<a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedStructGetMember"
title="DBusGTypeSpecializedStructGetMember
()">*DBusGTypeSpecializedStructGetMember</a>)
(<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-

```

```

Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*ret_value</code></em>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
(<a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedStructSetMember"
title="DBusGTypeSpecializedStructSetMember
()">*DBusGTypeSpecializedStructSetMember</a>)
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
<em>
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*new_value</code></em>;
<a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedStructVtable"
title="DBusGTypeSpecializedStructVtable">DBusGTypeSpecializedStructVta-
ble</a>;
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-collection" title="dbus_g_type_get_collection
()">dbus_g_type_get_collection</a>
class="parameter"><code>const <span class="type">char</span>
*container</code></em>,
<em>
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-

```

```

Information.html#GType"><span class="type">GType</span></a>
specialization</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-map" title="dbus_g_type_get_map ()">dbus_g_type_get_map</a>
(<em class="parameter"><code>const <span class="type">char</span>
*container</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
key_specialization</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
value_specialization</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-structv" title="dbus_g_type_get_structv
()">dbus_g_type_get_structv</a> (<em
class="parameter"><code>const <span class="type">char</span>
*container</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
num_members</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
*types</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-struct" title="dbus_g_type_get_struct
()">dbus_g_type_get_struct</a> (<em
class="parameter"><code>const <span class="type">char</span>
*container</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
first_type</code></em>,
<em
class="parameter"><code>...</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-

```

```
g-type-is-collection" title="dbus_g_type_is_collection
()">dbus_g_type_is_collection</a>          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-is-map" title="dbus_g_type_is_map ()">dbus_g_type_is_map</a>
(<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-is-struct" title="dbus_g_type_is_struct
()">dbus_g_type_is_struct</a>          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-collection-specialization"
title="dbus_g_type_get_collection_specialization
()">dbus_g_type_get_collection_specialization</a>
                                                                    (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-map-key-specialization"
title="dbus_g_type_get_map_key_specialization
()">dbus_g_type_get_map_key_specialization</a>
                                                                    (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-map-value-specialization"
title="dbus_g_type_get_map_value_specialization
()">dbus_g_type_get_map_value_specialization</a>
```



```

                                                                    (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-struct-member-type"
title="dbus_g_type_get_struct_member_type
()">dbus_g_type_get_struct_member_type</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>,

```

```

                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="returnvalue">guint</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-get-struct-size" title="dbus_g_type_get_struct_size
()">dbus_g_type_get_struct_size</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-specialized-construct" title="dbus_g_type_specialized_construct
()">dbus_g_type_specialized_construct</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-init-append" title="dbus_g_type_specialized_init_append
()">dbus_g_type_specialized_init_append</a> (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,

```

```

                                                                    <em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);

```

```

<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-collection-append"
title="dbus_g_type_specialized_collection_append
()">dbus_g_type_specialized_collection_append</a>
      (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
      <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*elt</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-collection-end-append"
title="dbus_g_type_specialized_collection_end_append
()">dbus_g_type_specialized_collection_end_append</a>
      (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-map-append" title="dbus_g_type_specialized_map_append
()">dbus_g_type_specialized_map_append</a> (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
      <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*key</code></em>,
      <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-collection-get-fixed" title="dbus_g_type_collection_get_fixed
()">dbus_g_type_collection_get_fixed</a> (<em
class="parameter"><code><a

```

```

href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
*data_ret</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
*len_ret</code></em>;
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
collection-value-iterate" title="dbus_g_type_collection_value_iterate
()">dbus_g_type_collection_value_iterate</a>
(<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedCollectionIterator"
title="DBusGTypeSpecializedCollectionIterator ()"><span
class="type">DBusGTypeSpecializedCollectionIterator</span></a>
iterator</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>;
<span class="returnvalue">void</span> <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-map-value-
iterate" title="dbus_g_type_map_value_iterate
()">dbus_g_type_map_value_iterate</a> (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedMapIterator"
title="DBusGTypeSpecializedMapIterator ()"><span
class="type">DBusGTypeSpecializedMapIterator</span></a>
iterator</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>;

```

```
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-get-member" title="dbus_g_type_struct_get_member()">dbus_g_type_struct_get_member</a>      (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*dest</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-set-member" title="dbus_g_type_struct_set_member()">dbus_g_type_struct_set_member</a>      (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
                                                                    <em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*src</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-get" title="dbus_g_type_struct_get()">dbus_g_type_struct_get</a>      (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                                                    <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
```

```

class="parameter"><code>...</code></em>);
<a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-struct-set" title="dbus_g_type_struct_set
()">dbus_g_type_struct_set</a>          (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
class="parameter"><code>...</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-init" title="dbus_g_type_specialized_init
()">dbus_g_type_specialized_init</a>          (<em
class="parameter"><code><span class="type">void</span></code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-
collection" title="dbus_g_type_register_collection
()">dbus_g_type_register_collection</a>          (<em
class="parameter"><code>const <span class="type">char</span>
*name</code></em>,
class="parameter"><code>const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedCollectionVtable"
title="DBusGTypeSpecializedCollectionVtable"><span
class="type">DBusGTypeSpecializedCollectionVtable</span></a>
*vtable</code></em>,
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
flags</code></em>);
<span class="returnvalue">void</span>          <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-
map" title="dbus_g_type_register_map ()">dbus_g_type_register_map</a>
(<em class="parameter"><code>const <span class="type">char</span>
*name</code></em>,
class="parameter"><code>const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedMapVtable"
title="DBusGTypeSpecializedMapVtable"><span
class="type">DBusGTypeSpecializedMapVtable</span></a>
*vtable</code></em>,

```

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-  
Types.html#guint"><span class="type">guint</span></a>  
flags</code></em>);  
const <a class="link" href="dbus-glib-Specializable-GType-  
System.html#DBusGTypeSpecializedMapVtable"  
title="DBusGTypeSpecializedMapVtable"><span  
class="returnvalue">DBusGTypeSpecializedMapVtable</span></a> * <a  
class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-  
type-map-peek-vtable" title="dbus_g_type_map_peek_vtable  
( )">dbus_g_type_map_peek_vtable</a>
```

(<em

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-  
Information.html#GType"><span class="type">GType</span></a>  
map_type</code></em>);  
const <a class="link" href="dbus-glib-Specializable-GType-  
System.html#DBusGTypeSpecializedCollectionVtable"  
title="DBusGTypeSpecializedCollectionVtable"><span  
class="returnvalue">DBusGTypeSpecializedCollectionVtable</span></a> *  
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-  
type-collection-peek-vtable"  
title="dbus_g_type_collection_peek_vtable  
( )">dbus_g_type_collection_peek_vtable</a>
```

(<em

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-  
Information.html#GType"><span class="type">GType</span></a>  
collection_type</code></em>);  
const <a class="link" href="dbus-glib-Specializable-GType-  
System.html#DBusGTypeSpecializedStructVtable"  
title="DBusGTypeSpecializedStructVtable"><span  
class="returnvalue">DBusGTypeSpecializedStructVtable</span></a> * <a  
class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-  
type-struct-peek-vtable" title="dbus_g_type_struct_peek_vtable  
( )">dbus_g_type_struct_peek_vtable</a>
```

(<em

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-  
Information.html#GType"><span class="type">GType</span></a>  
struct_type</code></em>);  
<span class="returnvalue">void</span> <a class="link"  
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-  
struct" title="dbus_g_type_register_struct  
( )">dbus_g_type_register_struct</a> <em  
class="parameter"><code>const <span class="type">char</span>  
*name</code></em>,&br/>
```

<em

```
class="parameter"><code>const <a class="link" href="dbus-glib-  
Specializable-GType-System.html#DBusGTypeSpecializedStructVtable"  
title="DBusGTypeSpecializedStructVtable"><span
```

```

class="type">DBusGTypeSpecializedStructVtable</span></a>
*vtable</code></em>,

                                                                    <em>
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
flags</code></em>;
<a href="http://library.gnome.org/devel/glib/unstable/glib-
GVariant.html#GVariant"><span class="returnvalue">GVariant</span></a>
*
  <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-value-build-g-variant"
title="dbus_g_value_build_g_variant
()">dbus_g_value_build_g_variant</a>      (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>);
<span class="returnvalue">void</span>      <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-value-parse-g-
variant" title="dbus_g_value_parse_g_variant
()">dbus_g_value_parse_g_variant</a>      (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-
GVariant.html#GVariant"><span class="type">GVariant</span></a>
*variant</code></em>,

                                                                    <em>
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>;
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-BOOLEAN-ARRAY:CAPS"
title="DBUS_TYPE_G_BOOLEAN_ARRAY">DBUS_TYPE_G_BOOLEAN_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-UCHAR-ARRAY:CAPS"
title="DBUS_TYPE_G_UCHAR_ARRAY">DBUS_TYPE_G_UCHAR_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-UINT-ARRAY:CAPS"
title="DBUS_TYPE_G_UINT_ARRAY">DBUS_TYPE_G_UINT_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-INT-ARRAY:CAPS"
title="DBUS_TYPE_G_INT_ARRAY">DBUS_TYPE_G_INT_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-UINT64-ARRAY:CAPS"
title="DBUS_TYPE_G_UINT64_ARRAY">DBUS_TYPE_G_UINT64_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-INT64-ARRAY:CAPS"
title="DBUS_TYPE_G_INT64_ARRAY">DBUS_TYPE_G_INT64_ARRAY</a>
#define      <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-OBJECT-ARRAY:CAPS"
title="DBUS_TYPE_G_OBJECT_ARRAY">DBUS_TYPE_G_OBJECT_ARRAY</a>

```

```

#define          <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-STRING-STRING-HASHTABLE:CAPS"
title="DBUS_TYPE_G_STRING_STRING_HASHTABLE">DBUS_TYPE_G_STRING_STRING_
HASHTABLE</a>
typedef          <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGSignature"
title="DBusGSignature">DBusGSignature</a>;
#define          <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-SIGNATURE:CAPS"
title="DBUS_TYPE_G_SIGNATURE">DBUS_TYPE_G_SIGNATURE</a>
typedef          <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGObjectPath"
title="DBusGObjectPath">DBusGObjectPath</a>;
#define          <a class="link" href="dbus-glib-Specializable-
GType-System.html#DBUS-TYPE-G-OBJECT-PATH:CAPS"
title="DBUS_TYPE_G_OBJECT_PATH">DBUS_TYPE_G_OBJECT_PATH</a>
</pre>
</div>
<div class="refsect1">
<a name="dbus-glib-Specializable-GType-
System.description"></a><h2>Description</h2>
<p>
Specialized gtypes are basically a way to allow the definition of
recursive <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GTypes</span></a>. It
allows the definition of 'containers' which is
basically a user defined structure capable of holding other data, and
a
set of functions defining how to access that structure. Containers
come in
3 flavors: collections, maps and structs.
</p>
<p>
A collection is a container that holds an ordered set of items, all
of which must be the same type. (This is an <span
class="emphasis"><em>array</em></span>
in standard D-Bus terminology.) dbus-glib specialized collections can
be
<a href="http://library.gnome.org/devel/glib/unstable/glib-
Arrays.html#GArray"><span class="type">GArray</span></a> (for numeric
elements), <a href="http://library.gnome.org/devel/glib/unstable/glib-
Pointer-Arrays.html#GPtrArray"><span class="type">GPtrArray</span></a>
(for string, object or
boxed elements), <a
href="http://library.gnome.org/devel/glib/unstable/glib-Singly-Linked-
Lists.html#GSList"><span class="type">GSList</span></a> (for boxed
elements, not recommended), or a
user-defined type.
</p>
<p>
A map is a container that holds a set of key/value pairs.

```


The keys have one type, and the values another; the type of the keys must be a numeric or string-like type. (This is a dict (dictionary) or array of dict entry in standard D-Bus terminology.) dbus-glib specialized maps can be [GHashTable or a user-defined type.](http://library.gnome.org/devel/glib/unstable/glib-Hash-Tables.html#GHashTable)

</p>

<p>

A struct is a container that holds a fixed number of members, each member

having a predefined type. (This is a struct in standard D-Bus terminology.) It is analogous to the C <code>struct</code> keyword, but dbus-glib does not generally represent D-Bus structs in C structs. dbus-glib specialized structs can be [GValueArray or a user-defined type.](http://library.gnome.org/devel/gobject/unstable/gobject-Value-arrays.html#GValueArray)

</p>

<p>

A specialization is a GType detailing a particular container with particular types (a type specialization).

</p>

<p>

Functions are provided for constructing and manipulating specializations.

</p>

<p>

This documentation needs splitting into two pages, one for defining new containers and using existing containers. I expect most users to only do the latter. I also need to add some examples.

</p>

</div>

<div class="refsect1">

<h2>Details</h2>

<div class="refsect2">

<h3>DBusGTypeSpecializedCollectionIterator ()</h3>

<pre class="programlisting">void (*DBusGTypeSpecializedCollectionIterator)

(<em

class="parameter"><code>const <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-

```
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
```

<em

```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
```

<p>

A library-user-supplied function, called for each element in the collection when [dbus_g_type_collection_value_iterate \(\)](dbus-glib-Specializable-GType-System.html#dbus-g-type-collection-value-iterate "dbus_g_type_collection_value_iterate ()") `dbus_g_type_collection_value_iterate ()` is called.

</p>

```
<div class="variablelist"><table border="0">
```

```
<col align="left" valign="top">
```

```
<tbody>
```

```
<tr>
```

```
<td><p><span class="term"><em
```

```
class="parameter"><code>value</code></em>Â :</span></p></td>
```

```
<td>an element of the collection</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em
```

```
class="parameter"><code>user_data</code></em>Â :</span></p></td>
```

```
<td>the data supplied when calling
```

```
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-collection-value-iterate"
```

```
title="dbus_g_type_collection_value_iterate ()"><code
```

```
class="function">dbus_g_type_collection_value_iterate ()</code></a>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table></div>
```

```
</div>
```

```
<hr>
```

```
<div class="refsect2">
```

```
<a
```

```
name="DBusGTypeSpecializedMapIterator"></a><h3>DBusGTypeSpecializedMap
Iterator ()</h3>
```

```
<pre class="programlisting"><span class="returnvalue">void</span>
```

```
(*DBusGTypeSpecializedMapIterator) (<em class="parameter"><code>const
```

```
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
```

```
*key_val</code></em>,</pre>
```

<em

```
class="parameter"><code>const <a
```

```
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
```

```
*value_val</code></em>,</pre>
```

<em

```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
```

<p>

A library-user-supplied function, called for each key/value pair in the collection when [<code>dbus_g_type_map_value_iterate \(\)</code>](dbus-glib-Specializable-GType-System.html#dbus-g-type-map-value-iterate "dbus_g_type_map_value_iterate ()") is called.

```
<div class="variablelist"><table border="0">
```

```
<col align="left" valign="top">
```

```
<tbody>
```

```
<tr>
```

```
<td><p><span class="term"><em
class="parameter"><code>key_val</code></em>Â :</span></p></td>
```

```
<td>a key from the map</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em
class="parameter"><code>value_val</code></em>Â :</span></p></td>
```

```
<td>a value from the map</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
```

```
<td>the data supplied when calling
```

```
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-
g-type-map-value-iterate" title="dbus_g_type_map_value_iterate
()"><code class="function">dbus_g_type_map_value_iterate ()</code></a>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table></div>
```

```
</div>
```

```
<hr>
```

```
<div class="refsect2">
```

```
<a
name="DBusGTypeSpecializedAppendContext"></a><h3>DBusGTypeSpecializedA
ppendContext</h3>
```

```
<pre class="programlisting">typedef struct {
    /* public */
    GValue *val;
    GType specialization_type;
} DBusGTypeSpecializedAppendContext;
</pre>
```

<p>

A context for appending. There are more fields, which are private.

```

<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a>Â *<em
class="structfield"><code><a
name="DBusGTypeSpecializedAppendContext.val"></a>val</code></em>;</spa
n></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a> containing
the array to which you're appending</td>
</tr>
<tr>
<td><p><span class="term"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>Â <em
class="structfield"><code><a
name="DBusGTypeSpecializedAppendContext.specialization-
type"></a>specialization_type</code></em>;</span></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of the
array's elements</td>
</tr>
</tbody>
</table></div>

```

```

</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedConstructor"></a><h3>DBusGTypeSpecializedCon-
structor ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
(*DBusGTypeSpecializedConstructor) (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>);</pre>
<p>

```

```

</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em>Â :</span></p></td>
<td>a specialized boxed type</td>

```

```

</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>a new instance of <em class="parameter"><code>type</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedFreeFunc"></a><h3>DBusGTypeSpecializedFreeFu
nc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedFreeFunc)      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
val</code></em>);</pre>
<p>
Frees <em class="parameter"><code>val</code></em> according to <em
class="parameter"><code>type</code></em>. This is analogous to <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Boxed-
Types.html#GBoxedFreeFunc"><span
class="type">GBoxedFreeFunc</span></a>, but
can use information from <em class="parameter"><code>type</code></em>
(for instance to free the contents of a
container before freeing the actual container).
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em>Â :</span></p></td>
<td>a specialized boxed type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>val</code></em>Â :</span></p></td>
<td>an instance of <em class="parameter"><code>type</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>

```

```

<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedCopyFunc"></a><h3>DBusGTypeSpecializedCopyFu
nc ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
(*DBusGTypeSpecializedCopyFunc)      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
src</code></em>);</pre>
<p>
Copies <em class="parameter"><code>src</code></em> according to <em
class="parameter"><code>type</code></em>. This is analogous to <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Boxed-
Types.html#GBoxedCopyFunc"><span
class="type">GBoxedCopyFunc</span></a>, but
can use information from <em class="parameter"><code>type</code></em>
(for instance to copy each element of a
collection).
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em> :</span></p></td>
<td>a specialized boxed type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>src</code></em> :</span></p></td>
<td>an instance of <em class="parameter"><code>type</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>a deep copy of <em class="parameter"><code>src</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">

```

<h3>DBusGTypeSpecializedVtable</h3>

```
<pre class="programlisting">typedef struct {
    DBusGTypeSpecializedConstructor    constructor;
    DBusGTypeSpecializedFreeFunc      free_func;
    DBusGTypeSpecializedCopyFunc      copy_func;
    GDestroyNotify                    simple_free_func; /* for type-
independent freeing if possible */
} DBusGTypeSpecializedVtable;
</pre>
```

<p>A table of methods used to implement specialized container behaviour on

user-defined collections, maps and structs. Exactly one of <em class="parameter"><code>free_func</code> and <em class="parameter"><code>simple_free_func</code> must be implemented; the other must be <code class="literal">NULL</code>. <em class="parameter"><code>constructor</code> and <em class="parameter"><code>copy_func</code> must always be implemented.

</p>

<p>

There are additional members, which are reserved for future expansion and

must be <code class="literal">NULL</code>.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p>DBusGTypeSpecializedConstructorÂ

<td>returns a new, blank instance of the <em class="parameter"><code>type</code>

</td>

</tr>

<tr>

<td><p>DBusGTypeSpecializedFreeFuncÂ

```

<td>if not <a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>,
freees the <em class="parameter"><code>type</code></em> instance <em
class="parameter"><code>val</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCopyFunc"
title="DBusGTypeSpecializedCopyFunc ()"><span
class="type">DBusGTypeSpecializedCopyFunc</span></a>Â <em
class="structfield"><code><a name="DBusGTypeSpecializedVtable.copy-
func"></a>copy_func</code></em>;</span></p></td>
<td>returns a "deep copy" of the <em
class="parameter"><code>type</code></em> instance <em
class="parameter"><code>val</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Datasets.html#GDestroyNotify"><span
class="type">GDestroyNotify</span></a>Â <em
class="structfield"><code><a name="DBusGTypeSpecializedVtable.simple-
free-func"></a>simple_free_func</code></em>;</span></p></td>
<td>if not <a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>,
freees its argument</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedCollectionFixedAccessorFunc"></a><h3>DBusGTY
peSpecializedCollectionFixedAccessorFunc ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
(*DBusGTypeSpecializedCollectionFixedAccessorFunc)
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,

```



```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
*values</code></em>,</pre>
```

```
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
*len</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-collection-get-fixed"
title="dbus_g_type_collection_get_fixed ()"><code>
class="function">dbus_g_type_collection_get_fixed()</code></a> for a
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a> with
type
<em class="parameter"><code>type</code></em>, containing <em
class="parameter"><code>instance</code></em>.</p>
```

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em> :</span></p></td>
<td>a specialized collection boxed type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>instance</code></em> :</span></p></td>
<td>an instance of <em class="parameter"><code>type</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>values</code></em> :</span></p></td>
<td>used to return a pointer to the contents of <em
class="parameter"><code>instance</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>len</code></em> :</span></p></td>
<td>used to return the number of elements in <em
class="parameter"><code>instance</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
```

```

<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> on
success</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedCollectionIteratorFunc"></a><h3>DBusGTypeSpe
cializedCollectionIteratorFunc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedCollectionIteratorFunc)
</pre>
<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
<em class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedCollectionIterator"
title="DBusGTypeSpecializedCollectionIterator ()"><span
class="type">DBusGTypeSpecializedCollectionIterator</span></a>
iterator</code></em>,
<em class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-collection-value-iterate"
title="dbus_g_type_collection_value_iterate ()"><code
class="function">dbus_g_type_collection_value_iterate()</code></a> for
a <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a> with
type <em class="parameter"><code>type</code></em>, containing <em
class="parameter"><code>instance</code></em>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em> :</span></p></td>

```

```

<td>a specialized collection boxed type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>instance</code></em>Â :</span></p></td>
<td>an instance of <em class="parameter"><code>type</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iterator</code></em>Â :</span></p></td>
<td>the function to call for each element</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
<td>data to pass to <em class="parameter"><code>iterator</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedCollectionAppendFunc"></a><h3>DBusGTypeSpeci
alizedCollectionAppendFunc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedCollectionAppendFunc)
                                (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-specialized-collection-append"
title="dbus_g_type_specialized_collection_append ()"><code
class="function">dbus_g_type_specialized_collection_append()</code></a
>.
</p>
<p>
This function should use the <em
class="parameter"><code>val</code></em> and <em
class="parameter"><code>specialization_type</code></em> members of <em
class="parameter"><code>ctx</code></em>.

```

```

</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td>an appending context returned by <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-specialized-init-append"
title="dbus_g_type_specialized_init_append ()"><code
class="function">dbus_g_type_specialized_init_append()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>val</code></em>Â :</span></p></td>
<td>a value to copy into the collection</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedCollectionEndAppendFunc"></a><h3>DBusGTypeSp
ecializedCollectionEndAppendFunc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedCollectionEndAppendFunc)
                                (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-specialized-collection-end-append"
title="dbus_g_type_specialized_collection_end_append ()"><code
class="function">dbus_g_type_specialized_collection_end_append()</code
></a>.
</p>
<p>
This function should use the <em
class="parameter"><code>val</code></em> and <em
class="parameter"><code>specialization_type</code></em> members of <em
class="parameter"><code>ctx</code></em>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>

```

<td>an appending context returned by <code class="function">dbus_g_type_specialized_init_append()</code></td>

</tr></tbody>
</table></div>
</div>

<hr>

<div class="refsect2">

<h3>DBusGTypeSpecializedCollectionVtable</h3>

```
typedef struct {  
    DBusGTypeSpecializedVtable          base_vtable;  
    DBusGTypeSpecializedCollectionFixedAccessorFunc fixed_accessor;  
    DBusGTypeSpecializedCollectionIteratorFunc      iterator;  
    DBusGTypeSpecializedCollectionAppendFunc        append_func;  
    DBusGTypeSpecializedCollectionEndAppendFunc     end_append_func;  
} DBusGTypeSpecializedCollectionVtable;
```

</pre>

<p>

A table of methods used to implement specialized collection behaviour on user-defined types. At least <em class="parameter"><code>iterator</code> and <em class="parameter"><code>append_func</code> must be implemented.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p>DBusGTypeSpecializedVtableÂ <em class="structfield"><code>base_vtable</code>;</p></td>

<td>base methods shared between collections and other types</td>

</tr>

<tr>

<td><p>DBusGTypeSpecializedCollectionFixedAccessorFuncÂ <em class="structfield"><code>fixed_accessor</code>;</p></td>

```

<td>if not <a href="http://library.gnome.org/devel/glib/unstable/glib-
Standard-Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>,
provides access to the contents of this
collection, as documented for <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-collection-get-fixed"
title="dbus_g_type_collection_get_fixed ()"><code
class="function">dbus_g_type_collection_get_fixed()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionIteratorFunc"
title="DBusGTypeSpecializedCollectionIteratorFunc ()"><span
class="type">DBusGTypeSpecializedCollectionIteratorFunc</span></a>Â <e
m class="structfield"><code><a
name="DBusGTypeSpecializedCollectionVtable.iterator"></a>iterator</cod
e></em>;</span></p></td>
<td>iterates through the members of <em
class="parameter"><code>instance</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionAppendFunc"
title="DBusGTypeSpecializedCollectionAppendFunc ()"><span
class="type">DBusGTypeSpecializedCollectionAppendFunc</span></a>Â <em
class="structfield"><code><a
name="DBusGTypeSpecializedCollectionVtable.append-
func"></a>append_func</code></em>;</span></p></td>
<td>appends a new member to <em
class="parameter"><code>instance</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionEndAppendFunc"
title="DBusGTypeSpecializedCollectionEndAppendFunc ()"><span
class="type">DBusGTypeSpecializedCollectionEndAppendFunc</span></a>Â <
em class="structfield"><code><a
name="DBusGTypeSpecializedCollectionVtable.end-append-
func"></a>end_append_func</code></em>;</span></p></td>
<td>if not <a href="http://library.gnome.org/devel/glib/unstable/glib-
Standard-Macros.html#NULL:CAPS"><code class="literal">NULL</code></a>,
called after each group of calls to
the <em class="parameter"><code>append_func</code></em>
</td>
</tr>
</tbody>
</table></div>

```

```

</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedMapIteratorFunc"></a><h3>DBusGTypeSpecialize
dMapIteratorFunc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedMapIteratorFunc)
                                (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
                                <em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedMapIterator"
title="DBusGTypeSpecializedMapIterator ()"><span
class="type">DBusGTypeSpecializedMapIterator</span></a>
iterator</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-map-value-iterate"
title="dbus_g_type_map_value_iterate ()"><code
class="function">dbus_g_type_map_value_iterate()</code></a> for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a> with
type <em class="parameter"><code>type</code></em>, containing <em
class="parameter"><code>instance</code></em>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em>Â :</span></p></td>
<td>a specialized map boxed type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>instance</code></em>Â :</span></p></td>
<td>an instance of <em class="parameter"><code>type</code></em>
</td>

```

```

</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iterator</code></em>Â :</span></p></td>
<td>the function to call for each key/value pair</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
<td>data to pass to <em class="parameter"><code>iterator</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedMapAppendFunc"></a><h3>DBusGTypeSpecializedM
apAppendFunc ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
(*DBusGTypeSpecializedMapAppendFunc
                                (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*key</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-specialized-map-append"
title="dbus_g_type_specialized_map_append ()"><code>
class="function">dbus_g_type_specialized_map_append()</code></a>.
</p>
<p>
This function should use the <em
class="parameter"><code>val</code></em> and <em
class="parameter"><code>specialization_type</code></em> members of <em
class="parameter"><code>ctx</code></em>,
and replace any existing value with key equal to <em
class="parameter"><code>key</code></em>.
</p>

```



```

<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td>an appending context returned by <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-specialized-init-append"
title="dbus_g_type_specialized_init_append ()"><code
class="function">dbus_g_type_specialized_init_append()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>key</code></em>Â :</span></p></td>
<td>a key to add to the collection</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>val</code></em>Â :</span></p></td>
<td>a value to add to the collection</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedMapVtable"></a><h3>DBusGTypeSpecializedMapVt
able</h3>
<pre class="programlisting">typedef struct {
    DBusGTypeSpecializedVtable                base_vtable;
    DBusGTypeSpecializedMapIteratorFunc      iterator;
    DBusGTypeSpecializedMapAppendFunc       append_func;
} DBusGTypeSpecializedMapVtable;
</pre>
<p>
A table of methods used to implement specialized collection behaviour
on user-defined types. Both methods must be implemented.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedVtable"
title="DBusGTypeSpecializedVtable"><span
class="type">DBusGTypeSpecializedVtable</span></a>Â <em
class="structfield"><code><a name="DBusGTypeSpecializedMapVtable.base-
vtable"></a>base_vtable</code></em>;</span></p></td>
<td>base methods shared between maps and other types</td>
</tr>

```

```

<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedMapIteratorFunc"
title="DBusGTypeSpecializedMapIteratorFunc ()"><span
class="type">DBusGTypeSpecializedMapIteratorFunc</span></a>Â 

```

<em

```
class="parameter"><code><a  
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-  
values.html#GValue"><span class="type">GValue</span></a>  
*ret_value</code></em>;</pre>
```

<p>

Implements [dbus-g-type-struct-get-member](dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-get-member) `dbus_g_type_struct_get_member()` for a [GValue](http://library.gnome.org/devel/gobject/unstable/gobject-Generi-values.html#GValue) with type `type`, containing `instance`.

</p>
<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em class="parameter"><code>type</code> :</p></td>

<td>a specialized struct boxed type</td>

</tr>

<tr>

<td><p><em class="parameter"><code>instance</code> :</p></td>

<td>an instance of `type`,</td>

</tr>

<tr>

<td><p><em class="parameter"><code>member</code> :</p></td>

<td>the index of the member, starting from 0</td>

</tr>

<tr>

<td><p><em class="parameter"><code>ret_value</code> :</p></td>

<td>an initialized [GValue](http://library.gnome.org/devel/gobject/unstable/gobject-Generi-values.html#GValue) of the appropriate type for the given member of `type`</td>

</tr>

<tr>

<td><p>Returns :</p></td>

<td>

[TRUE](http://library.gnome.org/devel/glib/unstable/glib-Standard-Macros.html#TRUE:CAPS) on success</td>

</tr>

```

</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a
name="DBusGTypeSpecializedStructSetMember"></a><h3>DBusGTypeSpecialize
dStructSetMember ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
(*DBusGTypeSpecializedStructSetMember)
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
type</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
instance</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*new_value</code></em>;</pre>
<p>
Implements <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-struct-set-member"
title="dbus_g_type_struct_set_member ()"><code>
class="function">dbus_g_type_struct_set_member()</code></a> for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a> with type <em
class="parameter"><code>type</code></em>,
containing <em class="parameter"><code>instance</code></em>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>type</code></em> :</span></p></td>
<td>a specialized struct boxed type</td>
</tr>
<tr>

```

<p> instance : </p>	<p> an instance of <code>type</code> </p>
<p> member : </p>	<p> the index of the member, starting from 0 </p>
<p> new_value : </p>	<p> an initialized GValue of the appropriate type for the given member of <code>type</code> </p>
<p> Returns : </p>	<p> TRUE on success </p>

[DBusGTypeSpecializedStructVtable](#)

DBusGTypeSpecializedStructVtable

```

typedef struct {
    DBusGTypeSpecializedVtable          base_vtable;
    DBusGTypeSpecializedStructGetMember get_member;
    DBusGTypeSpecializedStructSetMember set_member;
} DBusGTypeSpecializedStructVtable;

```

A table of methods used to implement specialized collection behaviour on user-defined types. Both methods must be implemented.

<p> dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedVtable </p>

```

title="DBusGTypeSpecializedVtable"><span
class="type">DBusGTypeSpecializedVtable</span></a>Â <em
class="structfield"><code><a
name="DBusGTypeSpecializedStructVtable.base-
vtable"></a>base_vtable</code></em>;</span></p></td>
<td>base methods shared between maps and other types</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedStructGetMember"
title="DBusGTypeSpecializedStructGetMember ()"><span
class="type">DBusGTypeSpecializedStructGetMember</span></a>Â <em
class="structfield"><code><a
name="DBusGTypeSpecializedStructVtable.get-
member"></a>get_member</code></em>;</span></p></td>
<td>returns a member by its index</td>
</tr>
<tr>
<td><p><span class="term"><a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedStructSetMember"
title="DBusGTypeSpecializedStructSetMember ()"><span
class="type">DBusGTypeSpecializedStructSetMember</span></a>Â <em
class="structfield"><code><a
name="DBusGTypeSpecializedStructVtable.set-
member"></a>set_member</code></em>;</span></p></td>
<td>sets a member by its index</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-
collection"></a><h3>dbus_g_type_get_collection ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_collection      (<em class="parameter"><code>const
<span class="type">char</span> *container</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
specialization</code></em>);</pre>
<p>
Gets a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for a
particular collection instance,
creating the type if not already created.
</p>
<div class="variablelist"><table border="0">

```

```

<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>container</code></em>Â :</span></p></td>
<td>a string specifying a registered collection type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>specialization</code></em>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of
collection elements</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of that
instance</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-map"></a><h3>dbus_g_type_get_map ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_map                (<em class="parameter"><code>const
<span class="type">char</span> *container</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
key_specialization</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
value_specialization</code></em>);</pre>
<p>
Gets a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for a
particular map instance,
creating the type if not already created.
</p>
<div class="variablelist"><table border="0">

```

```

<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>container</code></em>Â :</span></p></td>
<td>a string specifying a registered map type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>key_specialization</code></em>Â :</span></p></
td>
<td>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of
keys</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value_specialization</code></em>Â :</span></p></
td>
<td>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of
values</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of that
instance</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-structv"></a><h3>dbus_g_type_get_structv
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_structv          (<em class="parameter"><code>const
<span class="type">char</span> *container</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
num_members</code></em>,<em
class="parameter"><code><a

```



```

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
*types</code></em>);</pre>
<p>
Gets a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for a
particular struct instance,
creating the type if not already created.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>container</code></em>Â :</span></p></td>
<td>a string specifying a registered struct type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>num_members</code></em>Â :</span></p></td>
<td>number of members in the struct</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>types</code></em>Â :</span></p></td>
<td>an array specufying a GType for each struct element</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of that
instance</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-struct"></a><h3>dbus_g_type_get_struct
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_struct          (<em class="parameter"><code>const
<span class="type">char</span> *container</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-

```

```

Information.html#GType"><span class="type">GType</span></a>
first_type</code></em>,
<em
class="parameter"><code>...</code></em>);</pre>
<p>
Varags methsod to get a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for a
particular struct instance,
creating the type if not already created.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>container</code></em>Â :</span></p></td>
<td>a string specifying a registered struct type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>first_type</code></em>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> for the
struct's first member</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>more GTypes for the struct's members, terminated by
G_TYPE_INVALID</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a> of that
instance</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-is-collection"></a><h3>dbus_g_type_is_collection
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_is_collection          (<em class="parameter"><code><a

```

```

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
Tests if a given GType is a collection.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em>Â :</span></p></td>
<td>a GType to test</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>true if the given GType is a collection</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-is-map"></a><h3>dbus_g_type_is_map ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_is_map          (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
Tests if a given GType is a map,
i.e. it was created with <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-map"
title="dbus_g_type_get_map () "><code
class="function">dbus_g_type_get_map()</code></a>.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em>Â :</span></p></td>
<td>a GType to test</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>true if the given GType is a map</td>
</tr>

```

```

</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-is-struct"></a><h3>dbus_g_type_is_struct ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_is_struct          (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
Tests if a given GType is a struct,
i.e. it was created with <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-struct"
title="dbus_g_type_get_struct ()"><code
class="function">dbus_g_type_get_struct()</code></a>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em> :</span></p></td>
<td>a GType to test</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>true if the given GType is a struct</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-collection-
specialization"></a><h3>dbus_g_type_get_collection_specialization
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_collection_specialization

                                (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>

```

Return the type of each element in collections of type `<em class="parameter">gtype`.
It is an error to call this function on a non-collection type.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter">gtype</em> :</span></p></td>
<td>a collection <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>, as
created by <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-get-collection"
title="dbus_g_type_get_collection ()"><code
class="function">dbus_g_type_get_collection()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>the element type for a given collection GType.</td>
</tr>
</tbody>
</table></div>
```

```
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-map-key-
specialization"></a><h3>dbus_g_type_get_map_key_specialization ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_map_key_specialization
```

```

<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
```

Return the type of the keys in maps of type `<em class="parameter">gtype`.
It is an error to call this function on a non-map type.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter">gtype</em> :</span></p></td>
```

```

<td>a map <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>, as
created by <a class="link" href="dbus-glib-Specializable-GType-
System.html#dbus-g-type-get-map" title="dbus_g_type_get_map ()"><code
class="function">dbus_g_type_get_map()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the key type for a given map <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-map-value-
specialization"></a><h3>dbus_g_type_get_map_value_specialization
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_map_value_specialization
                                (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
Return the type of the values in maps of type <em
class="parameter"><code>gtype</code></em>.
It is an error to call this function on a non-map type.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em>Â :</span></p></td>
<td>a map GType, as created by <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-map"
title="dbus_g_type_get_map ()"><code
class="function">dbus_g_type_get_map()</code></a>.</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the value type for a given map GType.</td>

```

```

</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-struct-member-
type"></a><h3>dbus_g_type_get_struct_member_type ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="returnvalue">GType</span></a>
dbus_g_type_get_struct_member_type (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>);</pre>
<p>
Get the type of a member of a specialized struct.
It is an error to call this function on a non-struct type.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em> :</span></p></td>
<td><a struct GType, as created with <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-struct"
title="dbus_g_type_get_struct ()"><code
class="function">dbus_g_type_get_struct()</code></a>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>member</code></em> :</span></p></td>
<td>the index of a struct member</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>the type for a given member of a struct <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>,
or <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Type-Information.html#G-TYPE-INVALID:CAPS"><code
class="literal">G_TYPE_INVALID</code></a> if <em
class="parameter"><code>member</code></em> >= <a class="link"
href="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-

```

```

struct-size" title="dbus_g_type_get_struct_size ()"><code
class="function">dbus_g_type_get_struct_size ()</code></a>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-get-struct-
size"></a><h3>dbus_g_type_get_struct_size ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="returnvalue">guint</span></a>
dbus_g_type_get_struct_size      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
gtype</code></em>);</pre>
<p>
Get the number of members in a specialized struct.
It is an error to call this function on a non-struct type.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>gtype</code></em> :</span></p></td>
<td>a struct GType, as created with <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-struct"
title="dbus_g_type_get_struct ()"><code
class="function">dbus_g_type_get_struct ()</code></a>.</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span> :</span></p></td>
<td>the number of members in a given struct <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-specialized-
construct"></a><h3>dbus_g_type_specialized_construct ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="returnvalue">gpointer</span></a>
dbus_g_type_specialized_construct  (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-

```


Information.html#GType">GType
gtype</code>);</pre>

<p>

Create an instance of a given specialized type.

The structure created and returned will depend on the container type of the

GType. E.g. If the given type was created by

`dbus_g_type_get_collection("GArray", G_TYPE_INT),`

then this will return a GArray with `element_size` of `sizeof(int)`

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>gtype</code> :</p></td>

<td>a specialized <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-

Information.html#GType">GType, as

created by <a class="link" href="dbus-glib-Specializable-GType-

System.html#dbus-g-type-get-collection"

title="dbus_g_type_get_collection ()"><code

class="function">dbus_g_type_get_collection()</code>,</td>

<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-

g-type-get-map" title="dbus_g_type_get_map ()"><code

class="function">dbus_g_type_get_map()</code> or <a class="link"

href="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-

struct" title="dbus_g_type_get_struct ()"><code

class="function">dbus_g_type_get_struct()</code>

</td>

</tr>

<tr>

<td><p><span

class="emphasis">Returns :</p></td>

<td>a pointer to a newly constructed instance of the given type.</td>

</tr>

</tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<a name="dbus-g-type-specialized-init-

append"><h3>dbus_g_type_specialized_init_append ()</h3>

<pre class="programlisting">void

dbus_g_type_specialized_init_append (<em class="parameter"><code><a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-

values.html#GValue">GValue

*value</code>,</td>

<em

class="parameter"><code><a class="link" href="dbus-glib-Specializable-

GType-System.html#DBusGTypeSpecializedAppendContext"

title="DBusGTypeSpecializedAppendContext"><span

```

class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);</pre>
<p>
Create a new context for adding elements to a collection or key/value
pairs
to a map. You generally don't need or want to use this..
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a> containing an
instance of specialized type</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td><a <a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a> in which to
return a new appending context.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-specialized-collection-
append"></a><h3>dbus_g_type_specialized_collection_append ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_specialized_collection_append
</pre>
<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a>
*elt</code></em>);</pre>
<p>
Appends a given element to the end of a collection.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">

```

```

<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td>a context created by <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-specialized-init-append"
title="dbus_g_type_specialized_init_append ()"><code
class="function">dbus_g_type_specialized_init_append()</code></a>
for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a> containing a
collection</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>elt</code></em>Â :</span></p></td>
<td>a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a> containing an
element to append to the collection</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-specialized-collection-end-
append"></a><h3>dbus_g_type_specialized_collection_end_append ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_specialized_collection_end_append
                                (<em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>);</pre>
<p>
Finish appending elements to a given collection
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody><tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td>a context created by <a class="link" href="dbus-glib-
Specializable-GType-System.html#dbus-g-type-specialized-init-append"
title="dbus_g_type_specialized_init_append ()"><code
class="function">dbus_g_type_specialized_init_append()</code></a>
for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue"><span class="type">GValue</span></a> containing a
collection</td>

```

```

</tr></tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-specialized-map-append"></a><h3>dbus_g_type_specialized_map_append ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_specialized_map_append (<em class="parameter"><code><a
class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedAppendContext"
title="DBusGTypeSpecializedAppendContext"><span
class="type">DBusGTypeSpecializedAppendContext</span></a>
*ctx</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*key</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*val</code></em>);</pre>
<p>
Inserts the given key/value pair into the map instance.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>ctx</code></em>Â :</span></p></td>
<td>a context created by <a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-type-specialized-init-append"
title="dbus_g_type_specialized_init_append ()"><code
class="function">dbus_g_type_specialized_init_append()</code></a>
for a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a> containing a
map</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>key</code></em>Â :</span></p></td>
<td>a GValue containing a key, whose contents will be stolen by <em
class="parameter"><code>ctx</code></em>
</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>val</code></em>Â :</span></p></td>

```

```

<td>a GValue containing a value, whose contents will be stolen by <em
class="parameter"><code>ctx</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-collection-get-fixed"></a><h3>dbus_g_type_collection_get_fixed ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_collection_get_fixed (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
*data_ret</code></em>,
<em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
*len_ret</code></em>);</pre>
<p>
Calling this function is likely to be a bad idea. Consider using
<a class="link" href="dbus-glib-Specializable-GType-System.html#dbus-g-type-collection-value-iterate"
title="dbus_g_type_collection_value_iterate ()"><code>
class="function">dbus_g_type_collection_value_iterate()</code></a>
instead.
</p>
<p>
On success, <em class="parameter"><code>data_ret</code></em> is a
pointer to the underlying data in a collection
of fixed-length fundamental types. Knowledge of the underlying data
model
of the collection is needed in order to use <em
class="parameter"><code>data_ret</code></em> correctly.
</p>
<p>
It is an error to call this function on a specialized type that is not
a
collection, or on a collection that does not have a <em
class="parameter"><code>fixed_accessor</code></em> in its
<a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionVtable"

```

```

title="DBusGTypeSpecializedCollectionVtable"><span
class="type">DBusGTypeSpecializedCollectionVtable</span></a>.
</p>
<p>
Specialized <a
href="http://library.gnome.org/devel/glib/unstable/glib-
Arrays.html#GArray"><span class="type">GArray</span></a>s are the only
types provided by dbus-glib that
can be used with this function; user-defined types might also work.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a href="http://library.gnome.org/devel/glib/unstable/glib-
Arrays.html#GArray"><span class="type">GArray</span></a> containing a boxed specialized collection
that has a <em class="parameter"><code>fixed_accessor</code></em> in
its vtable</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>data_ret</code></em>Â :</span></p></td>
<td><p>used to return a pointer to the fixed data, which must not be
modified (for instance, for a <a
href="http://library.gnome.org/devel/glib/unstable/glib-
Arrays.html#GArray"><span class="type">GArray</span></a> of <a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gint"><span class="type">gint</span></a>, this would point
to an array of <a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gint"><span class="type">gint</span></a></td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>len_ret</code></em>Â :</span></p></td>
<td><p>used to return the length (counting collection elements, not
bytes: in a <a
href="http://library.gnome.org/devel/glib/unstable/glib-
Arrays.html#GArray"><span class="type">GArray</span></a> containing
one <a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gint"><span class="type">gint</span></a>, this would be
1)</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> on
success</td>
</tr>

```

```

</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-collection-value-iterate"></a><h3>dbus_g_type_collection_value_iterate ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_collection_value_iterate
                                     (<em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                     <em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionIterator"
title="DBusGTypeSpecializedCollectionIterator ()"><span
class="type">DBusGTypeSpecializedCollectionIterator</span></a>
iterator</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
<p>
Calls the given function for each element of the collection.
The function is passed a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a> containing
the element and the given
<em class="parameter"><code>user_data</code></em> parameter. The
collection may not be modified while iterating over
it.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a> holding a
collection type.</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iterator</code></em>Â :</span></p></td>
<td><a function to call for each element</td>
</tr>
<tr>

```

```

<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
<td>user data to pass to the <em
class="parameter"><code>iterator</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-map-value-
iterate"></a><h3>dbus_g_type_map_value_iterate ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_map_value_iterate      (<em class="parameter"><code>const
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                     <em
class="parameter"><code><a class="link" href="dbus-glib-Specializable-
GType-System.html#DBusGTypeSpecializedMapIterator"
title="DBusGTypeSpecializedMapIterator ()"><span
class="type">DBusGTypeSpecializedMapIterator</span></a>
iterator</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gpointer"><span class="type">gpointer</span></a>
user_data</code></em>);</pre>
<p>
Calls the given function for each key/value pair of the map.
The function is passed two GValues containing the key/value pair and
the given
<em class="parameter"><code>user_data</code></em> parameter. The map
may not be modified while iterating over
it.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a> holding a
specialized map</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>iterator</code></em>Â :</span></p></td>
<td><a function to call for each element</td>

```



```

</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>user_data</code></em>Â :</span></p></td>
<td>user data to pass to the <em
class="parameter"><code>iterator</code></em>
</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-struct-get-
member"></a><h3>dbus_g_type_struct_get_member ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_struct_get_member      (<em class="parameter"><code>const
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
*dest</code></em>);</pre>
<p>
Fetches a given member of a given struct instance. <em
class="parameter"><code>dest</code></em> must be initialised
was the correct type for that member, e.g. as returned by
<em
class="parameter"><code>dbus_g_type_get_struct_member_type</code></em>
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a> containing a
struct instance</td>
</tr>
<tr>

```

```

<td><p><span class="term"><em
class="parameter"><code>member</code></em>Â :</span></p></td>
<td>the index of a given member</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>dest</code></em>Â :</span></p></td>
<td>an initialised <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a> in which to
return the struct member</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> if
successful</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-struct-set-
member"></a><h3>dbus_g_type_struct_set_member ()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_struct_set_member      (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
                                     <em
class="parameter"><code>const <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
c-values.html#GValue"><span class="type">GValue</span></a>
*src</code></em>);</pre>
<p>
Sets a given member of a struct to a new value. The type of <em
class="parameter"><code>src</code></em> must match
the existing type of <em class="parameter"><code>member</code></em>
member of the struct.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">

```

```

<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a> containing a
struct instance</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>member</code></em>Â :</span></p></td>
<td>the index of a given member</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>src</code></em>Â :</span></p></td>
<td><an <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a> containing
the new value for that struct member</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#TRUE:CAPS"><code class="literal">TRUE</code></a> if
successful</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-struct-get"></a><h3>dbus_g_type_struct_get
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_struct_get          (<em class="parameter"><code>const
<a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Generic-values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
member</code></em>,
                                <em
class="parameter"><code>...</code></em>);</pre>
<p>

```

Collects the selected values of this struct into the return locations provided.

```
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>value</code></em>Â :</span></p></td>
<td><a <a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a> containing a
struct instance</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>member</code></em>Â :</span></p></td>
<td>struct member to get</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>...</code></em>Â :</span></p></td>
<td>location in which to return the value of this member,
followed optionally by more member/return locations pairs, followed by
by <code class="literal">G_MAXUINT</code>
</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>
<a href="http://library.gnome.org/devel/glib/unstable/glib-Standard-
Macros.html#FALSE:CAPS"><code class="literal">FALSE</code></a> on
failure</td>
</tr>
</tbody>
</table></div>
</div>
```

```
<hr>
<div class="refsect2">
<a name="dbus-g-type-struct-set"></a><h3>dbus_g_type_struct_set
()</h3>
<pre class="programlisting"><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#gboolean"><span class="returnvalue">gboolean</span></a>
dbus_g_type_struct_set          (<em class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generi-
values.html#GValue"><span class="type">GValue</span></a>
*value</code></em>,
                                <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
```

Types.html#guint">guint
member</code>,</p></div>

<em

class="parameter"><code>...</code>);</pre>

<p>

Sets the selected members of the struct in <em

class="parameter"><code>value</code>.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>value</code>Â :</p></td>

<td>a GValue containing a struct instance</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>member</code>Â :</p></td>

<td>struct member to set</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>...</code>Â :</p></td>

<td>value for the first member, followed optionally by more member/value pairs, followed by <code

class="literal">G_MAXUINT</code>

</td>

</tr>

<tr>

<td><p>ReturnsÂ :</p></td>

<td>

<code class="literal">FALSE</code> on failure</td>

</tr>

</tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<h3>dbus_g_type_specialized_init ()</h3>

<pre class="programlisting">void

dbus_g_type_specialized_init (<em class="parameter"><code>void</code>);</pre>

<p>

Initialize dbus-glib specialized [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType)s.

In older versions of dbus-glib, it was necessary to do this before instantiating or registering any specialized type. It is now done automatically whenever necessary.

```
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-register-collection"><h3>dbus_g_type_register_collection ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_register_collection      (<em class="parameter"><code>const
<span class="type">char</span> *name</code></em>,
                                     <em
class="parameter"><code>const <a class="link" href="dbus-glib-Specializable-GType-System.html#DBusGTypeSpecializedCollectionVtable"
title="DBusGTypeSpecializedCollectionVtable"><span
class="type">DBusGTypeSpecializedCollectionVtable</span></a>
*vtable</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a>
flags</code></em>);</pre>
<p>
Defines a new collection container.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>name</code></em>Â :</span></p></td>
<td>The name of a new collection container</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>vtable</code></em>Â :</span></p></td>
<td>the vtable defining the new container</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>flags</code></em>Â :</span></p></td>
<td>As yet unused.</td>
</tr>
</tbody>
</table></div>
</div>
```

```

<hr>
<div class="refsect2">
<a name="dbus-g-type-register-map"></a><h3>dbus_g_type_register_map
()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_register_map          (<em class="parameter"><code>const
<span class="type">char</span> *name</code></em>,
                                     <em
class="parameter"><code>const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedMapVtable"
title="DBusGTypeSpecializedMapVtable"><span
class="type">DBusGTypeSpecializedMapVtable</span></a>
*vtable</code></em>,
                                     <em
class="parameter"><code><a
href="http://library.gnome.org/devel/glib/unstable/glib-Basic-
Types.html#guint"><span class="type">guint</span></a>
flags</code></em>);</pre>
<p>
Defines a new map container.
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>name</code></em>Â :</span></p></td>
<td>The name of a new map container</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>vtable</code></em>Â :</span></p></td>
<td>the vtable defining the new container</td>
</tr>
<tr>
<td><p><span class="term"><em
class="parameter"><code>flags</code></em>Â :</span></p></td>
<td>As yet unused.</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-map-peek-
vtable"></a><h3>dbus_g_type_map_peek_vtable ()</h3>
<pre class="programlisting">const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedMapVtable"
title="DBusGTypeSpecializedMapVtable"><span
class="returnvalue">DBusGTypeSpecializedMapVtable</span></a> *
dbus_g_type_map_peek_vtable

```

```

class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
map_type</code></em>);</pre>
<p>
Peek the vtable for a given map specialization
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>map_type</code></em>Â :</span></p></td>
<td>a gtype of a map specialization</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the vtable</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-collection-peek-
vtable"></a><h3>dbus_g_type_collection_peek_vtable ()</h3>
<pre class="programlisting">const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedCollectionVtable"
title="DBusGTypeSpecializedCollectionVtable"><span
class="returnvalue">DBusGTypeSpecializedCollectionVtable</span></a> *
dbus_g_type_collection_peek_vtable

```

```

class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
collection_type</code></em>);</pre>
<p>
Peek the vtable for a given collection specialization
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>collection_type</code></em>Â :</span></p></td>
<td>a gtype of a collection specialization</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>

```



```

<td>the vtable</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-struct-peek-
vtable"></a><h3>dbus_g_type_struct_peek_vtable ()</h3>
<pre class="programlisting">const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedStructVtable"
title="DBusGTypeSpecializedStructVtable"><span
class="returnvalue">DBusGTypeSpecializedStructVtable</span></a> *
dbus_g_type_struct_peek_vtable
                                     (<em
class="parameter"><code><a
href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#GType"><span class="type">GType</span></a>
struct_type</code></em>);</pre>
<p>
Peek the vtable for a given struct specialization
</p>
<div class="variablelist"><table border="0">
<col align="left" valign="top">
<tbody>
<tr>
<td><p><span class="term"><em
class="parameter"><code>struct_type</code></em>Â :</span></p></td>
<td>a gtype of a struct specialization</td>
</tr>
<tr>
<td><p><span class="term"><span
class="emphasis"><em>Returns</em></span>Â :</span></p></td>
<td>the vtable</td>
</tr>
</tbody>
</table></div>
</div>
<hr>
<div class="refsect2">
<a name="dbus-g-type-register-
struct"></a><h3>dbus_g_type_register_struct ()</h3>
<pre class="programlisting"><span class="returnvalue">void</span>
dbus_g_type_register_struct
      (<em class="parameter"><code>const
<span class="type">char</span> *name</code></em>,
                                     <em
class="parameter"><code>const <a class="link" href="dbus-glib-
Specializable-GType-System.html#DBusGTypeSpecializedStructVtable"
title="DBusGTypeSpecializedStructVtable"><span
class="type">DBusGTypeSpecializedStructVtable</span></a>
*vtable</code></em>,

```

<em

```
class="parameter"><code><a href="http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#guint"><span class="type">guint</span></a> flags</code></em>;</pre>
```

<p>

Defines a new struct container.

</p>

```
<div class="variablelist"><table border="0">
```

```
<col align="left" valign="top">
```

```
<tbody>
```

```
<tr>
```

```
<td><p><span class="term"><em
```

```
class="parameter"><code>name</code></em>Â :</span></p></td>
```

```
<td>The name of a new struct container</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em
```

```
class="parameter"><code>vtable</code></em>Â :</span></p></td>
```

```
<td>the vtable defining the new container</td>
```

```
</tr>
```

```
<tr>
```

```
<td><p><span class="term"><em
```

```
class="parameter"><code>flags</code></em>Â :</span></p></td>
```

```
<td>As yet unused.</td>
```

```
</tr>
```

```
</tbody>
```

```
</table></div>
```

```
</div>
```

```
<hr>
```

```
<div class="refsect2">
```

```
<a name="dbus-g-value-build-g-variant"></a><h3>dbus_g_value_build_g_variant ()</h3>
```

```
<pre class="programlisting"><a
```

```
href="http://library.gnome.org/devel/glib/unstable/glib-GVariant.html#GVariant"><span class="returnvalue">GVariant</span></a> *
```

```
dbus_g_value_build_g_variant (em
```

```
class="parameter"><code>const <a
```

```
href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-values.html#GValue"><span class="type">GValue</span></a>
```

```
*value</code></em>;</pre>
```

```
<p>
```

Recurses `value` and converts its contents to a

`GVariant`.

</p>

<p>

The value must either be a simple value (integer, string, boolean, object path etc.) or a specialized container registered with

`dbus_g_type_get_collection ()`

`dbus_g_type_get_collection()`, [dbus_g_type_get_map \(\)](dbus-glib-Specializable-GType-System.html#dbus-g-type-get-map "dbus_g_type_get_map ()") or [dbus_g_type_get_struct \(\)](dbus-glib-Specializable-GType-System.html#dbus-g-type-get-struct "dbus_g_type_get_struct ()"). Providing any other type is a programming error (including as a child type).

--

--

--

--

<p><code>value</code> :</p>

<p><code>value</code> :</p>

<p>a simple or specialized GValue to convert to</p>
--

<p>a GVariant</p>
--

--

--

--

<p><code>Returns</code> :</p>

<p><code>Returns</code> :</p>

<p>a new GVariant containing <code>value</code> with a floating reference</p>
--

--

--

--

--

--

<p><code>dbus_g_value_parse_g_variant ()</code></p>

<p><code>dbus_g_value_parse_g_variant ()</code></p>

<pre>void</pre>

<pre>dbus_g_value_parse_g_variant (GVariant</pre>

<pre>variant</pre>

<pre>*variant</pre>

`value`

<pre>value</pre>

<pre>value</pre>

<pre>value</pre>

Deserialize `variant` and put an equivalent dbus-glib data structure in

`value`.

</p>

<p>

It is an error if `variant` contains any

<http://library.gnome.org/devel/glib/unstable/glib-GVariant.html#GVariant> extensions not supported

by dbus-glib, including handles (file descriptor passing) and 'maybe'

types.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody>

<tr>

<td><p><em

class="parameter"><code>variant</code> :</p></td>

<td>GVariant

</td>

</tr>

<tr>

<td><p><em

class="parameter"><code>value</code> :</p></td>

<td>GValue

</td>

</tr>

</tbody>

</table></div>

</div>

<hr>

<div class="refsect2">

<h3>DBUS_TYPE_G_BOOLEAN_ARRAY</h3>

<pre class="programlisting">#define DBUS_TYPE_G_BOOLEAN_ARRAY

(dbus_g_type_get_collection ("GArray", G_TYPE_BOOLEAN))

</pre>

<p>

<p>

Expands to a function call returning the

<http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType> of a

<http://library.gnome.org/devel/glib/unstable/glib-Arrays.html#GArray> of

<http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gboolean>

(corresponding to the D-Bus signature "ab").

</p>

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-UCHAR-

ARRAY:CAPS"><h3>DBUS_TYPE_G_UCHAR_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_UCHAR_ARRAY
(dbus_g_type_get_collection ("GArray", G_TYPE_UCHAR))
</pre>
```

<p>

Expands to a function call returning the GType of a GArray of guchar (corresponding to the D-Bus signature "ay").

</p>

<p>

Note that this is not the same thing as a GByteArray! dbus-glib does not know about the GByteArray type.

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-UINT-

ARRAY:CAPS"><h3>DBUS_TYPE_G_UINT_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_UINT_ARRAY
(dbus_g_type_get_collection ("GArray", G_TYPE_UINT))
</pre>
```

<p>

Expands to a function call returning the GType of a GArray of guint (corresponding to the D-Bus signature "au").

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-INT-

ARRAY:CAPS"><h3>DBUS_TYPE_G_INT_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_INT_ARRAY
(dbus_g_type_get_collection ("GArray", G_TYPE_INT))
</pre>
```

<p>

Expands to a function call returning the [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType) of a [GArray](http://library.gnome.org/devel/glib/unstable/glib-Arrays.html#GArray) of a [gint](http://library.gnome.org/devel/glib/unstable/glib-Basic-Types.html#gint) (corresponding to the D-Bus signature "ai").

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-UINT64-

ARRAY:CAPS"><h3>DBUS_TYPE_G_UINT64_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_UINT64_ARRAY
(dbus_g_type_get_collection ("GArray", G_TYPE_UINT64))
```

</pre>

<p>

Expands to a function call returning the [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType) of a [GArray](http://library.gnome.org/devel/glib/unstable/glib-Arrays.html#GArray) of guint64 (corresponding to the D-Bus signature "at").

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-INT64-

ARRAY:CAPS"><h3>DBUS_TYPE_G_INT64_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_INT64_ARRAY
(dbus_g_type_get_collection ("GArray", G_TYPE_INT64))
```

</pre>

<p>

Expands to a function call returning the [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType) of a [GArray](http://library.gnome.org/devel/glib/unstable/glib-Arrays.html#GArray) of gint64 (corresponding to the D-Bus signature "ax").

</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-OBJECT-

ARRAY:CAPS"><h3>DBUS_TYPE_G_OBJECT_ARRAY</h3>

```
<pre class="programlisting">#define DBUS_TYPE_G_OBJECT_ARRAY
(dbus_g_type_get_collection ("GPtrArray", G_TYPE_OBJECT))
```

</pre>

<p>

Expands to a function call returning the [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType) of a [GPtrArray](http://library.gnome.org/devel/glib/unstable/glib-Pointer-Arrays.html#GPtrArray) of [GObject](http://library.gnome.org/devel/gobject/unstable/gobject-The-Base-Object-Type.html#GObject).

Use this type with caution: it can sometimes be used as a representation of arrays whose D-Bus signature is "ao" (transferred as an array of object paths), but the conventional type for such arrays is `(dbus_g_type_get_collection ("GPtrArray", DBUS_TYPE_G_OBJECT_PATH))`.

```
#define DBUS_TYPE_G_STRING_STRING_HASHTABLE (dbus_g_type_get_map ("GHashTable", G_TYPE_STRING, G_TYPE_STRING))
```

Expands to a function call returning the [GType](http://library.gnome.org/devel/gobject/unstable/gobject-Type-Information.html#GType) of a [GHashTable](http://library.gnome.org/devel/glib/unstable/glib-Hash-Tables.html#GHashTable) where the keys are strings and the values are also strings (corresponding to the D-Bus signature "a{ss}").

```
typedef gchar DBusGSignature;
```

A typedef for a string used to represent D-Bus signatures. Its `GType` is `DBUS_TYPE_G_SIGNATURE`, derived from `G_TYPE_BOXED`.

<p>
Prior to version 0.FIXME this was used as the type name of
<a class="link" href="dbus-glib-Specializable-GType-System.html#DBUS-
TYPE-G-SIGNATURE:CAPS" title="DBUS_TYPE_G_SIGNATURE"><code
class="literal">DBUS_TYPE_G_SIGNATURE</code>, but did not actually
exist as a typedef.

</p>
<p class="since">Since 0.FIXME</p>

</div>

<hr>

<div class="refsect2">

<a name="DBUS-TYPE-G-
SIGNATURE:CAPS"><h3>DBUS_TYPE_G_SIGNATURE</h3>

<pre class="programlisting">#define DBUS_TYPE_G_SIGNATURE
(dbus_g_signature_get_g_type ())

</pre>

<p>

The <a href="http://library.gnome.org/devel/gobject/unstable/gobject-
Type-Information.html#GType">GType of a
<a class="link" href="dbus-glib-Specializable-GType-
System.html#DBusGSignature" title="DBusGSignature"><span
class="type">DBusGSignature, which is a boxed type
containing a

D-Bus signature as a zero-terminated string. Signatures can be
copied with <a

href="http://library.gnome.org/devel/glib/unstable/glib-String-
Utility-Functions.html#g_strdup"><code

class="function">g_strdup()</code> and freed with <a

href="http://library.gnome.org/devel/glib/unstable/glib-Memory-
Allocation.html#g_free"><code class="function">g_free()</code>,
just like <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-STRING:CAPS"><code

class="literal">G_TYPE_STRING</code>,
but have a distinct boxed type to allow them to be distinguished when

stored in a <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Generic-
values.html#GValue">GValue.

</p>

<div class="variablelist"><table border="0">

<col align="left" valign="top">

<tbody><tr>

<td><p><span
class="emphasis">Returns :</p></td>

<td>a type derived from <a

href="http://library.gnome.org/devel/gobject/unstable/gobject-Type-
Information.html#G-TYPE-BOXED:CAPS"><code

class="literal">G_TYPE_BOXED</code>

</td>

</tr></tbody>

</table></div>

</div>

[DBusGObjectPath](#)

DBusGObjectPath

```
typedef gchar DBusGObjectPath;
```

A typedef for a string used to represent D-Bus object paths. Its GType is [DBusGObjectPath](#), derived from [GType](#).

Prior to version 0.FIXME this was used as the type name of [DBusGObjectPath](#), but did not actually exist as a typedef.

Since 0.FIXME

[DBUS_TYPE_G_OBJECT_PATH](#)

DBUS_TYPE_G_OBJECT_PATH

```
#define DBUS_TYPE_G_OBJECT_PATH  
(dbus_g_object_path_get_g_type ())
```

The [GType](#) of a [DBusGObjectPath](#), which is a boxed type containing a D-Bus object path as a zero-terminated string. Object paths can be copied with [g_strdup\(\)](#) and freed with [g_free\(\)](#), just like [G_TYPE_STRING](#), but have a distinct boxed type to allow them to be distinguished when

```

stored in a GValue.

```

<pre> Returns </pre>	<pre> : </pre>
<pre> a type derived from G_TYPE_BOXED </pre>	<pre> </pre>

```


```

Generated by GTK-Doc V1.18

File = dbus-glib-tool.c

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-glib-tool.c Tool used by apps using glib bindings
 *
 * Copyright (C) 2003, 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

```

```

*/

#include <config.h>
#include "dbus-gidl.h"
#include "dbus-gparser.h"
#include "dbus-gutils.h"
#include "dbus-glib-tool.h"
#include "dbus-binding-tool-glib.h"
#include <glib/gstdio.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/stat.h>
#include <string.h>
#include <time.h>

typedef enum {
    DBUS_BINDING_OUTPUT_NONE,
    DBUS_BINDING_OUTPUT_PRETTY,
    DBUS_BINDING_OUTPUT_GLIB_SERVER,
    DBUS_BINDING_OUTPUT_GLIB_CLIENT
} DbusBindingOutputMode;

static void
indent (int depth)
{
    depth *= 2; /* 2-space indent */

    while (depth > 0)
        {
            putc (' ', stdout);
            --depth;
        }
}

static void pretty_print (BaseInfo *base,
                          int depth);

static void
pretty_print_list (GSList *list,
                  int depth)
{
    GSList *tmp;

    tmp = list;
    while (tmp != NULL)
        {
            pretty_print (tmp->data, depth);
            tmp = tmp->next;
        }
}

```

```

static void
pretty_print (BaseInfo *base,
              int      depth)
{
    InfoType t;
    const char *name;

    t = base_info_get_type (base);
    name = base_info_get_name (base);

    indent (depth);

    switch (t)
    {
        case INFO_TYPE_NODE:
            {
                NodeInfo *n = (NodeInfo*) base;

                if (name == NULL)
                    printf (" {\n}");
                else
                    printf ("node \"%s\" {\n}", name);

                pretty_print_list (node_info_get_interfaces (n), depth + 1);
                pretty_print_list (node_info_get_nodes (n), depth + 1);

                indent (depth);
                printf ("}\n");
            }
            break;
        case INFO_TYPE_INTERFACE:
            {
                InterfaceInfo *i = (InterfaceInfo*) base;
                GSList *annotations, *elt;

                g_assert (name != NULL);

                printf ("interface \"%s\" {\n}", name);

                annotations = interface_info_get_annotations (i);
                for (elt = annotations; elt; elt = elt->next)
                {
                    const char *name = elt->data;
                    const char *value = interface_info_get_annotation (i, name);

                    printf (" (binding \"%s\": \"%s\") ",
                            name, value);
                }
                g_slist_free (annotations);

                pretty_print_list (interface_info_get_methods (i), depth + 1);
                pretty_print_list (interface_info_get_signals (i), depth + 1);
            }
    }
}

```

```

1);
    pretty_print_list (interface_info_get_properties (i), depth +

    indent (depth);
    printf ("}\n");
}
break;
case INFO_TYPE_METHOD:
{
    MethodInfo *m = (MethodInfo*) base;
    GSList *annotations, *elt;

    g_assert (name != NULL);

    annotations = method_info_get_annotations (m);
    printf ("method \"%s\" (\n", name);
    for (elt = annotations; elt; elt = elt->next)
    {
        const char *name = elt->data;
        const char *value = method_info_get_annotation (m, name);

        printf (" (annotation \"%s\": \"%s\") ",
            name, value);
    }
    g_slist_free (annotations);

    pretty_print_list (method_info_get_args (m), depth + 1);

    indent (depth);
    printf (")\n");
}
break;
case INFO_TYPE_SIGNAL:
{
    SignalInfo *s = (SignalInfo*) base;

    g_assert (name != NULL);

    printf ("signal \"%s\" (\n", name);

    pretty_print_list (signal_info_get_args (s), depth + 1);

    indent (depth);
    printf (")\n");
}
break;
case INFO_TYPE_PROPERTY:
{
    PropertyInfo *a = (PropertyInfo*) base;
    const char *pt = property_info_get_type (a);
    PropertyAccessFlags acc = property_info_get_access (a);

```

```

        printf ("%s%s %s",
                acc & PROPERTY_READ ? "read" : "",
                acc & PROPERTY_WRITE ? "write" : "",
                pt);
        if (name)
            printf (" %s\n", name);
        else
            printf ("\n");
    }
    break;
case INFO_TYPE_ARG:
    {
        ArgInfo *a = (ArgInfo*) base;
        const char *at = arg_info_get_type (a);
        ArgDirection d = arg_info_get_direction (a);

        printf ("%s %s",
                d == ARG_IN ? "in" : "out",
                at);
        if (name)
            printf (" %s\n", name);
        else
            printf ("\n");
    }
    break;
}
}

GQuark
dbus_binding_tool_error_quark (void)
{
    static GQuark quark = 0;
    if (!quark)
        quark = g_quark_from_static_string ("dbus_binding_tool_error");

    return quark;
}

static void warn (const char *fmt, ...) G_GNUC_PRINTF (1, 2);
static void warn_gerror (const char *prefix, GError *error);

static void
warn (const char *str, ...)
{
    va_list args;

    va_start (args, str);

    vfprintf (stderr, str, args);
    fputc ('\n', stderr);

    va_end (args);
}

```

```

}

static void
warn_gerror (const char *prefix, GError *error)
{
    warn ("%s: %s", prefix, error->message);
}

static void
usage (int ecode)
{
    fprintf (stderr, "dbus-binding-tool [--version] [--help]\n");
    fprintf (stderr, "dbus-binding-tool --mode=[pretty|glib-server|glib-
client] [--prefix=SYMBOL_PREFIX] [--ignore-unsupported] [--force] [--
output=FILE]\n");
    fprintf (stderr, "dbus-binding-tool --mode=glib-server --
prefix=SYMBOL_PREFIX [--ignore-unsupported] [--force] [--
output=FILE]\n");
    exit (ecode);
}

static void
version (void)
{
    printf ("D-BUS Binding Tool %s\n"
           "Copyright (C) 2003-2005 Red Hat, Inc.\n"
           "This is free software; see the source for copying
conditions.\n"
           "There is NO warranty; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.\n",
           VERSION);
    exit (0);
}

int
main (int argc, char **argv)
{
    const char *output_file;
    const char *prefix;
    char *output_file_tmp = NULL;
    int i;
    GSList *files;
    DBusBindingOutputMode outputmode;
    gboolean end_of_args;
    GSList *tmp;
    GIOChannel *channel = NULL;
    GError *error;
    time_t newest_src;
    struct stat srcbuf;
    struct stat targetbuf;
    gboolean force;
    gboolean ignore_unsupported;

```

```

gboolean has_prefix = FALSE;

g_type_init ();

outputmode = DBUS_BINDING_OUTPUT_NONE;
end_of_args = FALSE;
files = NULL;
output_file = NULL;
prefix = "";
ignore_unsupported = FALSE;
force = FALSE;
i = 1;
while (i < argc)
{
    const char *arg = argv[i];

    if (!end_of_args)
    {
        if (strcmp (arg, "--help") == 0 ||
            strcmp (arg, "-h") == 0 ||
            strcmp (arg, "-?") == 0)
            usage (0);
        else if (strcmp (arg, "--version") == 0)
            version ();
        else if (strcmp (arg, "--force") == 0)
            force = TRUE;
        else if (strncmp (arg, "--mode=", 7) == 0)
        {
            const char *mode = arg + 7;
            if (!strcmp (mode, "pretty"))
                outputmode = DBUS_BINDING_OUTPUT_PRETTY;
            else if (!strcmp (mode, "glib-server"))
                outputmode = DBUS_BINDING_OUTPUT_GLIB_SERVER;
            else if (!strcmp (mode, "glib-client"))
                outputmode = DBUS_BINDING_OUTPUT_GLIB_CLIENT;
            else
                usage (1);
        }
        else if (strcmp (arg, "--ignore-unsupported") == 0)
            ignore_unsupported = TRUE;
        else if (strncmp (arg, "--output=", 9) == 0)
        {
            output_file = arg + 9;
        }
        else if (strncmp (arg, "--prefix=", 9) == 0)
        {
            has_prefix = TRUE;
            prefix = arg + 9;
        }
        else if (arg[0] == '-' &&
                arg[1] == '-' &&
                arg[2] == '\\0')
    }
}

```



```

        end_of_args = TRUE;
    else if (arg[0] == '-')
    {
        usage (1);
    }
    else
    {
        files = g_slist_prepend (files, (char*) arg);
    }
}
else
    files = g_slist_prepend (files, (char*) arg);

++i;
}

if (outputmode == DBUS_BINDING_OUTPUT_GLIB_SERVER && !has_prefix)
    usage (1);

error = NULL;

files = g_slist_reverse (files);

if (output_file && !force)
{
    newest_src = 0;
    for (tmp = files; tmp != NULL; tmp = tmp->next)
    {
        const char *filename;

        filename = tmp->data;
        if (stat (filename, &srcbuf) < 0)
        {
            warn ("Couldn't stat %s: %s", filename, g_strerror
(errno));
            goto lose;
        }

        if (srcbuf.st_mtime > newest_src)
            newest_src = srcbuf.st_mtime;
    }

    if (stat (output_file, &targetbuf) > 0
        && targetbuf.st_mtime >= newest_src)
        exit (0);
}

if (output_file)
{
    output_file_tmp = g_strconcat (output_file, ".tmp", NULL);

```



```

        case DBUS_BINDING_OUTPUT_GLIB_CLIENT:
            if (!dbus_binding_tool_output_glib_client ((BaseInfo *)
node, channel, ignore_unsupported, &error))
            {
                warn_gerror ("Compilation failed", error);
                node_info_unref (node);
                goto lose;
            }
            break;
        case DBUS_BINDING_OUTPUT_NONE:
            break;
    }
}

if (node)
    node_info_unref (node);
}

if (g_io_channel_shutdown (channel, TRUE, &error) !=
G_IO_STATUS_NORMAL)
{
    warn_gerror ("Failed to shutdown IO channel", error);
    goto lose;
}

g_io_channel_unref (channel);
channel = NULL;

if (output_file)
{
    if (rename (output_file_tmp, output_file) < 0)
    {
        warn ("Failed to rename %s to %s: %s", output_file_tmp,
output_file,
            g_strerror (errno));
        goto lose;
    }
    g_free (output_file_tmp);
}

return 0;

lose:
/* ignore errors, we're already handling an error */
if (channel != NULL)
{
    (void) g_io_channel_shutdown (channel, FALSE, NULL);
    g_io_channel_unref (channel);
}

if (output_file_tmp != NULL)
    (void) g_remove (output_file_tmp);

```

```
g_free (output_file_tmp);
return 1;
}
```

File = dbus-glib-tool.h

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-glib-tool.h: Definitions used internally by binding tool
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifdef DBUS_BINDING_TOOL_H
#define DBUS_BINDING_TOOL_H

#include <glib.h>

typedef enum
{
    DBUS_BINDING_TOOL_ERROR_UNSUPPORTED_CONVERSION,
    DBUS_BINDING_TOOL_ERROR_INVALID_ANNOTATION
} DBusBindingToolError;

#define DBUS_BINDING_TOOL_ERROR dbus_binding_tool_error_quark ()

GQuark dbus_binding_tool_error_quark (void);

#endif
```

```
File = dbus-glib-unused.sgml
```

```
File = dbus-glib.c
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-glib.c General GLib binding stuff
 *
 * Copyright (C) 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

#include <config.h>
#include "dbus-glib.h"
#include "dbus-glib-lowlevel.h"
#include "dbus-gtest.h"
#include "dbus-gutils.h"
#include "dbus-gobject.h"
#include <string.h>

/**
 * SECTION:dbus-gconnection
 * @short_description: DBus Connection
 * @see_also: #DBusConnection
 * @stability: Stable
 *
 * A #DBusGConnection is a boxed type abstracting a DDBusConnection.
 */

/**
```

```

* DBusGConnection:
*
* A #DBusGConnection is a boxed type abstracting a DBusConnection
from
* libdbus.
*/

/**
* dbus_g_connection_flush:
* @connection: the #DBusGConnection to flush
*
* Blocks until outgoing calls and signal emissions have been sent.
*/
void
dbus_g_connection_flush (DBusGConnection *connection)
{
    dbus_connection_flush (DBUS_CONNECTION_FROM_G_CONNECTION
(connection));
}

/**
* dbus_g_connection_ref:
* @connection: the #DBusGConnection to ref
*
* Increment refcount on a #DBusGConnection
*
* Returns: the connection that was ref'd
*/
DBusGConnection*
dbus_g_connection_ref (DBusGConnection *connection)
{
    DBusConnection *c;

    c = DBUS_CONNECTION_FROM_G_CONNECTION (connection);
    dbus_connection_ref (c);
    return connection;
}

/**
* dbus_g_connection_unref:
* @connection: the connection to unref
*
* Decrement refcount on a #DBusGConnection
*/
void
dbus_g_connection_unref (DBusGConnection *connection)
{
    DBusConnection *c;

    c = DBUS_CONNECTION_FROM_G_CONNECTION (connection);
    dbus_connection_unref (c);
}

```

```

}

/**
 * SECTION:dbus-gmessage
 * @short_description: Dbus Message
 * @see_also: #DBusMessage
 * @stability: Stable
 *
 * A #DBusGMessage is a boxed type abstracting a DBusMessage.
 */

/**
 * DBusGMessage:
 *
 * A #DBusGMessage is a boxed type abstracting a DBusMessage from
 * libdbus.
 */

/**
 * dbus_g_message_ref:
 * @message: the message to ref
 *
 * Increment refcount on a #DBusGMessage
 *
 * Returns: the message that was ref'd
 */
DBusGMessage*
dbus_g_message_ref (DBusGMessage *message)
{
    DBusMessage *c;

    c = DBUS_MESSAGE_FROM_G_MESSAGE (message);
    dbus_message_ref (c);
    return message;
}

/**
 * dbus_g_message_unref:
 * @message: the message to unref
 *
 * Decrement refcount on a #DBusGMessage
 */
void
dbus_g_message_unref (DBusGMessage *message)
{
    DBusMessage *c;

    c = DBUS_MESSAGE_FROM_G_MESSAGE (message);
    dbus_message_unref (c);
}

```

```

/**
 * SECTION:dbus-gerror
 * @short_description: DBus GError
 * @see_also: #GError
 * @stability: Stable
 *
 * #DBusGError is the #GError used by DBus.
 */

/**
 * DBusGError:
 *
 * A #GError enumeration for the domain %DBUS_GERROR. The values'
meanings
 * can be found by looking at the comments for the corresponding
constants
 * in dbus-protocol.h.
 */

/**
 * DBUS_GERROR:
 *
 * Expands to a function call returning the error domain quark for
#DBusGError,
 * for use with #GError.
 */
GQuark
dbus_g_error_quark (void)
{
    static GQuark quark = 0;
    if (quark == 0)
        quark = g_quark_from_static_string ("dbus-glib-error-quark");
    return quark;
}

/**
 * dbus_g_error_has_name:
 * @error: the GError given from the remote method
 * @name: the D-BUS error name
 *
 * Determine whether D-BUS error name for a remote exception matches
 * the given name. This function is intended to be invoked on a
 * #GError returned from an invocation of a remote method, e.g. via
 * dbus_g_proxy_end_call(). It will silently return %FALSE for errors
 * which are not remote D-BUS exceptions (i.e. with a domain other
 * than %DBUS_GERROR or a code other than
 * %DBUS_GERROR_REMOTE_EXCEPTION).
 *
 * Returns: %TRUE if and only if the remote error has the given name
 */
gboolean
dbus_g_error_has_name (GError *error, const char *name)

```



```

{
    g_return_val_if_fail (error != NULL, FALSE);

    if (error->domain != DBUS_GERROR
        || error->code != DBUS_GERROR_REMOTE_EXCEPTION)
        return FALSE;

    return !strcmp (dbus_g_error_get_name (error), name);
}

/**
 * dbus_g_error_get_name:
 * @error: the #GError given from the remote method
 *
 * This function may only be invoked on a #GError returned from an
 * invocation of a remote method, e.g. via dbus_g_proxy_end_call().
 * Moreover, you must ensure that the error's domain is %DBUS_GERROR,
 * and the code is %DBUS_GERROR_REMOTE_EXCEPTION.
 *
 * Returns: the D-BUS name for a remote exception.
 */
const char *
dbus_g_error_get_name (GError *error)
{
    g_return_val_if_fail (error != NULL, NULL);
    g_return_val_if_fail (error->domain == DBUS_GERROR, NULL);
    g_return_val_if_fail (error->code == DBUS_GERROR_REMOTE_EXCEPTION,
        NULL);

    return error->message + strlen (error->message) + 1;
}

/**
 * DBUS_TYPE_CONNECTION:
 *
 * Expands to a function call returning a boxed #GType representing a
 * #DBusConnection pointer from libdbus. Not to be confused with
 * %DBUS_TYPE_G_CONNECTION, which you should usually use instead.
 *
 * Returns: the GLib type
 */
GType
dbus_connection_get_g_type (void)
{
    static GType our_type = 0;

    if (our_type == 0)
        our_type = g_boxed_type_register_static ("DBusConnection",
            (GBoxedCopyFunc)
dbus_connection_ref,
            (GBoxedFreeFunc)
dbus_connection_unref);
}

```

```

    return our_type;
}

/**
 * DBUS_TYPE_MESSAGE:
 *
 * Expands to a function call returning a boxed #GType representing a
 * #DBusMessage pointer from libdbus. Not to be confused with
 * %DBUS_TYPE_G_MESSAGE, which you should usually use instead.
 *
 * Returns: the GLib type
 */
GType
dbus_message_get_g_type (void)
{
    static GType our_type = 0;

    if (our_type == 0)
        our_type = g_boxed_type_register_static ("DBusMessage",
                                                (GBoxedCopyFunc)
dbus_message_ref,
                                                (GBoxedFreeFunc)
dbus_message_unref);

    return our_type;
}

/**
 * DBUS_TYPE_G_CONNECTION:
 *
 * Expands to a function call returning the boxed #GType of a
 #DBusGConnection.
 *
 * Returns: the GLib type
 */
GType
dbus_g_connection_get_g_type (void)
{
    static GType our_type = 0;

    if (our_type == 0)
        our_type = g_boxed_type_register_static ("DBusGConnection",
                                                (GBoxedCopyFunc)
dbus_g_connection_ref,
                                                (GBoxedFreeFunc)
dbus_g_connection_unref);

    return our_type;
}

```

```

/**
 * DBUS_TYPE_G_MESSAGE:
 *
 * Expands to a function call returning the boxed #GType of a
 #DBusGConnection.
 *
 * Returns: the GLib type
 */
GType
dbus_g_message_get_g_type (void)
{
    static GType our_type = 0;

    if (our_type == 0)
        our_type = g_boxed_type_register_static ("DBusGMessage",
                                                (GBoxedCopyFunc)
dbus_g_message_ref,
                                                (GBoxedFreeFunc)
dbus_g_message_unref);

    return our_type;
}

/**
 * SECTION:dbus-glib-lowlevel
 * @short_description: DBus lower level functions
 * @stability: Unstable
 *
 * These functions can be used to access lower level of DBus.
 */

/**
 * dbus_g_connection_get_connection:
 * @gconnection: a #DBusGConnection
 *
 * Get the #DBusConnection corresponding to this #DBusGConnection.
 * The return value does not have its refcount incremented.
 *
 * Returns: #DBusConnection
 */
DBusConnection*
dbus_g_connection_get_connection (DBusGConnection *gconnection)
{
    g_return_val_if_fail (gconnection, NULL);
    return DBUS_CONNECTION_FROM_G_CONNECTION (gconnection);
}

extern dbus_int32_t _dbus_gmain_connection_slot;

/**
 * dbus_connection_get_g_connection:
 * @connection: a #DBusConnection

```

```

*
* Get the #DBusGConnection corresponding to this #DBusConnection.
This only
* makes sense if the #DBusConnection was originally a
#DBusGConnection that was
* registered with the GLib main loop. The return value does not have
its
* refcount incremented.
*
* Returns: #DBusGConnection
*/
DBusGConnection*
dbus_connection_get_g_connection (DBusConnection *connection)
{
    g_return_val_if_fail (connection, NULL);
    g_return_val_if_fail (dbus_connection_get_data (connection,
_dbus_gmain_connection_slot), NULL);

    return DBUS_G_CONNECTION_FROM_CONNECTION (connection);
}

/**
* dbus_g_message_get_message:
* @gmessage: a #DBusGMessage
*
* Get the #DBusMessage corresponding to this #DBusGMessage.
* The return value does not have its refcount incremented.
*
* Returns: #DBusMessage
*/
DBusMessage*
dbus_g_message_get_message (DBusGMessage *gmessage)
{
    return DBUS_MESSAGE_FROM_G_MESSAGE (gmessage);
}

#ifdef DBUS_BUILD_TESTS

/*
* Unit test for general glib stuff
* Returns: %TRUE on success.
*/
gboolean
_dbus_glib_test (const char *test_data_dir)
{
    DBusError err;
    GError *gerror = NULL;

    dbus_error_init (&err);
    dbus_set_error_const (&err, DBUS_ERROR_NO_MEMORY, "Out of memory!");

```

```

dbus_set_g_error (&gerror, &err);
g_assert (gerror != NULL);
g_assert (gerror->domain == DBUS_GERROR);
g_assert (gerror->code == DBUS_GERROR_NO_MEMORY);
g_assert (!strcmp (gerror->message, "Out of memory!"));

dbus_error_init (&err);
g_clear_error (&gerror);

return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-glib.devhelpt2

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE book PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "">
<book xmlns="http://www.devhelpt.net/book" title="D-Bus GLib bindings -
Reference Manual" link="index.html" author="" name="dbus-glib"
version="2" language="c">
  <chapters>
    <sub name="Introduction" link="ch01.html"/>
    <sub name="API Reference" link="ch02.html">
      <sub name="API for using D-BUS with GLib" link="ch02.html#api-
overview"/>
      <sub name="DBusGConnection" link="dbus-glib-
DBusGConnection.html"/>
      <sub name="DBus GObject related functions" link="dbus-glib-DBus-
GObject-related-functions.html"/>
      <sub name="DBusGMessage" link="dbus-glib-DBusGMessage.html"/>
      <sub name="DBusGMethod" link="dbus-glib-DBusGMethod.html"/>
      <sub name="DBusGError" link="dbus-glib-DBusGError.html"/>
      <sub name="DBusGProxy" link="dbus-glib-DBusGProxy.html"/>
      <sub name="Specializable GType System" link="dbus-glib-
Specializable-GType-System.html"/>
      <sub name="DBus GLib low level" link="dbus-glib-DBus-GLib-low-
level.html"/>
    </sub>
    <sub name="Tools Reference" link="ch03.html">
      <sub name="dbus-binding-tool" link="dbus-binding-tool.html"/>
    </sub>
  </chapters>
  <functions>
    <keyword type="struct" name="DBusGConnection" link="dbus-glib-
DBusGConnection.html#DBusGConnection"/>
    <keyword type="macro" name="DBUS_TYPE_G_CONNECTION" link="dbus-
glib-DBusGConnection.html#DBUS-TYPE-G-CONNECTION:CAPS"/>
    <keyword type="function" name="dbus_g_bus_get ()" link="dbus-glib-
DBusGConnection.html#dbus-g-bus-get"/>

```

```
<keyword type="function" name="dbus_g_bus_get_private ()"
link="dbus-glib-DBusGConnection.html#dbus-g-bus-get-private"/>
<keyword type="function" name="dbus_g_thread_init ()" link="dbus-
glib-DBusGConnection.html#dbus-g-thread-init"/>
<keyword type="function" name="dbus_g_connection_open ()"
link="dbus-glib-DBusGConnection.html#dbus-g-connection-open"/>
<keyword type="function" name="dbus_g_connection_ref ()"
link="dbus-glib-DBusGConnection.html#dbus-g-connection-ref"/>
<keyword type="function" name="dbus_g_connection_unref ()"
link="dbus-glib-DBusGConnection.html#dbus-g-connection-unref"/>
<keyword type="function" name="dbus_g_connection_flush ()"
link="dbus-glib-DBusGConnection.html#dbus-g-connection-flush"/>
<keyword type="function" name="dbus_g_connection_get_connection
()" link="dbus-glib-DBusGConnection.html#dbus-g-connection-get-
connection"/>
<keyword type="function" name="dbus_g_connection_register_g_object
()" link="dbus-glib-DBusGConnection.html#dbus-g-connection-register-g-
object"/>
<keyword type="function"
name="dbus_g_connection_unregister_g_object ()" link="dbus-glib-
DBusGConnection.html#dbus-g-connection-unregister-g-object"/>
<keyword type="function" name="dbus_g_connection_lookup_g_object
()" link="dbus-glib-DBusGConnection.html#dbus-g-connection-lookup-g-
object"/>
<keyword type="struct" name="struct DBusGObjectInfo" link="dbus-
glib-DBus-GObject-related-functions.html#DBusGObjectInfo"/>
<keyword type="function" name="dbus_g_object_type_install_info ()"
link="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
type-install-info"/>
<keyword type="function"
name="dbus_g_object_type_register_shadow_property ()" link="dbus-glib-
DBus-GObject-related-functions.html#dbus-g-object-type-register-
shadow-property"/>
<keyword type="function" name="dbus_g_object_path_get_g_type ()"
link="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
path-get-g-type"/>
<keyword type="function" name="dbus_g_object_register_marshall
er ()" link="dbus-glib-DBus-GObject-related-functions.html#dbus-g-object-
register-marshaller"/>
<keyword type="function"
name="dbus_g_object_register_marshall_array ()" link="dbus-glib-
DBus-GObject-related-functions.html#dbus-g-object-register-marshaller-
array"/>
<keyword type="function"
name="dbus_glib_global_set_disable_legacy_property_access ()"
link="dbus-glib-DBus-GObject-related-functions.html#dbus-glib-global-
set-disable-legacy-property-access" since="0.88"/>
<keyword type="struct" name="DBusGMessage" link="dbus-glib-
DBusGMessage.html#DBusGMessage"/>
<keyword type="macro" name="DBUS_TYPE_G_MESSAGE" link="dbus-glib-
DBusGMessage.html#DBUS-TYPE-G-MESSAGE:CAPS"/>
```

```
<keyword type="function" name="dbus_g_message_ref ()" link="dbus-glib-DBusGMessage.html#dbus-g-message-ref"/>
<keyword type="function" name="dbus_g_message_unref ()" link="dbus-glib-DBusGMessage.html#dbus-g-message-unref"/>
<keyword type="function" name="dbus_g_message_get_message ()" link="dbus-glib-DBusGMessage.html#dbus-g-message-get-message"/>
<keyword type="struct" name="struct DBusGMethodInfo" link="dbus-glib-DBusGMethod.html#DBusGMethodInfo"/>
<keyword type="struct" name="DBusGMethodInvocation" link="dbus-glib-DBusGMethod.html#DBusGMethodInvocation"/>
<keyword type="function" name="dbus_g_method_get_sender ()" link="dbus-glib-DBusGMethod.html#dbus-g-method-get-sender"/>
<keyword type="function" name="dbus_g_method_get_reply ()" link="dbus-glib-DBusGMethod.html#dbus-g-method-get-reply"/>
<keyword type="function" name="dbus_g_method_send_reply ()" link="dbus-glib-DBusGMethod.html#dbus-g-method-send-reply"/>
<keyword type="function" name="dbus_g_method_return ()" link="dbus-glib-DBusGMethod.html#dbus-g-method-return"/>
<keyword type="function" name="dbus_g_method_return_error ()" link="dbus-glib-DBusGMethod.html#dbus-g-method-return-error"/>
<keyword type="enum" name="enum DBusGError" link="dbus-glib-DBusGError.html#DBusGError"/>
<keyword type="function" name="dbus_g_error_has_name ()" link="dbus-glib-DBusGError.html#dbus-g-error-has-name"/>
<keyword type="function" name="dbus_g_error_get_name ()" link="dbus-glib-DBusGError.html#dbus-g-error-get-name"/>
<keyword type="function" name="dbus_g_error_domain_register ()" link="dbus-glib-DBusGError.html#dbus-g-error-domain-register"/>
<keyword type="macro" name="DBUS_GERROR" link="dbus-glib-DBusGError.html#DBUS-GERROR:CAPS"/>
<keyword type="struct" name="struct DBusGProxy" link="dbus-glib-DBusGProxy.html#DBusGProxy"/>
<keyword type="struct" name="DBusGProxyCall" link="dbus-glib-DBusGProxy.html#DBusGProxyCall"/>
<keyword type="function" name="DBusGProxyCallNotify ()" link="dbus-glib-DBusGProxy.html#DBusGProxyCallNotify"/>
<keyword type="function" name="dbus_g_proxy_new_for_name ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name"/>
<keyword type="function" name="dbus_g_proxy_new_for_name_owner ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-name-owner"/>
<keyword type="function" name="dbus_g_proxy_new_from_proxy ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-from-proxy"/>
<keyword type="function" name="dbus_g_proxy_new_for_peer ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-new-for-peer"/>
<keyword type="function" name="dbus_g_proxy_set_interface ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-set-interface"/>
<keyword type="function" name="dbus_g_proxy_get_path ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-path"/>
<keyword type="function" name="dbus_g_proxy_get_bus_name ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-bus-name"/>
<keyword type="function" name="dbus_g_proxy_get_interface ()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-get-interface"/>
```

```
<keyword type="function" name="dbus_g_proxy_add_signal ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-add-signal"/>
<keyword type="function" name="dbus_g_proxy_connect_signal ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-connect-signal"/>
<keyword type="function" name="dbus_g_proxy_disconnect_signal ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-disconnect-signal"/>
<keyword type="function" name="dbus_g_proxy_send ()" link="dbus-
glib-DBusGProxy.html#dbus-g-proxy-send"/>
<keyword type="function" name="dbus_g_proxy_call ()" link="dbus-
glib-DBusGProxy.html#dbus-g-proxy-call"/>
<keyword type="function" name="dbus_g_proxy_call_with_timeout ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-call-with-timeout"/>
<keyword type="function" name="dbus_g_proxy_call_no_reply ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-call-no-reply"/>
<keyword type="function" name="dbus_g_proxy_begin_call ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-begin-call"/>
<keyword type="function"
name="dbus_g_proxy_begin_call_with_timeout ()" link="dbus-glib-
DBusGProxy.html#dbus-g-proxy-begin-call-with-timeout"/>
<keyword type="function" name="dbus_g_proxy_end_call ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-end-call"/>
<keyword type="function" name="dbus_g_proxy_cancel_call ()"
link="dbus-glib-DBusGProxy.html#dbus-g-proxy-cancel-call"/>
<keyword type="function" name="dbus_g_proxy_set_default_timeout
()" link="dbus-glib-DBusGProxy.html#dbus-g-proxy-set-default-timeout"
since="0.75"/>
<keyword type="function"
name="DBusGTypeSpecializedCollectionIterator ()" link="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionIterator"/>
<keyword type="function" name="DBusGTypeSpecializedMapIterator ()"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapIterator"/>
<keyword type="struct" name="DBusGTypeSpecializedAppendContext"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedAppendContext"/>
<keyword type="function" name="DBusGTypeSpecializedConstructor ()"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedConstructor"/>
<keyword type="function" name="DBusGTypeSpecializedFreeFunc ()"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedFreeFunc"/>
<keyword type="function" name="DBusGTypeSpecializedCopyFunc ()"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCopyFunc"/>
<keyword type="struct" name="DBusGTypeSpecializedVtable"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedVtable"/>
<keyword type="function"
name="DBusGTypeSpecializedCollectionFixedAccessorFunc ()" link="dbus-
glib-Specializable-GType-
System.html#DBusGTypeSpecializedCollectionFixedAccessorFunc"/>
```



```
<keyword type="function"
name="DBusGTypeSpecializedCollectionIteratorFunc ()" link="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionIteratorFunc"/>
<keyword type="function"
name="DBusGTypeSpecializedCollectionAppendFunc ()" link="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionAppendFunc"/>
<keyword type="function"
name="DBusGTypeSpecializedCollectionEndAppendFunc ()" link="dbus-glib-
Specializable-GType-
System.html#DBusGTypeSpecializedCollectionEndAppendFunc"/>
<keyword type="struct" name="DBusGTypeSpecializedCollectionVtable"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedCollectionVtable"/>
<keyword type="function" name="DBusGTypeSpecializedMapIteratorFunc
()" link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapIteratorFunc"/>
<keyword type="function" name="DBusGTypeSpecializedMapAppendFunc
()" link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapAppendFunc"/>
<keyword type="struct" name="DBusGTypeSpecializedMapVtable"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedMapVtable"/>
<keyword type="function" name="DBusGTypeSpecializedStructGetMember
()" link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedStructGetMember"/>
<keyword type="function" name="DBusGTypeSpecializedStructSetMember
()" link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedStructSetMember"/>
<keyword type="struct" name="DBusGTypeSpecializedStructVtable"
link="dbus-glib-Specializable-GType-
System.html#DBusGTypeSpecializedStructVtable"/>
<keyword type="function" name="dbus_g_type_get_collection ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-
collection"/>
<keyword type="function" name="dbus_g_type_get_map ()" link="dbus-
glib-Specializable-GType-System.html#dbus-g-type-get-map"/>
<keyword type="function" name="dbus_g_type_get_structv ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-
structv"/>
<keyword type="function" name="dbus_g_type_get_struct ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-
struct"/>
<keyword type="function" name="dbus_g_type_is_collection ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-is-
collection"/>
<keyword type="function" name="dbus_g_type_is_map ()" link="dbus-
glib-Specializable-GType-System.html#dbus-g-type-is-map"/>
<keyword type="function" name="dbus_g_type_is_struct ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-is-
struct"/>
```

```
<keyword type="function"
name="dbus_g_type_get_collection_specialization ()" link="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-collection-
specialization"/>
<keyword type="function"
name="dbus_g_type_get_map_key_specialization ()" link="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-map-key-
specialization"/>
<keyword type="function"
name="dbus_g_type_get_map_value_specialization ()" link="dbus-glib-
Specializable-GType-System.html#dbus-g-type-get-map-value-
specialization"/>
<keyword type="function" name="dbus_g_type_get_struct_member_type
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-
struct-member-type"/>
<keyword type="function" name="dbus_g_type_get_struct_size ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-get-
struct-size"/>
<keyword type="function" name="dbus_g_type_specialized_construct
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-construct"/>
<keyword type="function" name="dbus_g_type_specialized_init_append
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-init-append"/>
<keyword type="function"
name="dbus_g_type_specialized_collection_append ()" link="dbus-glib-
Specializable-GType-System.html#dbus-g-type-specialized-collection-
append"/>
<keyword type="function"
name="dbus_g_type_specialized_collection_end_append ()" link="dbus-
glib-Specializable-GType-System.html#dbus-g-type-specialized-
collection-end-append"/>
<keyword type="function" name="dbus_g_type_specialized_map_append
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-map-append"/>
<keyword type="function" name="dbus_g_type_collection_get_fixed
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
collection-get-fixed"/>
<keyword type="function"
name="dbus_g_type_collection_value_iterate ()" link="dbus-glib-
Specializable-GType-System.html#dbus-g-type-collection-value-
iterate"/>
<keyword type="function" name="dbus_g_type_map_value_iterate ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-map-value-
iterate"/>
<keyword type="function" name="dbus_g_type_struct_get_member ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-
get-member"/>
<keyword type="function" name="dbus_g_type_struct_set_member ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-
set-member"/>
```

```
<keyword type="function" name="dbus_g_type_struct_get ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-
get"/>
<keyword type="function" name="dbus_g_type_struct_set ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-
set"/>
<keyword type="function" name="dbus_g_type_specialized_init ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
specialized-init"/>
<keyword type="function" name="dbus_g_type_register_collection ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-
collection"/>
<keyword type="function" name="dbus_g_type_register_map ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-
map"/>
<keyword type="function" name="dbus_g_type_map_peek_vtable ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-map-peek-
vtable"/>
<keyword type="function" name="dbus_g_type_collection_peek_vtable
()" link="dbus-glib-Specializable-GType-System.html#dbus-g-type-
collection-peek-vtable"/>
<keyword type="function" name="dbus_g_type_struct_peek_vtable ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-struct-
peek-vtable"/>
<keyword type="function" name="dbus_g_type_register_struct ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-type-register-
struct"/>
<keyword type="function" name="dbus_g_value_build_g_variant ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-value-build-g-
variant"/>
<keyword type="function" name="dbus_g_value_parse_g_variant ()"
link="dbus-glib-Specializable-GType-System.html#dbus-g-value-parse-g-
variant"/>
<keyword type="macro" name="DBUS_TYPE_G_BOOLEAN_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-BOOLEAN-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_UCHAR_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-UCHAR-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_UINT_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-UINT-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_INT_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-INT-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_UINT64_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-UINT64-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_INT64_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-INT64-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_OBJECT_ARRAY" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-OBJECT-ARRAY:CAPS"/>
<keyword type="macro" name="DBUS_TYPE_G_STRING_STRING_HASHTABLE"
link="dbus-glib-Specializable-GType-System.html#DBUS-TYPE-G-STRING-
STRING-HASHTABLE:CAPS"/>
<keyword type="typedef" name="DBusGSignature" link="dbus-glib-
Specializable-GType-System.html#DBusGSignature" since="0.FIXME"/>
```

```

    <keyword type="macro" name="DBUS_TYPE_G_SIGNATURE" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-SIGNATURE:CAPS"/>
    <keyword type="typedef" name="DBusGObjectPath" link="dbus-glib-
Specializable-GType-System.html#DBusGObjectPath" since="0.FIXME"/>
    <keyword type="macro" name="DBUS_TYPE_G_OBJECT_PATH" link="dbus-
glib-Specializable-GType-System.html#DBUS-TYPE-G-OBJECT-PATH:CAPS"/>
    <keyword type="function" name="dbus_set_g_error ()" link="dbus-
glib-DBus-GLib-low-level.html#dbus-set-g-error"/>
    <keyword type="function" name="dbus_connection_setup_with_g_main
()" link="dbus-glib-DBus-GLib-low-level.html#dbus-connection-setup-
with-g-main"/>
    <keyword type="function" name="dbus_connection_get_g_connection
()" link="dbus-glib-DBus-GLib-low-level.html#dbus-connection-get-g-
connection"/>
    <keyword type="function" name="dbus_server_setup_with_g_main ()"
link="dbus-glib-DBus-GLib-low-level.html#dbus-server-setup-with-g-
main"/>
    <keyword type="macro" name="DBUS_TYPE_CONNECTION" link="dbus-glib-
DBus-GLib-low-level.html#DBUS-TYPE-CONNECTION:CAPS"/>
    <keyword type="macro" name="DBUS_TYPE_MESSAGE" link="dbus-glib-
DBus-GLib-low-level.html#DBUS-TYPE-MESSAGE:CAPS"/>
    <keyword type="" name="Options" link="dbus-binding-
tool.html#idp4772416"/>
</functions>
</book>

```

File = dbus-glib.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-glib.h GLib integration
 *
 * Copyright (C) 2002, 2003 CodeFactory AB
 * Copyright (C) 2003, 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software

```

```

* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_GLIB_H
#define DBUS_GLIB_H

#include <glib-object.h>
#include <dbus/dbus-shared.h>

G_BEGIN_DECLS

#define DBUS_INSIDE_DBUS_GLIB_H 1

/*
 * Convert to DBusConnection with dbus_g_connection_get_connection()
in dbus-glib-lowlevel.h
 */
typedef struct _DBusGConnection DBusGConnection;
/*
 * Convert to DBusMessage with dbus_g_message_get_message() in dbus-
glib-lowlevel.h
 */
typedef struct _DBusGMessage DBusGMessage;

#define DBUS_TYPE_G_CONNECTION (dbus_g_connection_get_g_type ())
#define DBUS_TYPE_G_MESSAGE (dbus_g_message_get_g_type ())
GType dbus_g_connection_get_g_type (void) G_GNUC_CONST;
GType dbus_g_message_get_g_type (void) G_GNUC_CONST;

DBusGConnection* dbus_g_connection_ref (DBusGConnection
*connection);
void dbus_g_connection_unref (DBusGConnection
*connection);
DBusGMessage* dbus_g_message_ref (DBusGMessage
*message);
void dbus_g_message_unref (DBusGMessage
*message);

void dbus_g_connection_flush (DBusGConnection
*connection);

GQuark dbus_g_error_quark (void);
#define DBUS_GERROR dbus_g_error_quark ()

typedef enum
{
DBUS_GERROR_FAILED,
DBUS_GERROR_NO_MEMORY,

```

```

DBUS_GERROR_SERVICE_UNKNOWN,
DBUS_GERROR_NAME_HAS_NO_OWNER,
DBUS_GERROR_NO_REPLY,
DBUS_GERROR_IO_ERROR,
DBUS_GERROR_BAD_ADDRESS,
DBUS_GERROR_NOT_SUPPORTED,
DBUS_GERROR_LIMITS_EXCEEDED,
DBUS_GERROR_ACCESS_DENIED,
DBUS_GERROR_AUTH_FAILED,
DBUS_GERROR_NO_SERVER,
DBUS_GERROR_TIMEOUT,
DBUS_GERROR_NO_NETWORK,
DBUS_GERROR_ADDRESS_IN_USE,
DBUS_GERROR_DISCONNECTED,
DBUS_GERROR_INVALID_ARGS,
DBUS_GERROR_FILE_NOT_FOUND,
DBUS_GERROR_FILE_EXISTS,
DBUS_GERROR_UNKNOWN_METHOD,
DBUS_GERROR_TIMED_OUT,
DBUS_GERROR_MATCH_RULE_NOT_FOUND,
DBUS_GERROR_MATCH_RULE_INVALID,
DBUS_GERROR_SPAWN_EXEC_FAILED,
DBUS_GERROR_SPAWN_FORK_FAILED,
DBUS_GERROR_SPAWN_CHILD_EXITED,
DBUS_GERROR_SPAWN_CHILD_SIGNALED,
DBUS_GERROR_SPAWN_FAILED,
DBUS_GERROR_UNIX_PROCESS_ID_UNKNOWN,
DBUS_GERROR_INVALID_SIGNATURE,
DBUS_GERROR_INVALID_FILE_CONTENT,
DBUS_GERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN,
DBUS_GERROR_REMOTE_EXCEPTION
} DBusGError;

gboolean          dbus_g_error_has_name      (GError          *error,
                                             const char      *name);
const char *      dbus_g_error_get_name      (GError          *error);

void              dbus_g_thread_init        (void);

DBusGConnection* dbus_g_connection_open     (const gchar     *address,
                                             GError          **error);
DBusGConnection* dbus_g_bus_get             (DBusBusType     type,
                                             GError          **error);
DBusGConnection* dbus_g_bus_get_private    (DBusBusType     type,
                                             GMainContext    *context,
                                             GError          **error);

typedef struct _DBusGMethodInfo DBusGMethodInfo;
typedef struct _DBusGObjectInfo DBusGObjectInfo;

/**

```



```

                                                    const char
*shadow_prop_name);

void          dbus_g_error_domain_register      (GQuark
domain,
                                                    const char *      default_iface,
                                                    GType             code_enum);

void          dbus_g_connection_register_g_object (DBusGConnection
*connection,
                                                    const char        *at_path,
                                                    GObject           *object);

void          dbus_g_connection_unregister_g_object (DBusGConnection
*connection,
                                                    GObject *object);

GObject *     dbus_g_connection_lookup_g_object (DBusGConnection
*connection,
                                                    const char        *at_path);

#ifdef DBUS_COMPILATION
#include "dbus/dbus-gtype-specialized.h"
#else
#include <dbus/dbus-gtype-specialized.h>
#endif

/* definitions for some basic array types */
#define DBUS_TYPE_G_BOOLEAN_ARRAY (dbus_g_type_get_collection
("GArray", G_TYPE_BOOLEAN))
#define DBUS_TYPE_G_UCHAR_ARRAY   (dbus_g_type_get_collection
("GArray", G_TYPE_UCHAR))
#define DBUS_TYPE_G_UINT_ARRAY    (dbus_g_type_get_collection
("GArray", G_TYPE_UINT))
#define DBUS_TYPE_G_INT_ARRAY     (dbus_g_type_get_collection
("GArray", G_TYPE_INT))
#define DBUS_TYPE_G_UINT64_ARRAY  (dbus_g_type_get_collection
("GArray", G_TYPE_UINT64))
#define DBUS_TYPE_G_INT64_ARRAY   (dbus_g_type_get_collection
("GArray", G_TYPE_INT64))
#define DBUS_TYPE_G_OBJECT_ARRAY  (dbus_g_type_get_collection
("GPtrArray", G_TYPE_OBJECT))

#define DBUS_TYPE_G_STRING_STRING_HASHTABLE (dbus_g_type_get_map
("GHashTable", G_TYPE_STRING, G_TYPE_STRING))

typedef gchar DBusGObjectPath;
GType          dbus_g_object_path_get_g_type      (void)
G_GNUC_CONST;
#define DBUS_TYPE_G_OBJECT_PATH (dbus_g_object_path_get_g_type ())

typedef gchar DBusGSignature;
GType          dbus_g_signature_get_g_type      (void)
G_GNUC_CONST;

```



```

const char
*iface);
DBusGProxy*      dbus_g_proxy_new_for_name_owner (DBusGConnection
*connection,
const char
*name,
const char
*path,
const char
*iface,
GError
**error);
DBusGProxy*      dbus_g_proxy_new_from_proxy (DBusGProxy
*proxy,
const char
*iface,
const char
*path);
DBusGProxy*      dbus_g_proxy_new_for_peer (DBusGConnection
*connection,
const char
*path,
const char
*iface);

void      dbus_g_proxy_set_interface (DBusGProxy
*proxy,
const char
*interface_name);
void      dbus_g_proxy_add_signal (DBusGProxy
*proxy,
const char      *signal_name,
GType          first_type,
...);

void      dbus_g_proxy_connect_signal (DBusGProxy
*proxy,
const char
*signal_name,
GCallback
handler,
void
*data,
GClosureNotify
free_data_func);
void      dbus_g_proxy_disconnect_signal (DBusGProxy
*proxy,
const char
*signal_name,
GCallback
handler,

```

```

void
*data);

gboolean          dbus_g_proxy_call          (DBusGProxy
*proxy,
                                     const char      *method,
                                     GError          **error,
                                     GType
first_arg_type,
                                     ...);

gboolean          dbus_g_proxy_call_with_timeout (DBusGProxy
*proxy,
                                     const char      *method,
                                     int timeout,
                                     GError          **error,
                                     GType
first_arg_type,
                                     ...);

void              dbus_g_proxy_call_no_reply  (DBusGProxy
*proxy,
*method,
                                     const char
first_arg_type,
                                     GType
                                     ...);

DBusGProxyCall *  dbus_g_proxy_begin_call    (DBusGProxy
*proxy,
                                     const char
*method,
                                     DBusGProxyCallNotify notify,
                                     gpointer          user_data,
                                     GDestroyNotify   destroy,
                                     GType
first_arg_type,
                                     ...);

DBusGProxyCall *  dbus_g_proxy_begin_call_with_timeout (DBusGProxy
*proxy,
                                     const char
*method,
                                     DBusGProxyCallNotify
notify,
                                     gpointer
user_data,
                                     GDestroyNotify
destroy,
                                     int timeout,
                                     GType
first_arg_type,
                                     ...);

```

```

void          dbus_g_proxy_set_default_timeout (DBusGProxy
*proxy,
                                                    int
timeout);

gboolean      dbus_g_proxy_end_call           (DBusGProxy
*proxy,
*call,
**error,
                                                    DBusGProxyCall
                                                    GError
                                                    GType
                                                    ...);
first_arg_type,
void          dbus_g_proxy_cancel_call       (DBusGProxy
*proxy,
*call);
                                                    DBusGProxyCall

const char*   dbus_g_proxy_get_path         (DBusGProxy
*proxy);

const char*   dbus_g_proxy_get_bus_name     (DBusGProxy
*proxy);

const char*   dbus_g_proxy_get_interface   (DBusGProxy
*proxy);

typedef struct _DBusGMethodInvocation DBusGMethodInvocation;

void          dbus_g_method_return          (DBusGMethodInvocation *context, ...);

void          dbus_g_method_return_error    (DBusGMethodInvocation *context, const GError *error);

/* Probably possible to replace this with a closure */
typedef struct {
    GCallback cb;
    gpointer userdata;
} DBusGAsyncData;

#undef DBUS_INSIDE_DBUS_GLIB_H

#include <dbus/dbus-gvalue-parse-variant.h>

G_END_DECLS

#endif /* DBUS_GLIB_H */

```

```
File = dbus-glib.types
```

```
File = dbus-gloader-expat.c
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gloader-expat.c  expat XML loader
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */
```

```
#include <config.h>
```

```
#include "dbus-gparser.h"
```

```
#include <expat.h>
```

```
#include <string.h>
```

```
static void*
expat_g_malloc (size_t sz)
{
    return g_malloc (sz);
}
```

```
static void*
expat_g_realloc (void *mem, size_t sz)
{
    return g_realloc (mem, sz);
}
```

```

static XML_Memory_Handling_Suite memsuite =
{
    expat_g_malloc,
    expat_g_realloc,
    g_free
};

/*
 * Context for Expat parser for introspection data.
 */
typedef struct
{
    Parser *parser;          /**< The parser for the introspection data */
    const char *filename;   /**< The filename being loaded */
    GString *content;       /**< The content of the current element */
    GError **error;        /**< Error return location */
    gboolean failed;        /**< True if parse has failed */
} ExpatParseContext;

static dbus_bool_t
process_content (ExpatParseContext *context)
{
    if (context->failed)
        return FALSE;

    if (context->content->len > 0)
    {
        if (!parser_content (context->parser,
                             context->content->str,
                             context->content->len,
                             context->error))
        {
            context->failed = TRUE;
            return FALSE;
        }
        g_string_set_size (context->content, 0);
    }

    return TRUE;
}

static void
expat_StartElementHandler (void          *userData,
                           const XML_Char *name,
                           const XML_Char **atts)
{
    ExpatParseContext *context = userData;
    int i;
    char **names;
    char **values;

    /* Expat seems to suck and can't abort the parse if we

```

```

    * throw an error. Expat 2.0 is supposed to fix this.
    */
    if (context->failed)
        return;

    if (!process_content (context))
        return;

    /* "atts" is key, value, key, value, NULL */
    for (i = 0; atts[i] != NULL; ++i)
        ; /* nothing */

    g_assert (i % 2 == 0);
    names = g_new0 (char *, i / 2 + 1);
    values = g_new0 (char *, i / 2 + 1);

    i = 0;
    while (atts[i] != NULL)
    {
        g_assert (i % 2 == 0);
        names [i / 2] = (char*) atts[i];
        values[i / 2] = (char*) atts[i+1];

        i += 2;
    }

    if (!parser_start_element (context->parser,
                               name,
                               (const char **) names,
                               (const char **) values,
                               context->error))
    {
        g_free (names);
        g_free (values);
        context->failed = TRUE;
        return;
    }

    g_free (names);
    g_free (values);
}

static void
expat_EndElementHandler (void *userData,
                        const XML_Char *name)
{
    ExpatParseContext *context = userData;

    if (!process_content (context))
        return;

    if (!parser_end_element (context->parser,

```

```

        name,
        context->error))
    {
        context->failed = TRUE;
        return;
    }
}

/* s is not 0 terminated. */
static void
expat_CharacterDataHandler (void          *userData,
                           const XML_Char *s,
                           int            len)
{
    ExpatParseContext *context = userData;

    if (context->failed)
        return;

    g_string_append_len (context->content,
                        s, len);
}

NodeInfo*
description_load_from_file (const char    *filename,
                           GError        **error)
{
    char *contents;
    gsize len;
    NodeInfo *nodes;

    contents = NULL;
    if (!g_file_get_contents (filename, &contents, &len, error))
        return NULL;

    nodes = description_load_from_string (contents, len, error);
    g_free (contents);

    return nodes;
}

NodeInfo*
description_load_from_string (const char *str,
                              int        len,
                              GError     **error)
{
    XML_Parser expat;
    ExpatParseContext context;
    NodeInfo *nodes;

    g_return_val_if_fail (error == NULL || *error == NULL, NULL);

```



```

if (len < 0)
    len = strlen (str);

expat = NULL;
context.parser = NULL;
context.error = error;
context.failed = FALSE;

expat = XML_ParserCreate_MM ("UTF-8", &memsuite, NULL);
if (expat == NULL)
    g_error ("No memory to create XML parser\n");

context.parser = parser_new ();
context.content = g_string_new (NULL);

XML_SetUserData (expat, &context);
XML_SetElementHandler (expat,
                        expat_StartElementHandler,
                        expat_EndElementHandler);
XML_SetCharacterDataHandler (expat,
                              expat_CharacterDataHandler);

if (!XML_Parse (expat, str, len, TRUE))
{
    if (context.error != NULL &&
        *context.error == NULL)
    {
        enum XML_Error e;

        e = XML_GetErrorCode (expat);
        if (e == XML_ERROR_NO_MEMORY)
            g_error ("Not enough memory to parse XML document");
        else
            g_set_error (error,
                        G_MARKUP_ERROR,
                        G_MARKUP_ERROR_PARSE,
                        "Error in D-BUS description XML, line %ld,
column %ld: %s\n",
                        (gulong)XML_GetCurrentLineNumber (expat),
                        (gulong)XML_GetCurrentColumnNumber (expat),
                        XML_ErrorString (e));
    }

    goto failed;
}

if (context.failed)
    goto failed;

if (!parser_finished (context.parser, error))
    goto failed;

```

```

XML_ParserFree (expat);
g_string_free (context.content, TRUE);

g_return_val_if_fail (error == NULL || *error == NULL, NULL);
nodes = parser_get_nodes (context.parser);
node_info_ref (nodes);
parser_unref (context.parser);
return nodes;

failed:
g_return_val_if_fail (error == NULL || *error != NULL, NULL);

g_string_free (context.content, TRUE);
if (expat)
    XML_ParserFree (expat);
if (context.parser)
    parser_unref (context.parser);
return NULL;
}

```

File = dbus-gmain.c

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gmain.c GLib main loop integration
 *
 * Copyright (C) 2002, 2003 CodeFactory AB
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

```

```

#include <config.h>
#include <dbus/dbus-glib.h>
#include <dbus/dbus-glib-lowlevel.h>
#include "dbus-gtest.h"
#include "dbus-gutils.h"
#include "dbus-gvalue.h"
#include "dbus-gobject.h"
#include "dbus-gvalue-utils.h"
#include "dbus-gsignature.h"
#include <string.h>

/*
 * DBusGMessageQueue:
 *
 * A GSource subclass for dispatching DBusConnection messages.
 * We need this on top of the IO handlers, because sometimes
 * there are messages to dispatch queued up but no IO pending.
 */
typedef struct
{
    GSource source; /**< the parent GSource */
    DBusConnection *connection; /**< the connection to dispatch */
} DBusGMessageQueue;

static gboolean message_queue_prepare (GSource      *source,
                                       gint          *timeout);
static gboolean message_queue_check   (GSource      *source);
static gboolean message_queue_dispatch (GSource      *source,
                                       GSourceFunc    callback,
                                       gpointer       user_data);

static const GSourceFuncs message_queue_funcs = {
    message_queue_prepare,
    message_queue_check,
    message_queue_dispatch,
    NULL
};

static gboolean
message_queue_prepare (GSource *source,
                      gint     *timeout)
{
    DBusConnection *connection = ((DBusGMessageQueue *)source)-
>connection;

    *timeout = -1;

    return (dbus_connection_get_dispatch_status (connection) ==
    DBUS_DISPATCH_DATA_REMAINS);
}

static gboolean

```

```

message_queue_check (GSource *source)
{
    return FALSE;
}

static gboolean
message_queue_dispatch (GSource      *source,
                       GSourceFunc  callback,
                       gpointer      user_data)
{
    DBusConnection *connection = ((DBusGMessageQueue *)source)-
>connection;

    dbus_connection_ref (connection);

    /* Only dispatch once - we don't want to starve other GSource */
    dbus_connection_dispatch (connection);

    dbus_connection_unref (connection);

    return TRUE;
}

typedef struct
{
    GMainContext *context;      /**< the main context */
    GSList *ios;               /**< all IOHandler */
    GSList *timeouts;         /**< all TimeoutHandler */
    DBusConnection *connection; /**< NULL if this is really for a server
not a connection */
    GSource *message_queue_source; /**< DBusGMessageQueue */
} ConnectionSetup;

typedef struct
{
    ConnectionSetup *cs;
    GSource *source;
    DBusWatch *watch;
} IOHandler;

typedef struct
{
    ConnectionSetup *cs;
    GSource *source;
    DBusTimeout *timeout;
} TimeoutHandler;

dbus_int32_t _dbus_gmain_connection_slot = -1;
static dbus_int32_t server_slot = -1;

static ConnectionSetup*

```

```

connection_setup_new (GMainContext *context,
                     DBusConnection *connection)
{
    ConnectionSetup *cs;

    cs = g_new0 (ConnectionSetup, 1);

    g_assert (context != NULL);

    cs->context = context;
    g_main_context_ref (cs->context);

    if (connection)
    {
        cs->connection = connection;

        cs->message_queue_source = g_source_new ((GSourceFuncs *)
&message_queue_funcs,
                                                sizeof
(DBusGMessageQueue));
        ((DBusGMessageQueue*)cs->message_queue_source)->connection =
connection;
        g_source_attach (cs->message_queue_source, cs->context);
    }

    return cs;
}

static void
io_handler_source_finalized (gpointer data)
{
    IOHandler *handler;

    handler = data;

    if (handler->watch)
        dbus_watch_set_data (handler->watch, NULL, NULL);

    g_free (handler);
}

static void
io_handler_destroy_source (void *data)
{
    IOHandler *handler;

    handler = data;

    if (handler->source)
    {
        GSource *source = handler->source;
        handler->source = NULL;
    }
}

```

```

        handler->cs->ios = g_slist_remove (handler->cs->ios, handler);
        g_source_destroy (source);
        g_source_unref (source);
    }
}

static void
io_handler_watch_freed (void *data)
{
    IOHandler *handler;

    handler = data;

    handler->watch = NULL;

    io_handler_destroy_source (handler);
}

static gboolean
io_handler_dispatch (GIOChannel *source,
                    GIOCondition condition,
                    gpointer data)
{
    IOHandler *handler;
    guint dbus_condition = 0;
    DBusConnection *connection;

    handler = data;

    connection = handler->cs->connection;

    if (connection)
        dbus_connection_ref (connection);

    if (condition & G_IO_IN)
        dbus_condition |= DBUS_WATCH_READABLE;
    if (condition & G_IO_OUT)
        dbus_condition |= DBUS_WATCH_WRITABLE;
    if (condition & G_IO_ERR)
        dbus_condition |= DBUS_WATCH_ERROR;
    if (condition & G_IO_HUP)
        dbus_condition |= DBUS_WATCH_HANGUP;

    /* Note that we don't touch the handler after this, because
     * dbus may have disabled the watch and thus killed the
     * handler.
     */
    dbus_watch_handle (handler->watch, dbus_condition);
    handler = NULL;

    if (connection)
        dbus_connection_unref (connection);
}

```

```

    return TRUE;
}

/* Attach the connection setup to the given watch, removing any
 * previously-attached connection setup.
 */
static void
connection_setup_add_watch (ConnectionSetup *cs,
                           DBusWatch      *watch)
{
    guint flags;
    GIOCondition condition;
    GIOChannel *channel;
    IOHandler *handler;

    if (!dbus_watch_get_enabled (watch))
        return;

    flags = dbus_watch_get_flags (watch);

    condition = G_IO_ERR | G_IO_HUP;
    if (flags & DBUS_WATCH_READABLE)
        condition |= G_IO_IN;
    if (flags & DBUS_WATCH_WRITABLE)
        condition |= G_IO_OUT;

    handler = g_new0 (IOHandler, 1);
    handler->cs = cs;
    handler->watch = watch;

    channel = g_io_channel_unix_new (dbus_watch_get_unix_fd (watch));

    handler->source = g_io_create_watch (channel, condition);
    g_source_set_callback (handler->source, (GSourceFunc)
io_handler_dispatch, handler,
                           io_handler_source_finalized);
    g_source_attach (handler->source, cs->context);

    cs->ios = g_slist_prepend (cs->ios, handler);

    dbus_watch_set_data (watch, handler, io_handler_watch_freed);
    g_io_channel_unref (channel);
}

static void
connection_setup_remove_watch (ConnectionSetup *cs,
                              DBusWatch      *watch)
{
    IOHandler *handler;

    handler = dbus_watch_get_data (watch);

```

```

    if (handler == NULL || handler->cs != cs)
        return;

    io_handler_destroy_source (handler);
}

static void
timeout_handler_source_finalized (gpointer data)
{
    TimeoutHandler *handler;

    handler = data;

    if (handler->timeout)
        dbus_timeout_set_data (handler->timeout, NULL, NULL);

    g_free (handler);
}

static void
timeout_handler_destroy_source (void *data)
{
    TimeoutHandler *handler;

    handler = data;

    if (handler->source)
    {
        GSource *source = handler->source;
        handler->source = NULL;
        handler->cs->timeouts = g_slist_remove (handler->cs->timeouts,
handler);
        g_source_destroy (source);
        g_source_unref (source);
    }
}

static void
timeout_handler_timeout_freed (void *data)
{
    TimeoutHandler *handler;

    handler = data;

    handler->timeout = NULL;

    timeout_handler_destroy_source (handler);
}

static gboolean
timeout_handler_dispatch (gpointer      data)

```



```

{
    TimeoutHandler *handler;

    handler = data;

    dbus_timeout_handle (handler->timeout);

    return TRUE;
}

static void
connection_setup_add_timeout (ConnectionSetup *cs,
                              DBusTimeout     *timeout)
{
    TimeoutHandler *handler;

    if (!dbus_timeout_get_enabled (timeout))
        return;

    g_assert (dbus_timeout_get_data (timeout) == NULL);

    handler = g_new0 (TimeoutHandler, 1);
    handler->cs = cs;
    handler->timeout = timeout;

    handler->source = g_timeout_source_new (dbus_timeout_get_interval
                                           (timeout));
    g_source_set_callback (handler->source, timeout_handler_dispatch,
                           handler,
                           timeout_handler_source_finalized);
    g_source_attach (handler->source, handler->cs->context);

    cs->timeouts = g_slist_prepend (cs->timeouts, handler);

    dbus_timeout_set_data (timeout, handler,
                           timeout_handler_timeout_freed);
}

static void
connection_setup_remove_timeout (ConnectionSetup *cs,
                                DBusTimeout     *timeout)
{
    TimeoutHandler *handler;

    handler = dbus_timeout_get_data (timeout);

    if (handler == NULL)
        return;

    timeout_handler_destroy_source (handler);
}

```

```

static void
connection_setup_free (ConnectionSetup *cs)
{
    while (cs->ios)
        io_handler_destroy_source (cs->ios->data);

    while (cs->timeouts)
        timeout_handler_destroy_source (cs->timeouts->data);

    if (cs->message_queue_source)
    {
        GSource *source;

        source = cs->message_queue_source;
        cs->message_queue_source = NULL;

        g_source_destroy (source);
        g_source_unref (source);
    }

    g_main_context_unref (cs->context);
    g_free (cs);
}

static dbus_bool_t
add_watch (DBusWatch *watch,
           gpointer data)
{
    ConnectionSetup *cs;

    cs = data;

    connection_setup_add_watch (cs, watch);

    return TRUE;
}

static void
remove_watch (DBusWatch *watch,
              gpointer data)
{
    ConnectionSetup *cs;

    cs = data;

    connection_setup_remove_watch (cs, watch);
}

static void
watch_toggled (DBusWatch *watch,
               void *data)
{

```

```

/* Because we just exit on OOM, enable/disable is
 * no different from add/remove
 */
if (dbus_watch_get_enabled (watch))
    add_watch (watch, data);
else
    remove_watch (watch, data);
}

static dbus_bool_t
add_timeout (DBusTimeout *timeout,
            void *data)
{
    ConnectionSetup *cs;

    cs = data;

    if (!dbus_timeout_get_enabled (timeout))
        return TRUE;

    connection_setup_add_timeout (cs, timeout);

    return TRUE;
}

static void
remove_timeout (DBusTimeout *timeout,
               void *data)
{
    ConnectionSetup *cs;

    cs = data;

    connection_setup_remove_timeout (cs, timeout);
}

static void
timeout_toggled (DBusTimeout *timeout,
                 void *data)
{
    /* Because we just exit on OOM, enable/disable is
     * no different from add/remove
     */
    if (dbus_timeout_get_enabled (timeout))
        add_timeout (timeout, data);
    else
        remove_timeout (timeout, data);
}

static void
wakeup_main (void *data)
{

```

```

    ConnectionSetup *cs = data;

    g_main_context_wakeup (cs->context);
}

/* Move to a new context */
static ConnectionSetup*
connection_setup_new_from_old (GMainContext *context,
                               ConnectionSetup *old)
{
    ConnectionSetup *cs;

    g_assert (old->context != context);

    cs = connection_setup_new (context, old->connection);

    while (old->ios != NULL)
    {
        IOHandler *handler = old->ios->data;

        connection_setup_add_watch (cs, handler->watch);
        /* The old handler will be removed from old->ios as a side-
effect */
    }

    while (old->timeouts != NULL)
    {
        TimeoutHandler *handler = old->timeouts->data;

        connection_setup_add_timeout (cs, handler->timeout);
    }

    return cs;
}

/**
 * dbus_connection_setup_with_g_main:
 * @connection: the connection
 * @context: the #GMainContext or %NULL for default context
 *
 * Sets the watch and timeout functions of a #DBusConnection
 * to integrate the connection with the GLib main loop.
 * Pass in %NULL for the #GMainContext unless you're
 * doing something specialized.
 *
 * If called twice for the same context, does nothing the second
 * time. If called once with context A and once with context B,
 * context B replaces context A as the context monitoring the
 * connection.
 */
void

```

```

dbus_connection_setup_with_g_main (DBusConnection *connection,
                                   GMainContext   *context)
{
    ConnectionSetup *old_setup;
    ConnectionSetup *cs;

    /* FIXME we never free the slot, so its refcount just keeps growing,
     * which is kind of broken.
     */
    dbus_connection_allocate_data_slot (&_dbus_gmain_connection_slot);
    if (_dbus_gmain_connection_slot < 0)
        goto nomem;

    if (context == NULL)
        context = g_main_context_default ();

    cs = NULL;

    old_setup = dbus_connection_get_data (connection,
    _dbus_gmain_connection_slot);
    if (old_setup != NULL)
    {
        if (old_setup->context == context)
            return; /* nothing to do */

        cs = connection_setup_new_from_old (context, old_setup);

        /* Nuke the old setup */
        dbus_connection_set_data (connection,
    _dbus_gmain_connection_slot, NULL, NULL);
        old_setup = NULL;
    }

    if (cs == NULL)
        cs = connection_setup_new (context, connection);

    if (!dbus_connection_set_data (connection,
    _dbus_gmain_connection_slot, cs,
    (DBusFreeFunction)connection_setup_free))
        goto nomem;

    if (!dbus_connection_set_watch_functions (connection,
    add_watch,
    remove_watch,
    watch_toggled,
    cs, NULL))
        goto nomem;

    if (!dbus_connection_set_timeout_functions (connection,
    add_timeout,
    remove_timeout,

```

```

                                                                    timeout_toggled,
                                                                    cs, NULL))

    goto nomem;

dbus_connection_set_wakeup_main_function (connection,
                                          wakeup_main,
                                          cs, NULL);

return;

nomem:
    g_error ("Not enough memory to set up DBusConnection for use with
GLib");
}

/**
 * dbus_server_setup_with_g_main:
 * @server: the server
 * @context: the #GMainContext or %NULL for default
 *
 * Sets the watch and timeout functions of a #DBusServer
 * to integrate the server with the GLib main loop.
 * In most cases the context argument should be %NULL.
 *
 * If called twice for the same context, does nothing the second
 * time. If called once with context A and once with context B,
 * context B replaces context A as the context monitoring the
 * connection.
 */
void
dbus_server_setup_with_g_main (DBusServer *server,
                              GMainContext *context)
{
    ConnectionSetup *old_setup;
    ConnectionSetup *cs;

    /* FIXME we never free the slot, so its refcount just keeps growing,
     * which is kind of broken.
     */
    dbus_server_allocate_data_slot (&server_slot);
    if (server_slot < 0)
        goto nomem;

    if (context == NULL)
        context = g_main_context_default ();

    cs = NULL;

    old_setup = dbus_server_get_data (server, server_slot);
    if (old_setup != NULL)
    {
        if (old_setup->context == context)

```

```

        return; /* nothing to do */

        cs = connection_setup_new_from_old (context, old_setup);

        /* Nuke the old setup */
        if (!dbus_server_set_data (server, server_slot, NULL, NULL))
            goto nomem;
        old_setup = NULL;
    }

    if (cs == NULL)
        cs = connection_setup_new (context, NULL);

    if (!dbus_server_set_data (server, server_slot, cs,
                              (DBusFreeFunction)connection_setup_free))
        goto nomem;

    if (!dbus_server_set_watch_functions (server,
                                         add_watch,
                                         remove_watch,
                                         watch_toggled,
                                         cs, NULL))

        goto nomem;

    if (!dbus_server_set_timeout_functions (server,
                                           add_timeout,
                                           remove_timeout,
                                           timeout_toggled,
                                           cs, NULL))

        goto nomem;

    return;

nomem:
    g_error ("Not enough memory to set up DBusServer for use with
GLib");
}

/**
 * dbus_g_connection_open:
 * @address: address of the connection to open
 * @error: address where an error can be returned.
 *
 * Returns a connection to the given address.
 *
 * (Internally, calls dbus_connection_open() then calls
 * dbus_connection_setup_with_g_main() on the result.)
 *
 * Returns: a DBusConnection
 */
DBusGConnection*
dbus_g_connection_open (const gchar *address,

```

```

                                GError      **error)
{
    DBusConnection *connection;
    DBusError derror;

    g_return_val_if_fail (error == NULL || *error == NULL, NULL);

    _dbus_g_value_types_init ();

    dbus_error_init (&derror);

    connection = dbus_connection_open (address, &derror);
    if (connection == NULL)
    {
        dbus_set_g_error (error, &derror);
        dbus_error_free (&derror);
        return NULL;
    }

    /* does nothing if it's already been done */
    dbus_connection_setup_with_g_main (connection, NULL);

    return DBUS_G_CONNECTION_FROM_CONNECTION (connection);
}

/**
 * dbus_g_bus_get:
 * @type: bus type
 * @error: address where an error can be returned.
 *
 * Returns a connection to the given bus. The connection is a global
variable
 * shared with other callers of this function.
 *
 * (Internally, calls dbus_bus_get() then calls
 * dbus_connection_setup_with_g_main() on the result.)
 *
 * Returns: a DBusConnection
 */
DBusGConnection*
dbus_g_bus_get (DBusBusType      type,
                GError          **error)
{
    DBusConnection *connection;
    DBusError derror;

    g_return_val_if_fail (error == NULL || *error == NULL, NULL);

    _dbus_g_value_types_init ();

    dbus_error_init (&derror);

```



```

connection = dbus_bus_get (type, &derror);
if (connection == NULL)
{
    dbus_set_g_error (error, &derror);
    dbus_error_free (&derror);
    return NULL;
}

/* does nothing if it's already been done */
dbus_connection_setup_with_g_main (connection, NULL);

return DBUS_G_CONNECTION_FROM_CONNECTION (connection);
}

/**
 * dbus_g_bus_get_private:
 * @type: bus type
 * @context: Mainloop context to attach to
 * @error: address where an error can be returned.
 *
 * Returns a connection to the given bus. The connection will be a
private
 * non-shared connection and should be closed when usage is complete.
 *
 * Internally this function calls dbus_bus_get_private() then calls
 * dbus_connection_setup_with_g_main() on the result; see the
documentation
 * of the former function for more information on private connections.
 *
 * Returns: a DBusConnection
 */
DBusGConnection*
dbus_g_bus_get_private (DBusBusType      type,
                       GMainContext    *context,
                       GError          **error)
{
    DBusConnection *connection;
    DBusError derror;

    g_return_val_if_fail (error == NULL || *error == NULL, NULL);

    _dbus_g_value_types_init ();

    dbus_error_init (&derror);

    connection = dbus_bus_get_private (type, &derror);
    if (connection == NULL)
    {
        dbus_set_g_error (error, &derror);
        dbus_error_free (&derror);
        return NULL;
    }
}

```

```

    /* does nothing if it's already been done */
    dbus_connection_setup_with_g_main (connection, context);

    return DBUS_G_CONNECTION_FROM_CONNECTION (connection);
}

#ifdef DBUS_BUILD_TESTS

/*
 * Unit test for GLib main loop integration
 * Returns: %TRUE on success.
 */
gboolean
_dbus_gmain_test (const char *test_data_dir)
{
    GType type;
    GType rectype;

    g_type_init ();
    _dbus_g_value_types_init ();

    rectype = dbus_g_type_get_collection ("GArray", G_TYPE_UINT);
    g_assert (rectype != G_TYPE_INVALID);
    g_assert (!strcmp (g_type_name (rectype), "GArray_guint_"));

    type = _dbus_gtype_from_signature ("au", TRUE);
    g_assert (type == rectype);

    rectype = dbus_g_type_get_map ("GHashTable", G_TYPE_STRING,
G_TYPE_STRING);
    g_assert (rectype != G_TYPE_INVALID);
    g_assert (!strcmp (g_type_name (rectype),
"GHashTable_gchararray+gchararray_"));

    type = _dbus_gtype_from_signature ("a{ss}", TRUE);
    g_assert (type == rectype);

    type = _dbus_gtype_from_signature ("o", FALSE);
    g_assert (type == DBUS_TYPE_G_OBJECT_PATH);
    type = _dbus_gtype_from_signature ("o", TRUE);
    g_assert (type == DBUS_TYPE_G_OBJECT_PATH);

    type = _dbus_gtype_from_signature ("g", TRUE);
    g_assert (type == DBUS_TYPE_G_SIGNATURE);

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-gmarshal.c

```
#include <config.h>
```

```
#include "dbus-gmarshal.h"
```

```
#include <glib-object.h>
```

```
#ifndef G_ENABLE_DEBUG
```

```
#define g_marshal_value_peek_boolean(v) g_value_get_boolean (v)
```

```
#define g_marshal_value_peek_char(v) g_value_get_char (v)
```

```
#define g_marshal_value_peek_uchar(v) g_value_get_uchar (v)
```

```
#define g_marshal_value_peek_int(v) g_value_get_int (v)
```

```
#define g_marshal_value_peek_uint(v) g_value_get_uint (v)
```

```
#define g_marshal_value_peek_long(v) g_value_get_long (v)
```

```
#define g_marshal_value_peek_ulong(v) g_value_get_ulong (v)
```

```
#define g_marshal_value_peek_int64(v) g_value_get_int64 (v)
```

```
#define g_marshal_value_peek_uint64(v) g_value_get_uint64 (v)
```

```
#define g_marshal_value_peek_enum(v) g_value_get_enum (v)
```

```
#define g_marshal_value_peek_flags(v) g_value_get_flags (v)
```

```
#define g_marshal_value_peek_float(v) g_value_get_float (v)
```

```
#define g_marshal_value_peek_double(v) g_value_get_double (v)
```

```
#define g_marshal_value_peek_string(v) (char*) g_value_get_string
```

```
(v)
```

```
#define g_marshal_value_peek_param(v) g_value_get_param (v)
```

```
#define g_marshal_value_peek_boxed(v) g_value_get_boxed (v)
```

```
#define g_marshal_value_peek_pointer(v) g_value_get_pointer (v)
```

```
#define g_marshal_value_peek_object(v) g_value_get_object (v)
```

```
#define g_marshal_value_peek_variant(v) g_value_get_variant (v)
```

```
/* !G_ENABLE_DEBUG */  
/* WARNING: This code accesses GValues directly, which is UNSUPPORTED  
API.
```

```
 * Do not access GValues directly in your code. Instead, use  
the
```

```
 * g_value_get_*() functions
```

```
 */
```

```
#define g_marshal_value_peek_boolean(v) (v)->data[0].v_int
```

```
#define g_marshal_value_peek_char(v) (v)->data[0].v_int
```

```
#define g_marshal_value_peek_uchar(v) (v)->data[0].v_uint
```

```
#define g_marshal_value_peek_int(v) (v)->data[0].v_int
```

```
#define g_marshal_value_peek_uint(v) (v)->data[0].v_uint
```

```
#define g_marshal_value_peek_long(v) (v)->data[0].v_long
```

```
#define g_marshal_value_peek_ulong(v) (v)->data[0].v_ulong
```

```
#define g_marshal_value_peek_int64(v) (v)->data[0].v_int64
```

```
#define g_marshal_value_peek_uint64(v) (v)->data[0].v_uint64
```

```
#define g_marshal_value_peek_enum(v) (v)->data[0].v_long
```

```
#define g_marshal_value_peek_flags(v) (v)->data[0].v_ulong
```

```
#define g_marshal_value_peek_float(v) (v)->data[0].v_float
```

```
#define g_marshal_value_peek_double(v) (v)->data[0].v_double
```

```
#define g_marshal_value_peek_string(v) (v)->data[0].v_pointer
```

```
#define g_marshal_value_peek_param(v) (v)->data[0].v_pointer
```

```

#define g_marshal_value_peek_boxed(v)      (v)->data[0].v_pointer
#define g_marshal_value_peek_pointer(v)    (v)->data[0].v_pointer
#define g_marshal_value_peek_object(v)     (v)->data[0].v_pointer
#define g_marshal_value_peek_variant(v)    (v)->data[0].v_pointer
#endif /* !G_ENABLE_DEBUG */

/* NONE:STRING,STRING,STRING (dbus-gmarshal.list:1) */
void
_dbus_g_marshal_VOID__STRING_STRING_STRING (GClosure      *closure,
                                             GValue         *return_value,
                                             G_GNUC_UNUSED,
                                             guint,
                                             n_param_values,
                                             const GValue,
                                             *param_values,
                                             gpointer,
                                             invocation_hint G_GNUC_UNUSED,
                                             gpointer,
                                             marshal_data)
{
    typedef void (*GMarshalFunc_VOID__STRING_STRING_STRING) (gpointer
data1,
                                                             gpointer
arg_1,
                                                             gpointer
arg_2,
                                                             gpointer
arg_3,
                                                             gpointer
data2);
    register GMarshalFunc_VOID__STRING_STRING_STRING callback;
    register GCClosure *cc = (GCClosure*) closure;
    register gpointer data1, data2;

    g_return_if_fail (n_param_values == 4);

    if (G_CCLOSURE_SWAP_DATA (closure))
    {
        data1 = closure->data;
        data2 = g_value_peek_pointer (param_values + 0);
    }
    else
    {
        data1 = g_value_peek_pointer (param_values + 0);
        data2 = closure->data;
    }
    callback = (GMarshalFunc_VOID__STRING_STRING_STRING) (marshal_data ?
marshal_data : cc->callback);

    callback (data1,
              g_marshal_value_peek_string (param_values + 1),

```

```
        g_marshal_value_peek_string (param_values + 2),
        g_marshal_value_peek_string (param_values + 3),
        data2);
}
```

File = dbus-gmarshal.h

```
#ifndef __dbus_g_marshal_MARSHAL_H__
#define __dbus_g_marshal_MARSHAL_H__

#include <glib-object.h>

G_BEGIN_DECLS

/* NONE:STRING,STRING,STRING (dbus-gmarshal.list:1) */
extern void _dbus_g_marshal_VOID__STRING_STRING_STRING (GClosure
*closure,
                                                    GValue
*return_value,
                                                    guint
n_param_values,
                                                    const GValue
*param_values,
                                                    gpointer
invocation_hint,
                                                    gpointer
marshal_data);
#define _dbus_g_marshal_NONE__STRING_STRING_STRING
        _dbus_g_marshal_VOID__STRING_STRING_STRING

G_END_DECLS

#endif /* __dbus_g_marshal_MARSHAL_H__ */
```

File = dbus-gmarshal.list

NONE:STRING,STRING,STRING

File = dbus-gmessage.sgml

```
<!-- ##### SECTION Title ##### -->
DBusGMessage
```

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->

<para>

</para>

<!-- ##### SECTION See_Also ##### -->

<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### STRUCT DBusGMessage ##### -->

<para>

</para>

<!-- ##### MACRO DBUS_TYPE_G_MESSAGE ##### -->

<para>

</para>

<!-- ##### FUNCTION dbus_g_message_ref ##### -->

<para>

</para>

@message:

@Returns:

<!-- ##### FUNCTION dbus_g_message_unref ##### -->

<para>

</para>

@message:

<!-- ##### FUNCTION dbus_g_message_get_message ##### -->

<para>

</para>

@gmessage:

@Returns:

File = dbus-gmethod.sgml

<!-- ##### SECTION Title ##### -->

DBusGMethod

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->

<para>

</para>

<!-- ##### SECTION See_Also ##### -->

<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### STRUCT DBusGMethodInfo ##### -->

<para>

</para>

@function:

@marshaller:

@data_offset:

<!-- ##### STRUCT DBusGMethodInvocation ##### -->

<para>

</para>

```
<!-- ##### FUNCTION dbus_g_method_get_sender ##### -->
<para>
```

```
</para>
```

```
@context:
```

```
@Returns:
```

```
<!-- ##### FUNCTION dbus_g_method_get_reply ##### -->
<para>
```

```
</para>
```

```
@context:
```

```
@Returns:
```

```
<!-- ##### FUNCTION dbus_g_method_send_reply ##### -->
<para>
```

```
</para>
```

```
@context:
```

```
@reply:
```

```
<!-- ##### FUNCTION dbus_g_method_return ##### -->
<para>
```

```
</para>
```

```
@context:
```

```
@...:
```

```
<!-- ##### FUNCTION dbus_g_method_return_error ##### -->
<para>
```

```
</para>
```

```
@context:
```

```
@error:
```

```
File = dbus-gobject.c
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gobject.c Exporting a GObject remotely
```



```
*
* Copyright (C) 2003, 2004, 2005 Red Hat, Inc.
* Copyright (C) 2005 Nokia
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include <gobject/gvaluecollector.h>
#include <dbus/dbus-glib.h>
#include <dbus/dbus-glib-lowlevel.h>
#include "dbus-gtest.h"
#include "dbus-gutils.h"
#include "dbus-gobject.h"
#include "dbus-gsignature.h"
#include "dbus-gvalue.h"
#include "dbus-gmarshal.h"
#include "dbus-gvalue-utils.h"
#include <string.h>
```

```
#include <gio/gio.h>
```

```
G_GNUC_NORETURN static void
oom (const gchar *explanation)
{
    g_error ("%s", explanation == NULL ? "Out of memory" : explanation);
    g_assert_not_reached ();
}
```

```
static DBusMessage *
reply_or_die (DBusMessage *in_reply_to)
{
    DBusMessage *reply;
```

```

g_return_val_if_fail (in_reply_to != NULL, NULL);

reply = dbus_message_new_method_return (in_reply_to);

if (reply == NULL)
    oom ("dbus_message_new_method_return failed: out of memory?");

return reply;
}

static DBusMessage *
error_or_die (DBusMessage *in_reply_to,
             const gchar *error_name,
             const gchar *error_message)
{
    DBusMessage *reply;

    g_return_val_if_fail (in_reply_to != NULL, NULL);
    /* error names are syntactically the same as interface names */
    g_return_val_if_fail (g_dbus_is_interface_name (error_name), NULL);
    g_return_val_if_fail (g_utf8_validate (error_message, -1, NULL),
                          NULL);

    reply = dbus_message_new_error (in_reply_to, error_name,
                                   error_message);

    if (reply == NULL)
        oom ("dbus_message_new_error failed: out of memory?");

    return reply;
}

static void
connection_send_or_die (DBusConnection *connection,
                       DBusMessage *message)
{
    g_return_if_fail (connection != NULL);
    g_return_if_fail (message != NULL);

    if (!dbus_connection_send (connection, message, NULL))
        oom ("dbus_connection_send failed: out of memory?");
}

static char *lookup_property_name (GObject *object,
                                   const char *wincaps_propiface,
                                   const char *requested_propname);

typedef struct
{
    char *default_iface;
    GType code_enum;
} DBusGErrorInfo;

```

```

static GStaticRWLock globals_lock = G_STATIC_RW_LOCK_INIT;
/* See comments in check_property_access */
static gboolean disable_legacy_property_access = FALSE;
static GHashTable *marshal_table = NULL;
static GData *error_metadata = NULL;

static char*
uscore_to_wincaps_full (const char *uscore,
                        gboolean    uppercase_first,
                        gboolean    strip_underscores)
{
    const char *p;
    GString *str;
    gboolean last_was_uscore;

    last_was_uscore = uppercase_first;

    str = g_string_new (NULL);
    p = uscore;
    while (p && *p)
        {
            if (*p == '-' || (strip_underscores && *p == '_'))
                {
                    last_was_uscore = TRUE;
                }
            else
                {
                    if (last_was_uscore)
                        {
                            g_string_append_c (str, g_ascii_toupper (*p));
                            last_was_uscore = FALSE;
                        }
                    else
                        g_string_append_c (str, *p);
                }
            ++p;
        }

    return g_string_free (str, FALSE);
}

/* Ugly yes - but we have to accept strings from both formats */
static gboolean
compare_strings_ignoring_uscore_vs_dash (const char *a, const char *b)
{
    guint i;

    for (i = 0; a[i] && b[i]; i++)
        {
            if ((a[i] == '-' && b[i] == '_')
                || (a[i] == '_' && b[i] == '-'))
        }

```

```

        continue;
        if (a[i] != b[i])
            return FALSE;
    }
    return (a[i] == '\0') && (b[i] == '\0');
}

static char *
uscore_to_wincaps (const char *uscore)
{
    return uscore_to_wincaps_full (uscore, TRUE, TRUE);
}

static const char *
string_table_next (const char *table)
{
    return (table + (strlen (table) + 1));
}

static const char *
string_table_lookup (const char *table, int index)
{
    const char *ret;

    ret = table;

    while (index--)
        ret = string_table_next (ret);

    return ret;
}

static const char *
get_method_data (const DBusGObjectInfo *object,
                 const DBusGMethodInfo *method)
{
    return object->data + method->data_offset;
}

static char *
object_error_domain_prefix_from_object_info (const DBusGObjectInfo
*info)
{
    /* FIXME */
    return NULL;
}

static char *
object_error_code_from_object_info (const DBusGObjectInfo *info,
GQuark domain, gint code)
{
    /* FIXME */

```

```

    return NULL;
}

static const char *
method_interface_from_object_info (const DBusGObjectInfo *object,
                                   const DBusGMethodInfo *method)
{
    return string_table_lookup (get_method_data (object, method), 0);
}

static const char *
method_name_from_object_info (const DBusGObjectInfo *object,
                              const DBusGMethodInfo *method)
{
    return string_table_lookup (get_method_data (object, method), 1);
}

static const char *
method_arg_info_from_object_info (const DBusGObjectInfo *object,
                                  const DBusGMethodInfo *method)
{
    return string_table_lookup (get_method_data (object, method),
3);/*RB was 2*/
}

typedef enum
{
    RETVAL_NONE,
    RETVAL_NOERROR,
    RETVAL_ERROR
} RetvalType;

/*
 * arg_iterate:
 * @data: a pointer to the beginning of an argument entry in a string
table
 * @name: (out) (allow-none): used to return the name of the next
argument
 * @in: (out) (allow-none): used to return %TRUE for an "in" argument
or
 * %FALSE for an "out" argument
 * @constval: (out) (allow-none): used to return %TRUE if the argument
is
 * an "out" argument and has the "C" (const) flag indicating that
it
 * should not be freed after it is returned; normally, "out"
arguments
 * are freed
 * @retval: (out) (allow-none): used to return %RETVAL_NONE if this
 * D-Bus argument is not an "out" argument or is obtained like a C
"out"
 * parameter, %RETVAL_ERROR if this argument is obtained from the C

```

```

*      return value and is also used to signal errors, or
%RETVAL_NOERROR
*      if this argument is obtained from the C return value and the
method
*      can never raise an error
* @type: (out) (allow-none): used to return the D-Bus signature of
this
*      argument
*
* The data format is:
*
* argument name
* \0
* direction: I or O
* \0
* if direction == "O":
*     freeable? F or C
*     \0
*     retval? N, E or R
*     \0
* signature
* \0
*
* If none of the arguments has @retval != %RETVAL_NONE, the method is
* assumed to return a gboolean, which behaves like %RETVAL_ERROR but
is
* not sent over D-Bus at all.
*
* Returns: the value of @data to use for the next call, or a pointer
to '\0'
*     if this function must not be called again
*/
static const char *
arg_iterate (const char    *data,
             const char    **name,
             gboolean      *in,
             gboolean      *constval,
             RetvalType     *retval,
             const char    **type)
{
    gboolean inarg;

    if (name)
        *name = data;

    data = string_table_next (data);
    switch (*data)
    {
        case 'I':
            inarg = TRUE;
            break;
        case 'O':

```

```

    inarg = FALSE;
    break;
default:
    g_warning ("invalid arg direction '%c'", *data);
    inarg = FALSE;
    break;
}
if (in)
    *in = inarg;

if (!inarg)
{
    data = string_table_next (data);
    switch (*data)
    {
    case 'F':
        if (constval)
            *constval = FALSE;
        break;
    case 'C':
        if (constval)
            *constval = TRUE;
        break;
    default:
        g_warning ("invalid arg const value '%c'", *data);
        break;
    }
    data = string_table_next (data);
    switch (*data)
    {
    case 'N':
        if (retval)
            *retval = RETVAL_NONE;
        break;
    case 'E':
        if (retval)
            *retval = RETVAL_ERROR;
        break;
    case 'R':
        if (retval)
            *retval = RETVAL_NOERROR;
        break;
    default:
        g_warning ("invalid arg ret value '%c'", *data);
        break;
    }
}
else
{
    if (constval)
        *constval = FALSE;
    if (retval)

```

```

        *retval = FALSE;
    }

    data = string_table_next (data);
    if (type)
        *type = data;

    return string_table_next (data);
}

static char *
method_dir_signature_from_object_info (const DBusGObjectInfo *object,
                                       const DBusGMethodInfo *method,
                                       gboolean                in)
{
    const char *arg;
    GString *ret;

    arg = method_arg_info_from_object_info (object, method);

    ret = g_string_new (NULL);

    while (*arg)
    {
        const char *name;
        gboolean arg_in;
        const char *type;

        arg = arg_iterate (arg, &name, &arg_in, NULL, NULL, &type);

        if (arg_in == in)
            g_string_append (ret, type);
    }

    return g_string_free (ret, FALSE);
}

static char *
method_input_signature_from_object_info (const DBusGObjectInfo
*object,
                                       const DBusGMethodInfo *method)
{
    return method_dir_signature_from_object_info (object, method, TRUE);
}

static char *
method_output_signature_from_object_info (const DBusGObjectInfo
*object,
                                       const DBusGMethodInfo *method)
{
    return method_dir_signature_from_object_info (object, method,
FALSE);
}

```



```

}

static const char *
signal_iterate (const char *data, const char **iface, const char
**name)
{
    *iface = data;

    data = string_table_next (data);
    *name = data;

    return string_table_next (data);
}

static const char *
property_iterate (const char *data,
                  int          format_version,
                  const char **iface,
                  const char **exported_name,
                  const char **name_uscored,
                  const char **access_type)
{
    *iface = data;

    data = string_table_next (data);
    *exported_name = data;

    data = string_table_next (data);
    if (format_version == 1)
    {
        *name_uscored = data;
        data = string_table_next (data);
        *access_type = data;
        return string_table_next (data);
    }
    else
    {
        /* This tells the caller they need to compute it */
        *name_uscored = NULL;
        /* We don't know here, however note that we will still check
against the
        * readable/writable flags from GObject's metadata.
        */
        *access_type = "readwrite";
        return data;
    }
}

/**
 * property_info_from_object_info:
 * @object: introspection data

```

```

* @interface_name: (allow-none): Expected interface name, or %NULL
for any
* @property_name: Expected property name (can use "-" or "_" as
separator)
* @access_type: (out): Can be one of "read", "write", "readwrite"
*
* Look up property introspection data for the given interface/name
pair.
*
* Returns: %TRUE if property was found
*/
static gboolean
property_info_from_object_info (const DBusGObjectInfo *object,
                                const char
*interface_name,
                                const char
                                const char
                                *property_name,
                                **access_type)
{
    const char *properties_iter;

    properties_iter = object->exported_properties;
    while (properties_iter != NULL && *properties_iter)
    {
        const char *cur_interface_name;
        const char *cur_property_name;
        const char *cur_uscore_property_name;
        const char *cur_access_type;

        properties_iter = property_iterate (properties_iter, object-
>format_version,
                                            &cur_interface_name,
&cur_property_name,
                                            &cur_uscore_property_name,
&cur_access_type);

        if (interface_name && strcmp (interface_name,
cur_interface_name) != 0)
            continue;

        /* This big pile of ugly is necessary to support the matrix
resulting from multiplying
        * (v0 data, v1 data) * (FooBar, foo-bar)
        * In v1 data we have both forms of string, so we do a
comparison against both without
        * having to malloc.
        * For v0 data, we need to reconstruct the foo-bar form.
        *
        * Adding to the complexity is that we *also* have to ignore the
distinction between
        * '-' and '_', because g_object_{get,set} does.
        */
    }
}

```

```

        /* First, compare against the primary property name - no malloc
        required */
        if (!compare_strings_ignoring_uscore_vs_dash (property_name,
        cur_property_name))
            {
                if (cur_uscore_property_name != NULL
                    && !compare_strings_ignoring_uscore_vs_dash
        (property_name, cur_uscore_property_name))
                    continue;
                else
                    {
                        /* v0 metadata, construct uscore */
                        char *tmp_uscored;
                        gboolean matches;
                        tmp_uscored = _dbus_gutils_wincaps_to_uscore
        (cur_property_name);
                        matches = compare_strings_ignoring_uscore_vs_dash
        (property_name, tmp_uscored);
                        g_free (tmp_uscored);
                        if (!matches)
                            continue;
                    }
            }

        *access_type = cur_access_type;
        return TRUE;
    }
    return FALSE;
}

static GQuark
dbus_g_object_type_dbus_metadata_quark (void)
{
    static GQuark quark;

    if (!quark)
        quark = g_quark_from_static_string
        ("DBusGObjectTypeDBusMetadataQuark");
    return quark;
}

/* Iterator function should return FALSE to stop iteration, TRUE to
continue */
typedef gboolean (*ForeachObjectInfoFn) (const DBusGObjectInfo *info,
                                         GType                       gtype,
                                         gpointer

user_data);

static void
foreach_object_info (GObject *object,
                    ForeachObjectInfoFn callback,
                    gpointer user_data)

```

```

{
    GType *interfaces, *p;
    const DBusGObjectInfo *info;
    GType classtype;

    interfaces = g_type_interfaces (G_TYPE_FROM_INSTANCE (object),
    NULL);

    for (p = interfaces; *p != 0; p++)
    {
        info = g_type_get_qdata (*p,
    dbus_g_object_type_dbus_metadata_quark ());
        if (info != NULL && info->format_version >= 0)
        {
            if (!callback (info, *p, user_data))
                break;
        }
    }

    g_free (interfaces);

    for (classtype = G_TYPE_FROM_INSTANCE (object); classtype != 0;
    classtype = g_type_parent (classtype))
    {
        info = g_type_get_qdata (classtype,
    dbus_g_object_type_dbus_metadata_quark ());
        if (info != NULL && info->format_version >= 0)
        {
            if (!callback (info, classtype, user_data))
                break;
        }
    }
}

static gboolean
lookup_object_info_cb (const DBusGObjectInfo *info,
                      GType gtype,
                      gpointer user_data)
{
    GList **list = (GList **) user_data;

    *list = g_list_prepend (*list, (gpointer) info);
    return TRUE;
}

static GList *
lookup_object_info (GObject *object)
{
    GList *info_list = NULL;

    foreach_object_info (object, lookup_object_info_cb, &info_list);
}

```

```

    return info_list;
}

typedef struct {
    const char *iface;
    const DBusGObjectInfo *info;
    gboolean fallback;
    GType iface_type;
} LookupObjectInfoByIfaceData;

static gboolean
lookup_object_info_by_iface_cb (const DBusGObjectInfo *info,
                                GType gtype,
                                gpointer user_data)
{
    LookupObjectInfoByIfaceData *lookup_data =
    (LookupObjectInfoByIfaceData *) user_data;

    /* If interface is not specified, choose the first info */
    if (lookup_data->fallback && (!lookup_data->iface || strlen
(lookup_data->iface) == 0))
        {
            lookup_data->info = info;
            lookup_data->iface_type = gtype;
        }
    else if (info->exported_properties && !strcmp (info-
>exported_properties, lookup_data->iface))
        {
            lookup_data->info = info;
            lookup_data->iface_type = gtype;
        }

    return !lookup_data->info;
}

static const DBusGObjectInfo *
lookup_object_info_by_iface (GObject *object,
                             const char *iface,
                             gboolean fallback,
                             GType *out_iface_type)
{
    LookupObjectInfoByIfaceData data;

    data.iface = iface;
    data.info = NULL;
    data.fallback = fallback;
    data.iface_type = 0;

    foreach_object_info (object, lookup_object_info_by_iface_cb, &data);

    if (out_iface_type && data.info)

```

```

    *out_iface_type = data.iface_type;

    return data.info;
}

typedef struct {
    /* owned */
    GSList *registrations;
    /* weak ref, or NULL if the object has been disposed */
    GObject *object;
} ObjectExport;

typedef struct {
    /* pseudo-weak ref, never NULL */
    DBusGConnection *connection;
    /* owned, never NULL */
    gchar *object_path;
    /* borrowed pointer to parent, never NULL */
    ObjectExport *export;
} ObjectRegistration;

static void object_export_object_died (gpointer user_data, GObject
*dead);

static void
object_export_unregister_all (ObjectExport *oe)
{
    while (oe->registrations != NULL)
    {
        GSList *old = oe->registrations;
        ObjectRegistration *o = oe->registrations->data;

        dbus_connection_unregister_object_path (
            DBUS_CONNECTION_FROM_G_CONNECTION (o->connection), o-
>object_path);

        /* the link should have been removed by doing that */
        g_assert (oe->registrations != old);
    }
}

static void
object_export_free (ObjectExport *oe)
{
    g_slice_free (ObjectExport, oe);
}

static ObjectExport *
object_export_new (void)
{
    return g_slice_new0 (ObjectExport);
}

```

```

static ObjectRegistration *
object_registration_new (DBusGConnection *connection,
                        const gchar *object_path,
                        ObjectExport *export)
{
    ObjectRegistration *o = g_slice_new0 (ObjectRegistration);

    o->connection = connection;
    o->object_path = g_strdup (object_path);
    o->export = export;

    return o;
}

static void
object_registration_free (ObjectRegistration *o)
{
    g_assert (o->export != NULL);
    o->export->registrations = g_slist_remove (o->export->registrations,
    o);

    g_free (o->object_path);

    g_slice_free (ObjectRegistration, o);
}

/* Called when the object falls off the bus (e.g. because connection
just
* closed) */
static void
object_registration_unregister (DBusConnection *connection,
                                void *user_data)
{
    object_registration_free (user_data);
}

typedef struct
{
    GObject *object;
    GString *xml;
    GType gtype;
    const DBusGObjectInfo *object_info;
} DBusGLibWriteInterfaceData;

typedef struct
{
    GSList *methods;
    GSList *signals;
    GSList *properties;
} DBusGLibWriteInterfaceValues;

```

```

static void
write_interface (gpointer key, gpointer val, gpointer user_data)
{
    const char *name;
    GSList *methods;
    GSList *signals;
    GSList *properties;
    GString *xml;
    const DBusGObjectInfo *object_info;
    DBusGLibWriteInterfaceData *data;
    DBusGLibWriteInterfaceValues *values;

    name = key;

    values = val;
    methods = values->methods;
    signals = values->signals;
    properties = values->properties;

    data = user_data;
    xml = data->xml;
    object_info = data->object_info;

    g_string_append_printf (xml, " <interface name=\"%s\">\n", name);

    /* FIXME: recurse to parent types ? */
    for (; methods; methods = methods->next)
    {
        DBusGMethodInfo *method;
        const char *args;
        method = methods->data;

        g_string_append_printf (xml, " <method name=\"%s\">\n",
                                method_name_from_object_info (object_info,
method));

        args = method_arg_info_from_object_info (object_info, method);

        while (*args)
        {
            const char *name;
            gboolean arg_in;
            const char *type;

            args = arg_iterate (args, &name, &arg_in, NULL, NULL, &type);

            /* FIXME - handle container types */
            g_string_append_printf (xml, " <arg name=\"%s\"
type=\"%s\" direction=\"%s\"/>\n",
                                name, type, arg_in ? "in" : "out");

        }
    }
}

```



```

        g_string_append (xml, "    </method>\n");
    }
    g_slist_free (values->methods);

    for (; signals; signals = signals->next)
    {
        guint id;
        guint arg;
        const char *signame;
        GSignalQuery query;
        char *s;

        signame = signals->data;

        s = _dbus_gutils_wincaps_to_uscore (signame);

        id = g_signal_lookup (s, data->gtype);
        g_assert (id != 0);

        g_signal_query (id, &query);
        g_assert (query.return_type == G_TYPE_NONE);

        g_string_append_printf (xml, "    <signal name=\"%s\">\n",
signame);

        for (arg = 0; arg < query.n_params; arg++)
        {
            char *dbus_type = _dbus_gtype_to_signature
(query.param_types[arg]);

            g_assert (dbus_type != NULL);

            g_string_append (xml, "        <arg type=\"");
            g_string_append (xml, dbus_type);
            g_string_append (xml, "\"/>\n");
            g_free (dbus_type);
        }

        g_string_append (xml, "    </signal>\n");
        g_free (s);
    }
    g_slist_free (values->signals);

    for (; properties; properties = properties->next)
    {
        const char *iface;
        const char *propname;
        const char *propname_uscore;
        const char *access_type;
        GParamSpec *spec;
        char *dbus_type;

```

```

gboolean can_set;
gboolean can_get;
char *s;

spec = NULL;

property_iterate (properties->data, object_info->format_version,
&iface, &propname, &propname_uscore, &access_type);

s = lookup_property_name (data->object, name, propname);

spec = g_object_class_find_property (g_type_class_peek (data-
>gtype), s);
g_assert (spec != NULL);
g_free (s);

dbus_type = _dbus_gtype_to_signature (G_PARAM_SPEC_VALUE_TYPE
(spec));
g_assert (dbus_type != NULL);

can_set = strcmp (access_type, "readwrite") == 0
&& ((spec->flags & G_PARAM_WRITABLE) != 0
&& (spec->flags & G_PARAM_CONSTRUCT_ONLY) == 0);

can_get = (spec->flags & G_PARAM_READABLE) != 0;

if (can_set || can_get)
{
g_string_append_printf (xml, "    <property name=\"%s\" ",
propname);
g_string_append (xml, "type=\"");
g_string_append (xml, dbus_type);
g_string_append (xml, "\" access=\"");

if (can_set && can_get)
g_string_append (xml, "readwrite");
else if (can_get)
g_string_append (xml, "read");
else
{
g_assert (can_set);
g_string_append (xml, "write");
}

g_string_append (xml, "\"/>\n");
}

g_free (dbus_type);
}
g_slist_free (values->properties);

g_free (values);

```

```

    g_string_append (xml, " </interface>\n");
}

static DBusGLibWriteInterfaceValues *
lookup_values (GHashTable *interfaces, const char *method_interface)
{
    DBusGLibWriteInterfaceValues *values;
    if ((values = g_hash_table_lookup (interfaces, (gpointer)
method_interface)) == NULL)
    {
        values = g_new0 (DBusGLibWriteInterfaceValues, 1);
        g_hash_table_insert (interfaces, (gpointer) method_interface,
values);
    }
    return values;
}

static void
introspect_interfaces (GObject *object, GString *xml)
{
    GList *info_list;
    const GList *info_list_walk;
    const DBusGObjectInfo *info;
    DBusGLibWriteInterfaceData data;
    int i;
    GHashTable *interfaces;
    DBusGLibWriteInterfaceValues *values;
    const char *propsig;

    info_list = lookup_object_info (object);

    g_assert (info_list != NULL);

    /* Gather a list of all interfaces, indexed into their methods */
    for (info_list_walk = info_list; info_list_walk != NULL;
info_list_walk = g_list_next (info_list_walk))
    {
        info = (DBusGObjectInfo *) info_list_walk->data;
        interfaces = g_hash_table_new (g_str_hash, g_str_equal);

        g_assert (info != NULL);

        for (i = 0; i < info->n_method_infos; i++)
        {
            const char *method_interface;
            const DBusGMethodInfo *method;

            method = &(info->method_infos[i]);

            method_interface = method_interface_from_object_info (info,
method);

```

```

        values = lookup_values (interfaces, method_interface);
        values->methods = g_slist_prepend (values->methods,
(gpoiner) method);
    }

    propsig = info->exported_signals;
    while (propsig != NULL && *propsig)
    {
        const char *iface;
        const char *signame;

        propsig = signal_iterate (propsig, &iface, &signame);

        values = lookup_values (interfaces, iface);
        values->signals = g_slist_prepend (values->signals,
(gpoiner) signame);
    }

    propsig = info->exported_properties;
    while (propsig != NULL && *propsig)
    {
        const char *iface;
        const char *propname;
        const char *propname_uscore;
        const char *access_type;

        propsig = property_iterate (propsig, info->format_version,
&iface, &propname, &propname_uscore, &access_type);

        values = lookup_values (interfaces, iface);
        values->properties = g_slist_prepend (values->properties,
(gpoiner)iface);
    }

    memset (&data, 0, sizeof (data));
    data.xml = xml;
    data.gtype = G_TYPE_FROM_INSTANCE (object);
    data.object_info = info;
    data.object = object;

    g_hash_table_foreach (interfaces, write_interface, &data);
    g_hash_table_destroy (interfaces);
}

g_list_free (info_list);
}

static DBusHandlerResult
handle_introspect (DBusConnection *connection,
                  DBusMessage *message,
                  GObject *object)
{

```

```

GString *xml;
unsigned int i;
DBusMessage *ret;
char **children;

if (!dbus_connection_list_registered (connection,
                                     dbus_message_get_path
(message),
                                     &children))
    oom (NULL);

xml = g_string_new (NULL);

g_string_append (xml, DBUS_INTROSPECT_1_0_XML_DOCTYPE_DECL_NODE);

g_string_append (xml, "<node>\n");

/* We are introspectable, though I guess that was pretty obvious */
g_string_append_printf (xml, " <interface name=\"%s\">\n",
DBUS_INTERFACE_INTROSPECTABLE);
g_string_append (xml, " <method name=\"Introspect\">\n");
g_string_append_printf (xml, " <arg name=\"data\"
direction=\"out\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append (xml, " </method>\n");
g_string_append (xml, " </interface>\n");

/* We support get/set/getall properties */
g_string_append_printf (xml, " <interface name=\"%s\">\n",
DBUS_INTERFACE_PROPERTIES);
g_string_append (xml, " <method name=\"Get\">\n");
g_string_append_printf (xml, " <arg name=\"interface\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append_printf (xml, " <arg name=\"propname\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append_printf (xml, " <arg name=\"value\"
direction=\"out\" type=\"%s\"/>\n", DBUS_TYPE_VARIANT_AS_STRING);
g_string_append (xml, " </method>\n");
g_string_append (xml, " <method name=\"Set\">\n");
g_string_append_printf (xml, " <arg name=\"interface\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append_printf (xml, " <arg name=\"propname\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append_printf (xml, " <arg name=\"value\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_VARIANT_AS_STRING);
g_string_append (xml, " </method>\n");
g_string_append (xml, " <method name=\"GetAll\">\n");
g_string_append_printf (xml, " <arg name=\"interface\"
direction=\"in\" type=\"%s\"/>\n", DBUS_TYPE_STRING_AS_STRING);
g_string_append_printf (xml, " <arg name=\"props\"
direction=\"out\" type=\"%s\"/>\n",
DBUS_TYPE_ARRAY_AS_STRING
DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING

```

```

        DBUS_TYPE_STRING_AS_STRING
        DBUS_TYPE_VARIANT_AS_STRING
        DBUS_DICT_ENTRY_END_CHAR_AS_STRING
    );

    g_string_append (xml, "    </method>\n");
    g_string_append (xml, " </interface>\n");

    introspect_interfaces (object, xml);

    /* Append child nodes */
    for (i = 0; children[i]; i++)
        g_string_append_printf (xml, " <node name=\"%s\"/>\n",
                                children[i]);

    /* Close the XML, and send it to the requesting app */
    g_string_append (xml, "</node>\n");

    ret = reply_or_die (message);

    dbus_message_append_args (ret,
                              DBUS_TYPE_STRING, &xml->str,
                              DBUS_TYPE_INVALID);

    connection_send_or_die (connection, ret);
    dbus_message_unref (ret);

    g_string_free (xml, TRUE);

    dbus_free_string_array (children);

    return DBUS_HANDLER_RESULT_HANDLED;
}

static DBusMessage*
set_object_property (DBusConnection *connection,
                    DBusMessage *message,
                    DBusMessageIter *iter,
                    GObject *object,
                    GParamSpec *pspec)
{
    GValue value = { 0, };
    DBusMessage *ret;
    DBusMessageIter sub;
    DBusGValueMarshalCtx context;

    dbus_message_iter_recurse (iter, &sub);

    context.recursion_depth = 0;
    context.gconnection = DBUS_G_CONNECTION_FROM_CONNECTION
(connection);
    context.proxy = NULL;

```

```

g_value_init (&value, pspec->value_type);
if (_dbus_gvalue_demarshal (&context, &sub, &value, NULL))
{
    g_object_set_property (object,
                           pspec->name,
                           &value);

    g_value_unset (&value);

    ret = reply_or_die (message);
}
else
{
    ret = error_or_die (message,
                       DBUS_ERROR_INVALID_ARGS,
                       "Argument's D-BUS type can't be converted to a GType");
}

return ret;
}

/*
 * @pspec: the paramspec for a D-Bus-exported property
 *
 * Returns: a reply for the Get() D-Bus method, either successful or
error
 */
static DBusMessage*
get_object_property (DBusConnection *connection,
                    DBusMessage *message,
                    GObject *object,
                    GParamSpec *pspec)
{
    GType value_gtype;
    GValue value = {0, };
    gchar *variant_sig;
    DBusMessage *ret;
    DBusMessageIter iter, subiter;
    gchar *error_message = NULL;

    ret = reply_or_die (message);

    g_value_init (&value, pspec->value_type);
    g_object_get_property (object, pspec->name, &value);

    variant_sig = _dbus_gvalue_to_signature (&value);
    if (variant_sig == NULL)
    {
        value_gtype = G_VALUE_TYPE (&value);
        error_message = g_strdup_printf (
            "Internal error: cannot marshal type \"%s\" in variant",

```

```

        g_type_name (value_gtype));
    goto out;
}

dbus_message_iter_init_append (ret, &iter);
if (!dbus_message_iter_open_container (&iter,
                                       DBUS_TYPE_VARIANT,
                                       variant_sig,
                                       &subiter))
{
    error_message = g_strdup_printf (
        "Internal error: cannot open variant container for signature
%s",
        variant_sig);
    goto out;
}

if (!_dbus_gvalue_marshal (&subiter, &value))
{
    dbus_message_iter_abandon_container (&iter, &subiter);
    error_message = g_strdup_printf (
        "Internal error: could not marshal type \"%s\" in variant",
        G_VALUE_TYPE_NAME (&value));
    goto out;
}

dbus_message_iter_close_container (&iter, &subiter);

out:
g_value_unset (&value);
g_free (variant_sig);

if (error_message != NULL)
{
    dbus_message_unref (ret);
    ret = error_or_die (message, DBUS_ERROR_FAILED, error_message);
    g_critical ("%s", error_message);
    g_free (error_message);
}

return ret;
}

#define SHADOW_PROP_QUARK
(dbus_g_object_type_dbus_shadow_property_quark ())

static GQuark
dbus_g_object_type_dbus_shadow_property_quark (void)
{
    static GQuark quark;

    if (!quark)

```



```

    quark = g_quark_from_static_string
("DBusGObjectTypeDBusShadowPropertyQuark");
    return quark;
}

/* Look for shadow properties on the given interface first, otherwise
 * just return the original property name. This allows
implementations to
 * get around the glib limitation of unique property names among all
 * GInterfaces by registering a "shadow" property name that the
get/set
 * request will be redirected to.
 *
 * Shadow property data is stored as qdata on each GInterface. If
there
 * is no interface info, or there is no registered shadow property,
just
 * return the original property name.
 */
static char *
lookup_property_name (GObject      *object,
                     const char *wincaps_propiface,
                     const char *requested_propname)
{
    const DBusGObjectInfo *object_info;
    GHashTable *shadow_props;
    char *shadow_prop_name = NULL, *uscore_name;
    GType iface_type = 0;

    g_assert (wincaps_propiface != NULL);
    g_assert (requested_propname != NULL);

    uscore_name = _dbus_gutils_wincaps_to_uscore (requested_propname);

    object_info = lookup_object_info_by_iface (object,
wincaps_propiface, FALSE, &iface_type);
    if (!object_info)
        return uscore_name;

    shadow_props = (GHashTable *) g_type_get_qdata (iface_type,
SHADOW_PROP_QUARK);
    if (shadow_props)
    {
        shadow_prop_name = g_strdup (g_hash_table_lookup (shadow_props,
requested_propname));
        if (shadow_prop_name)
            g_free (uscore_name);
    }

    return shadow_prop_name ? shadow_prop_name : uscore_name;
}

```

```

/**
 * dbus_g_object_type_register_shadow_property:
 * @iface_type: #GType for the #GInterface
 * @dbus_prop_name: D-Bus property name (as specified in the
introspection data)
 * to override with the shadow property name (as specified in the
GType's
 * initialization function, ie glib-style)
 * @shadow_prop_name: property name which should override the shadow
property
 *
 * Registers a new property name @shadow_prop_name that overrides the
 * @dbus_prop_name in D-Bus property get/set requests. Since all
properties for
 * all interfaces implemented by a GObject exist in the same
namespace, this
 * allows implementations to use the same property name in two or more
D-Bus
 * interfaces implemented by the same GObject, as long as one of those
D-Bus
 * interface properties is registered with a shadow property name.
 *
 * For example, if both org.foobar.Baz.InterfaceA and
org.foobar.Baz.InterfaceB
 * have a D-Bus property called "Bork", the developer assigns a shadow
property
 * name to the conflicting property name in one or both of these
GInterfaces to
 * resolve the conflict. Assume the GInterface implementing
 * org.foobar.Baz.InterfaceA registers a shadow property called "a-
bork", while
 * the GInterface implementing org.foobar.Baz.InterfaceB registers a
shadow
 * property called "b-bork". The GObject implementing both these
GInterfaces
 * would then use #g_object_class_override_property() to implement
both
 * "a-bork" and "b-bork" and D-Bus requests for "Bork" on either D-Bus
interface
 * will not conflict.
 */
void
dbus_g_object_type_register_shadow_property (GType          iface_type,
                                             const char
*dbus_prop_name,
                                             const char
*shadow_prop_name)
{
    GHashTable *shadow_props;

    g_return_if_fail (G_TYPE_IS_CLASSSED (iface_type) ||
G_TYPE_IS_INTERFACE (iface_type));

```

```

g_return_if_fail (dbus_prop_name != NULL);
g_return_if_fail (shadow_prop_name != NULL);

shadow_props = (GHashTable *) g_type_get_qdata (iface_type,
SHADOW_PROP_QUARK);
if (!shadow_props)
{
    shadow_props = g_hash_table_new_full (g_str_hash, g_str_equal,
g_free, g_free);
    g_type_set_qdata (iface_type,
dbus_g_object_type_dbus_shadow_property_quark
()),
        shadow_props);
}

g_assert (shadow_props);
g_hash_table_insert (shadow_props, g_strdup (dbus_prop_name),
g_strdup (shadow_prop_name));
}

static DBusMessage*
get_all_object_properties (DBusConnection      *connection,
                          DBusMessage        *message,
                          const DBusGObjectInfo *object_info,
                          const char         *wincaps_propiface,
                          GObject            *object)
{
    DBusMessage *ret;
    DBusMessageIter iter_ret;
    DBusMessageIter iter_dict;
    DBusMessageIter iter_dict_entry;
    DBusMessageIter iter_dict_value;
    const char *p;
    char *uscore_propname;

    ret = reply_or_die (message);

    dbus_message_iter_init_append (ret, &iter_ret);

    /* the types are all hard-coded, so this can only fail via OOM */
    if (!dbus_message_iter_open_container (&iter_ret,
                                          DBUS_TYPE_ARRAY,
                                          DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
                                          DBUS_TYPE_STRING_AS_STRING
                                          DBUS_TYPE_VARIANT_AS_STRING

                                          DBUS_DICT_ENTRY_END_CHAR_AS_STRING,
                                          &iter_dict))
        oom (NULL);

    p = object_info->exported_properties;

```

```

while (p != NULL && *p != '\0')
{
    const char *prop_ifname;
    const char *prop_name;
    const char *prop_uscored;
    const char *access_flags;
    GParamSpec *pspec;
    GType value_gtype;
    GValue value = {0, };
    gchar *variant_sig;

    p = property_iterate (p, object_info->format_version,
&prop_ifname, &prop_name, &prop_uscored, &access_flags);

    /* Conventionally, property names are valid member names, but
dbus-glib
* doesn't enforce this, and some dbus-glib services use
GObject-style
* property names (e.g. "foo-bar"). */
    if (!g_utf8_validate (prop_name, -1, NULL))
    {
        g_critical ("property name isn't UTF-8: %s", prop_name);
        continue;
    }

    uscore_propname = lookup_property_name (object,
wincaps_propiface, prop_name);

    pspec = g_object_class_find_property (G_OBJECT_GET_CLASS
(object), uscore_propname);
    if (pspec == NULL)
    {
        g_warning ("introspection data references non-existing
property %s", uscore_propname);
        g_free (uscore_propname);
        continue;
    }

    g_free (uscore_propname);

    g_value_init (&value, pspec->value_type);
    g_object_get_property (object, pspec->name, &value);

    variant_sig = _dbus_gvalue_to_signature (&value);
    if (variant_sig == NULL)
    {
        value_gtype = G_VALUE_TYPE (&value);
        g_warning ("Cannot marshal type \"%s\" in variant",
g_type_name (value_gtype));
        g_value_unset (&value);
        continue;
    }
}

```

```

/* a signature returned by _dbus_gvalue_to_signature had better
be
 * valid */
g_assert (g_variant_is_signature (variant_sig));

/* type is hard-coded, so this can't fail except by OOM */
if (!dbus_message_iter_open_container (&iter_dict,
                                      DBUS_TYPE_DICT_ENTRY,
                                      NULL,
                                      &iter_dict_entry))

    oom (NULL);

/* prop_name is valid UTF-8, so this can't fail except by OOM;
no point
 * in abandoning @iter_dict_entry since we're about to crash out
*/
if (!dbus_message_iter_append_basic (&iter_dict_entry,
DBUS_TYPE_STRING, &prop_name))
    oom (NULL);

/* variant_sig has been asserted to be valid, so this can't fail
 * except by OOM */
if (!dbus_message_iter_open_container (&iter_dict_entry,
                                      DBUS_TYPE_VARIANT,
                                      variant_sig,
                                      &iter_dict_value))

    oom (NULL);

g_free (variant_sig);

/* this can fail via programming error: the GObject property was
 * malformed (non-UTF8 string or something) */
if (!_dbus_gvalue_marshal (&iter_dict_value, &value))
{
    gchar *contents = g_strdup_value_contents (&value);
    gchar *error_message = g_strdup_printf (
        "cannot GetAll(%s): failed to serialize %s value of type
%s: %s",
        wincaps_propiface, prop_name, G_VALUE_TYPE_NAME
(&value),
        contents);

    g_critical ("%s", error_message);

    /* abandon ship! */
    dbus_message_iter_abandon_container (&iter_dict_entry,
&iter_dict_value);
    dbus_message_iter_abandon_container (&iter_dict,
&iter_dict_entry);
    dbus_message_unref (ret);

```

```

        ret = error_or_die (message, DBUS_ERROR_FAILED,
error_message);

        g_free (contents);
        g_free (error_message);
        g_value_unset (&value);
        return ret;
    }

    /* these shouldn't fail except by OOM now that we were
successful */
    if (!dbus_message_iter_close_container (&iter_dict_entry,
&iter_dict_value))
        oom (NULL);
    if (!dbus_message_iter_close_container (&iter_dict,
&iter_dict_entry))
        oom (NULL);

    g_value_unset (&value);
}

if (!dbus_message_iter_close_container (&iter_ret, &iter_dict))
    oom (NULL);

return ret;
}

static gboolean
lookup_object_and_method (GObject      *object,
                          DBusMessage *message,
                          const DBusGObjectInfo **object_ret,
                          const DBusGMethodInfo **method_ret)
{
    const char *interface;
    const char *member;
    const char *signature;
    GList *info_list;
    const GList *info_list_walk;
    const DBusGObjectInfo *info;
    int i;

    interface = dbus_message_get_interface (message);
    member = dbus_message_get_member (message);
    signature = dbus_message_get_signature (message);

    info_list = lookup_object_info (object);

    for (info_list_walk = info_list; info_list_walk != NULL;
info_list_walk = g_list_next (info_list_walk))
    {
        info = (DBusGObjectInfo *) info_list_walk->data;
        *object_ret = info;

```

```

for (i = 0; i < info->n_method_infos; i++)
{
    const char *expected_member;
    const char *expected_interface;
    char *expected_signature;
    const DBusGMethodInfo *method;

    method = &(info->method_infos[i]);

    /* Check method interface/name and input signature */
    expected_interface = method_interface_from_object_info
(*object_ret, method);
    expected_member = method_name_from_object_info (*object_ret,
method);
    expected_signature = method_input_signature_from_object_info
(*object_ret, method);

    if ((interface == NULL
        || strcmp (expected_interface, interface) == 0)
        && strcmp (expected_member, member) == 0
        && strcmp (expected_signature, signature) == 0)
    {
        g_free (expected_signature);
        *method_ret = method;
        g_list_free (info_list);
        return TRUE;
    }
    g_free (expected_signature);
}

if (info_list)
    g_list_free (info_list);

return FALSE;
}

static char *
gerror_domaincode_to_dbus_error_name (const DBusGObjectInfo
*object_info,
                                     const char *msg_interface,
                                     GQuark domain, gint code)
{
    const char *domain_str;
    const char *code_str;
    GString *dbus_error_name;

    domain_str = object_error_domain_prefix_from_object_info
(object_info);
    code_str = object_error_code_from_object_info (object_info, domain,
code);

```

```

if (!domain_str || !code_str)
{
    DBusGErrorInfo *info;

    g_static_rw_lock_reader_lock (&globals_lock);

    if (error_metadata != NULL)
info = g_datalist_id_get_data (&error_metadata, domain);
    else
info = NULL;

    g_static_rw_lock_reader_unlock (&globals_lock);

    if (info)
    {
        GEnumValue *value;
        GEnumClass *klass;

        klass = g_type_class_ref (info->code_enum);
        value = g_enum_get_value (klass, code);
        g_type_class_unref (klass);

        domain_str = info->default_iface;
        if (value)
            {
                code_str = value->value_nick;
            }
        else
            {
                g_warning ("Error code %d out of range for GError domain
%s",
                            code, g_quark_to_string (domain));
                code_str = NULL;
            }
    }
}

if (!domain_str)
    domain_str = msg_interface;

if (!domain_str || !code_str)
{
    const char *domain_string;
    /* If we can't map it sensibly, make up an error name */

    dbus_error_name = g_string_new
("org.freedesktop.DBus.GLib.UnmappedError.");

    domain_string = g_quark_to_string (domain);
    if (domain_string != NULL)
        {

```



```

        char *uscoded = uscore_to_wincaps (domain_string);
        g_string_append (dbus_error_name, uscoded);
        g_string_append_c (dbus_error_name, '.');
        g_free (uscoded);
    }

    /* Map -1 to (unsigned) -1 to avoid "-", which is not valid */
    g_string_append_printf (dbus_error_name, "Code%u", (unsigned)
code);
    }
    else
    {
        gchar *code_str_wincaps;
        dbus_error_name = g_string_new (domain_str);
        g_string_append_c (dbus_error_name, '.');
        /* We can't uppercase here for backwards compatibility
        * reasons; if someone had a lowercase enumeration value,
        * previously we'd just send it across unaltered.
        */
        code_str_wincaps = uscore_to_wincaps_full (code_str, FALSE,
FALSE);
        g_string_append (dbus_error_name, code_str_wincaps);
        g_free (code_str_wincaps);
    }

    return g_string_free (dbus_error_name, FALSE);
}

static DBusMessage *
gerror_to_dbus_error_message (const DBusGObjectInfo *object_info,
                             DBusMessage          *message,
                             const GError          *error)
{
    DBusMessage *reply;

    if (!error)
    {
        char *error_msg;

        error_msg = g_strdup_printf ("Method invoked for %s returned
FALSE but did not set error", dbus_message_get_member (message));
        reply = error_or_die (message,
"org.freedesktop.DBus.GLib.ErrorError", error_msg);
        g_free (error_msg);
    }
    else
    {
        if (error->domain == DBUS_GERROR)
        {
            const gchar *name = DBUS_ERROR_FAILED;

            switch (error->code)

```

```
{
case DBUS_GERROR_FAILED:
    name = DBUS_ERROR_FAILED;
    break;
case DBUS_GERROR_NO_MEMORY:
    name = DBUS_ERROR_NO_MEMORY;
    break;
case DBUS_GERROR_SERVICE_UNKNOWN:
    name = DBUS_ERROR_SERVICE_UNKNOWN;
    break;
case DBUS_GERROR_NAME_HAS_NO_OWNER:
    name = DBUS_ERROR_NAME_HAS_NO_OWNER;
    break;
case DBUS_GERROR_NO_REPLY:
    name = DBUS_ERROR_NO_REPLY;
    break;
case DBUS_GERROR_IO_ERROR:
    name = DBUS_ERROR_IO_ERROR;
    break;
case DBUS_GERROR_BAD_ADDRESS:
    name = DBUS_ERROR_BAD_ADDRESS;
    break;
case DBUS_GERROR_NOT_SUPPORTED:
    name = DBUS_ERROR_NOT_SUPPORTED;
    break;
case DBUS_GERROR_LIMITS_EXCEEDED:
    name = DBUS_ERROR_LIMITS_EXCEEDED;
    break;
case DBUS_GERROR_ACCESS_DENIED:
    name = DBUS_ERROR_ACCESS_DENIED;
    break;
case DBUS_GERROR_AUTH_FAILED:
    name = DBUS_ERROR_AUTH_FAILED;
    break;
case DBUS_GERROR_NO_SERVER:
    name = DBUS_ERROR_NO_SERVER;
    break;
case DBUS_GERROR_TIMEOUT:
    name = DBUS_ERROR_TIMEOUT;
    break;
case DBUS_GERROR_NO_NETWORK:
    name = DBUS_ERROR_NO_NETWORK;
    break;
case DBUS_GERROR_ADDRESS_IN_USE:
    name = DBUS_ERROR_ADDRESS_IN_USE;
    break;
case DBUS_GERROR_DISCONNECTED:
    name = DBUS_ERROR_DISCONNECTED;
    break;
case DBUS_GERROR_INVALID_ARGS:
    name = DBUS_ERROR_INVALID_ARGS;
    break;
}
```

```

        case DBUS_GERROR_FILE_NOT_FOUND:
            name = DBUS_ERROR_FILE_NOT_FOUND;
            break;
        case DBUS_GERROR_REMOTE_EXCEPTION:
            name = dbus_g_error_get_name ((GError*) error);
            break;
    }

    reply = error_or_die (message, name, error->message);
}
else
{
    char *error_name;
    error_name = gerror_domaincode_to_dbus_error_name (object_info,
                                                       dbus_message_get_interface
(message),
                                                       error->domain, error-
>code);
    reply = error_or_die (message, error_name, error->message);
    g_free (error_name);
}
}

return reply;
}

/**
 * SECTION:dbus-gmethod
 * @short_description: GMethod Info & Invocation
 * @see_also: #DBusGMessage
 * @stability: Stable
 *
 * These types are used to call methods on #GObject objects.
 */

/**
 * DBusGMethodInvocation:
 *
 * The context of an asynchronous method call. See
dbus_g_method_return() and
 * dbus_g_method_return_error().
 */
struct _DBusGMethodInvocation {
    DBusGConnection *connection; /**< The connection */
    DBusGMessage *message; /**< The message which generated the method
call */
    const DBusGObjectInfo *object; /**< The object the method was called
on */
    const DBusGMethodInfo *method; /**< The method called */
    gboolean send_reply;
};

```

```

static DBusHandlerResult
invoke_object_method (GObject          *object,
                     const DBusGObjectInfo *object_info,
                     const DBusGMethodInfo *method,
                     DBusConnection    *connection,
                     DBusMessage        *message)
{
    gboolean had_error, is_async, send_reply;
    GError *gerror;
    GValueArray *value_array;
    GValue return_value = {0,};
    GClosure closure;
    char *in_signature;
    GArray *out_param_values = NULL;
    GValueArray *out_param_gvalues = NULL;
    int out_param_count;
    int out_param_pos, out_param_gvalue_pos;
    DBusMessage *reply = NULL;
    gboolean have_retval;
    gboolean retval_signals_error;
    gboolean retval_is_synthetic;
    gboolean retval_is_constant;
    const char *arg_metadata;

    gerror = NULL;

    /* This flag says whether invokee is handed a special
    DBusGMethodInfoInvocation structure,
    * instead of being required to fill out all return values in the
    context of the function.
    * Some additional data is also exposed, such as the message sender.
    */
    is_async = strcmp (string_table_lookup (get_method_data
    (object_info, method), 2), "A") == 0;

    /* Messages can be sent with a flag that says "I don't need a
    reply". This is an optimization
    * normally, but in the context of the system bus it's important to
    not send a reply
    * to these kinds of messages, because they will be unrequested
    replies, and thus subject
    * to denial and logging. We don't want to fill up logs.
    * http://bugs.freedesktop.org/show_bug.cgi?id=19441
    */
    send_reply = !dbus_message_get_no_reply (message);

    have_retval = FALSE;
    retval_signals_error = FALSE;
    retval_is_synthetic = FALSE;
    retval_is_constant = FALSE;

    /* This is evil. We do this to work around the fact that

```

```

    * the generated glib marshallers check a flag in the closure object
    * which we don't care about. We don't need/want to create
    * a new closure for each invocation.
    */
memset (&closure, 0, sizeof (closure));

in_signature = method_input_signature_from_object_info (object_info,
method);

/* Convert method IN parameters to GValueArray */
{
    GArray *types_array;
    guint n_params;
    const GType *types;
    DBusGValueMarshalCtx context;
    GError *error = NULL;

    context.recursion_depth = 0;
    context.gconnection = DBUS_G_CONNECTION_FROM_CONNECTION
(connection);
    context.proxy = NULL;

    types_array = _dbus_gtypes_from_arg_signature (in_signature,
FALSE);
    n_params = types_array->len;
    types = (const GType*) types_array->data;

    value_array = _dbus_gvalue_demarshal_message (&context, message,
n_params, types, &error);
    if (value_array == NULL)
        {
            g_free (in_signature);
            g_array_free (types_array, TRUE);
            reply = error_or_die (message,
"org.freedesktop.DBus.GLib.ErrorError", error->message);
            connection_send_or_die (connection, reply);
            dbus_message_unref (reply);
            g_error_free (error);
            return DBUS_HANDLER_RESULT_HANDLED;
        }
    g_array_free (types_array, TRUE);
}

/* Prepend object as first argument */
g_value_array_prepend (value_array, NULL);
g_value_init (g_value_array_get_nth (value_array, 0),
G_TYPE_OBJECT);
g_value_set_object (g_value_array_get_nth (value_array, 0), object);

if (is_async)
    {
        GValue context_value = {0,};

```

```

    DBusGMethodInvocation *context;
    context = g_new (DBusGMethodInvocation, 1);
    context->connection = dbus_g_connection_ref
(DBUS_G_CONNECTION_FROM_CONNECTION (connection));
    context->message = dbus_g_message_ref
(DBUS_G_MESSAGE_FROM_MESSAGE (message));
    context->object = object_info;
    context->method = method;
    context->send_reply = send_reply;
    g_value_init (&context_value, G_TYPE_POINTER);
    g_value_set_pointer (&context_value, context);
    g_value_array_append (value_array, &context_value);
}
else
{
    RetvalType retval;
    gboolean arg_in;
    gboolean arg_const;
    const char *argsig;

    arg_metadata = method_arg_info_from_object_info (object_info,
method);

    /* Count number of output parameters, and look for a return
value */
    out_param_count = 0;
    while (*arg_metadata)
    {
        arg_metadata = arg_iterate (arg_metadata, NULL, &arg_in,
&arg_const, &retval, &argsig);
        if (arg_in)
            continue;
        if (retval != RETVAL_NONE)
        {
            DBusSignatureIter tmp_sigiter;
            /* This is the function return value */
            g_assert (!have_retval);
            have_retval = TRUE;
            retval_is_synthetic = FALSE;

            switch (retval)
            {
            case RETVAL_NONE:
                g_assert_not_reached ();
                break;
            case RETVAL_NOERROR:
                retval_signals_error = FALSE;
                break;
            case RETVAL_ERROR:
                retval_signals_error = TRUE;
                break;
            }
        }
    }
}

```

```

        retval_is_constant = arg_const;

        /* Initialize our return GValue with the specified type */
        dbus_signature_iter_init (&tmp_sigiter, argsig);
        g_value_init (&return_value,
_dbus_gtype_from_signature_iter (&tmp_sigiter, FALSE));
    }
    else
    {
        /* It's a regular output value */
        out_param_count++;
    }
}

/* For compatibility, if we haven't found a return value, we
assume
* the function returns a gboolean for signalling an error
* (and therefore also takes a GError). We also note that it
* is a "synthetic" return value; i.e. we aren't going to be
* sending it over the bus, it's just to signal an error.
*/
if (!have_retval)
{
    have_retval = TRUE;
    retval_is_synthetic = TRUE;
    retval_signals_error = TRUE;
    g_value_init (&return_value, G_TYPE_BOOLEAN);
}

/* Create an array to store the actual values of OUT parameters
* (other than the real function return, if any). Then, create
* a GValue boxed POINTER to each of those values, and append to
* the invocation, so the method can return the OUT parameters.
*/
out_param_values = g_array_sized_new (FALSE, TRUE, sizeof
(GTypeCValue), out_param_count);

/* We have a special array of GValues for toplevel GValue return
* types.
*/
out_param_gvalues = g_value_array_new (out_param_count);
out_param_pos = 0;
out_param_gvalue_pos = 0;

/* Reset argument metadata pointer */
arg_metadata = method_arg_info_from_object_info (object_info,
method);

/* Iterate over output arguments again, this time allocating
space for
* them as appropriate.

```

```

    */
while (*arg_metadata)
{
    GValue value = {0, };
    GTypeCValue storage;
    DBusSignatureIter tmp_sigiter;
    GType current_gtype;

    arg_metadata = arg_iterate (arg_metadata, NULL, &arg_in, NULL,
&retval, &argsig);
    /* Skip over input arguments and the return value, if any */
    if (arg_in || retval != RETVAL_NONE)
        continue;

    dbus_signature_iter_init (&tmp_sigiter, argsig);
    current_gtype = _dbus_gtype_from_signature_iter (&tmp_sigiter,
FALSE);

    g_value_init (&value, G_TYPE_POINTER);

    /* We special case variants to make method invocation a bit
nicer */
    if (current_gtype != G_TYPE_VALUE)
    {
        memset (&storage, 0, sizeof (storage));
        g_array_append_val (out_param_values, storage);
        g_value_set_pointer (&value, &(g_array_index
(out_param_values, GTypeCValue, out_param_pos)));
        out_param_pos++;
    }
    else
    {
        g_value_array_append (out_param_gvalues, NULL);
        g_value_set_pointer (&value, out_param_gvalues->values +
out_param_gvalue_pos);
        out_param_gvalue_pos++;
    }
    g_value_array_append (value_array, &value);
}

/* Append GError as final argument if necessary */
if (retval_signals_error)
{
    g_assert (have_retval);
    g_value_array_append (value_array, NULL);
    g_value_init (g_value_array_get_nth (value_array, value_array-
>n_values - 1), G_TYPE_POINTER);
    g_value_set_pointer (g_value_array_get_nth (value_array,
value_array->n_values - 1), &gerror);
}

```



```

/* Actually invoke method */
method->marshaller (&closure, have_retval ? &return_value : NULL,
                    value_array->n_values,
                    value_array->values,
                    NULL, method->function);

if (is_async)
{
    goto done;
}

if (retval_signals_error)
    had_error = _dbus_gvalue_signals_error (&return_value);
else
    had_error = FALSE;

if (!had_error)
{
    DBusMessageIter iter;

    /* Careful here - there are two major cases in this section of
the code.
    * If send_reply is TRUE, we're constructing a dbus message and
freeing
    * the return values.  If it's FALSE, then we just need to free
the
    * values.
    */
    if (send_reply)
    {
        reply = reply_or_die (message);

        /* Append output arguments to reply */
        dbus_message_iter_init_append (reply, &iter);
    }

    /* First, append the return value, unless it's synthetic */
    if (have_retval && !retval_is_synthetic)
    {
        if (reply != NULL && !_dbus_gvalue_marshal (&iter,
&return_value))
        {
            gchar *desc = g_strdup_value_contents (&return_value);

            g_critical ("unable to append retval of type %s for %s:
%s",
                        G_VALUE_TYPE_NAME (&return_value),
                        method_name_from_object_info (object_info, method),
                        desc);
            g_free (desc);
            /* the reply is now unusable but we still need to free
            * everything */
            dbus_message_unref (reply);

```

```

        reply = NULL;
    }

    if (!retval_is_constant)
        g_value_unset (&return_value);
}

/* Grab the argument metadata and iterate over it */
arg_metadata = method_arg_info_from_object_info (object_info,
method);

/* Now append any remaining return values */
out_param_pos = 0;
out_param_gvalue_pos = 0;
while (*arg_metadata)
{
    GValue gvalue = {0, };
    const char *arg_name;
    gboolean arg_in;
    gboolean constval;
    RetvalType retval;
    const char *arg_signature;
    DBusSignatureIter argsigiter;

    do
    {
        /* Iterate over only output values; skip over input
        arguments and the return value */
        arg_metadata = arg_iterate (arg_metadata, &arg_name,
&arg_in, &constval, &retval, &arg_signature);
    }
    while ((arg_in || retval != RETVAL_NONE) && *arg_metadata);

    /* If the last argument we saw was input or the return
    * value, we must be done iterating over output arguments.
    */
    if (arg_in || retval != RETVAL_NONE)
        break;

    dbus_signature_iter_init (&argsigiter, arg_signature);

    g_value_init (&gvalue, _dbus_gtype_from_signature_iter
(&argsigiter, FALSE));
    if (G_VALUE_TYPE (&gvalue) != G_TYPE_VALUE)
    {
        if (!_dbus_gvalue_take (&gvalue,
&(g_array_index (out_param_values,
GTypeCValue, out_param_pos))))
            g_assert_not_reached ();
        out_param_pos++;
    }
    else

```

```

        {
            g_value_set_static_boxed (&gvalue, out_param_gvalues-
>values + out_param_gvalue_pos);
            out_param_gvalue_pos++;
        }

        if (reply && !_dbus_gvalue_marshal (&iter, &gvalue))
        {
            gchar *desc = g_strdup_value_contents (&gvalue);

            g_critical ("unable to append OUT arg of type %s for %s:
%s",
                G_VALUE_TYPE_NAME (&gvalue),
                method_name_from_object_info (object_info, method),
                desc);
            g_free (desc);
            /* the reply is now unusable but we still need to free
             * everything */
            dbus_message_unref (reply);
            reply = NULL;
        }

        /* Here we actually free the allocated value; we
         * took ownership of it with _dbus_gvalue_take, unless
         * an annotation has specified this value as constant.
         */
        if (!constval)
            g_value_unset (&gvalue);
    }
}
else if (send_reply)
    reply = gerror_to_dbus_error_message (object_info, message,
gerror);

if (reply)
    {
        connection_send_or_die (connection, reply);
        dbus_message_unref (reply);
    }

done:
    g_free (in_signature);

    if (!is_async)
        {
            g_array_free (out_param_values, TRUE);
            g_value_array_free (out_param_gvalues);
        }

    if (gerror != NULL)
        g_clear_error (&gerror);

```

```

    g_value_array_free (value_array);
    return DBUS_HANDLER_RESULT_HANDLED;
}

/*
 * @wincaps_propiface: the D-Bus interface name, conventionally
WindowsCaps
 * @requested_propname: the D-Bus property name, conventionally
WindowsCaps
 * @uscore_propname: the GObject property name, conventionally
 *   words_with_underscores or words-with-dashes
 * @is_set: %TRUE if we're going to set the property, %FALSE if we're
going
 *   to get it
 *
 * Check that the requested property exists and the requested access
is
 * allowed. If not, reply with a D-Bus AccessDenied error message.
 *
 * Returns: %TRUE if property access can continue, or %FALSE if an
error
 *   reply has been sent
 */
static gboolean
check_property_access (DBusConnection *connection,
                      DBusMessage *message,
                      GObject *object,
                      const char *wincaps_propiface,
                      const char *requested_propname,
                      const char *uscore_propname,
                      gboolean is_set)
{
    const DBusGObjectInfo *object_info;
    const char *access_type;
    DBusMessage *ret;
    gchar *error_message;

    if (!is_set && !disable_legacy_property_access)
        return TRUE;

    object_info = lookup_object_info_by_iface (object,
wincaps_propiface, TRUE, NULL);
    if (!object_info)
    {
        error_message = g_strdup_printf (
            "Interface \"%s\" isn't exported (or may not exist), can't
access property \"%s\"",
            wincaps_propiface, requested_propname);

        goto error;
    }
}

```

```

    /* Try both forms of property names: "foo_bar" or "FooBar"; for
historical
    * reasons we accept both.
    */
    if (object_info
        && !(property_info_from_object_info (object_info,
wincaps_propiface, requested_propname, &access_type)
        || property_info_from_object_info (object_info,
wincaps_propiface, uscore_propname, &access_type))
        {
            error_message = g_strdup_printf (
                "Property \"%s\" of interface \"%s\" isn't exported (or may
not exist)",
                requested_propname, wincaps_propiface);

            goto error;
        }

    if (strcmp (access_type, "readwrite") == 0)
        return TRUE;

    if (is_set ? strcmp (access_type, "read") == 0
        : strcmp (access_type, "write") == 0)
        {
            error_message = g_strdup_printf (
                "Property \"%s\" of interface \"%s\" is not %s",
                requested_propname,
                wincaps_propiface,
                is_set ? "settable" : "readable");

            goto error;
        }

    return TRUE;

error:
    ret = error_or_die (message, DBUS_ERROR_ACCESS_DENIED,
error_message);
    g_free (error_message);

    connection_send_or_die (connection, ret);
    dbus_message_unref (ret);
    return FALSE;
}

static DBusHandlerResult
object_registration_message (DBusConnection *connection,
                             DBusMessage *message,
                             void *user_data)
{
    GParamSpec *pspec;
    GObject *object;

```



```

if (!(setter || getter || getall))
    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

ret = NULL;

dbus_message_iter_init (message, &iter);

if (dbus_message_iter_get_arg_type (&iter) != DBUS_TYPE_STRING)
{
    ret = error_or_die (message, DBUS_ERROR_INVALID_ARGS,
        "First argument to Get(), GetAll() or Set() must be an
interface string");
    goto out;
}

dbus_message_iter_get_basic (&iter, &wincaps_propiface);
dbus_message_iter_next (&iter);

if (getall)
{
    object_info = lookup_object_info_by_iface (object,
wincaps_propiface, TRUE, NULL);
    if (object_info != NULL)
        ret = get_all_object_properties (connection, message,
object_info, wincaps_propiface, object);
    else
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}
else
{
    g_assert (getter || setter);

    if (dbus_message_iter_get_arg_type (&iter) != DBUS_TYPE_STRING)
    {
        ret = error_or_die (message, DBUS_ERROR_INVALID_ARGS,
            "Second argument to Get() or Set() must be a property
name string");
        goto out;
    }

    dbus_message_iter_get_basic (&iter, &requested_propname);
    dbus_message_iter_next (&iter);

    s = lookup_property_name (object, wincaps_propiface,
requested_propname);

    if (!check_property_access (connection, message, object,
wincaps_propiface, requested_propname, s, setter))
    {
        g_free (s);
        return DBUS_HANDLER_RESULT_HANDLED;
    }
}

```

```

    }

    pspec = g_object_class_find_property (G_OBJECT_GET_CLASS
(object),
                                        s);

    g_free (s);

    if (pspec != NULL)
    {
        if (setter)
        {
            if (dbus_message_iter_get_arg_type (&iter) !=
DBUS_TYPE_VARIANT)
            {
                ret = error_or_die (message,
DBUS_ERROR_INVALID_ARGS,
                "Third argument to Set() must be a variant");
                goto out;
            }

            ret = set_object_property (connection, message, &iter,
object, pspec);
            dbus_message_iter_next (&iter);
        }
        else
        {
            g_assert (getter);
            ret = get_object_property (connection, message,
object, pspec);
        }
    }
    else
    {
        gchar *error_message = g_strdup_printf ("No such property
%s",
        requested_propname);

        ret = error_or_die (message, DBUS_ERROR_INVALID_ARGS,
error_message);
        g_free (error_message);
    }
}

g_assert (ret != NULL);

/* FIXME: this should be returned as a D-Bus error, not spammed out
* as a warning. This is too late to do that, though - we've already
* had any side-effects we were going to have - and it would break
* anything that's relying on ability to give us too many arguments.
*/
if (dbus_message_iter_get_arg_type (&iter) != DBUS_TYPE_INVALID)

```



```

    g_warning ("Property get, set or set all had too many
arguments\n");

out:
    connection_send_or_die (connection, ret);
    dbus_message_unref (ret);
    return DBUS_HANDLER_RESULT_HANDLED;
}

static const DBusObjectPathVTable gobject_dbus_vtable = {
    object_registration_unregister,
    object_registration_message,
    NULL
};

typedef struct {
    GClosure          closure;
    DBusGConnection *connection;
    GObject          *object;
    const char       *signame;
    const char       *sigiface;
} DBusGSignalClosure;

static GClosure *
dbus_g_signal_closure_new (GObject          *object,
                          const char       *signame,
                          const char       *sigiface)
{
    DBusGSignalClosure *closure;

    closure = (DBusGSignalClosure*) g_closure_new_simple (sizeof
(DBusGSignalClosure), NULL);

    closure->object = object;
    closure->signame = signame;
    closure->sigiface = sigiface;
    return (GClosure*) closure;
}

static void
emit_signal_for_registration (ObjectRegistration *o,
                             DBusGSignalClosure *sigclosure,
                             GValue            *retval,
                             guint              n_param_values,
                             const GValue      *param_values)
{
    DBusMessage *signal;
    DBusMessageIter iter;
    guint i;

    g_assert (g_variant_is_object_path (o->object_path));
    g_assert (g_dbus_is_interface_name (sigclosure->sigiface));

```

```

g_assert (g_dbus_is_member_name (sigclosure->signame));

signal = dbus_message_new_signal (o->object_path,
                                sigclosure->sigiface,
                                sigclosure->signame);

if (!signal)
    oom (NULL);

dbus_message_iter_init_append (signal, &iter);

/* First argument is the object itself, and we can't marshall that
*/
for (i = 1; i < n_param_values; i++)
{
    if (!_dbus_gvalue_marshall (&iter,
                                (GValue *) (&(param_values[i])))
        {
            g_warning ("failed to marshal parameter %d for signal %s",
                        i, sigclosure->signame);
            goto out;
        }
}

connection_send_or_die (DBUS_CONNECTION_FROM_G_CONNECTION (o-
>connection),
                        signal);
out:
dbus_message_unref (signal);
}

static void
signal_emitter_marshall (GClosure          *closure,
                        GValue             *retval,
                        guint               n_param_values,
                        const GValue       *param_values,
                        gpointer            invocation_hint,
                        gpointer            marshal_data)
{
    DBusGSignalClosure *sigclosure;
    const ObjectExport *oe;
    const GSList *iter;

    sigclosure = (DBusGSignalClosure *) closure;

    g_assert (retval == NULL);

    oe = g_object_get_data (sigclosure->object,
                            "dbus_glib_object_registrations");
    /* If the object has ever been exported, this should exist; it
    persists until
    * the object is actually freed. */
    g_assert (oe != NULL);

```

```

for (iter = oe->registrations; iter; iter = iter->next)
{
    ObjectRegistration *o = iter->data;

    emit_signal_for_registration (o, sigclosure, retval,
n_param_values, param_values);
}
}

static void
export_signals (const GList *info_list, GObject *object)
{
    GType gtype;
    const char *sigdata;
    const char *iface;
    const char *signame;
    const DBusGObjectInfo *info;

    gtype = G_TYPE_FROM_INSTANCE (object);

    for (; info_list != NULL; info_list = g_list_next (info_list))
    {
        info = (DBusGObjectInfo *) info_list->data;

        sigdata = info->exported_signals;

        while (*sigdata != '\0')
        {
            guint id;
            GSignalQuery query;
            GClosure *closure;
            char *s;

            sigdata = signal_iterate (sigdata, &iface, &signame);

            if (!g_dbus_is_interface_name (iface))
            {
                g_critical ("invalid interface name found in %s: %s",
                    g_type_name (gtype), iface);
                continue;
            }

            if (!g_dbus_is_member_name (signame))
            {
                g_critical ("invalid signal name found in %s: %s",
                    g_type_name (gtype), signame);
                continue;
            }

            s = _dbus_gutils_wincaps_to_uscore (signame);

```

```

        id = g_signal_lookup (s, gtype);
        if (id == 0)
        {
            g_warning ("signal \"%s\" (from \"%s\") exported but not
found in object class \"%s\"",
                s, signame, g_type_name (gtype));
            g_free (s);
            continue;
        }

        g_signal_query (id, &query);

        if (query.return_type != G_TYPE_NONE)
        {
            g_warning ("Not exporting signal \"%s\" for object class
\"%s\" as it has a return type \"%s\"",
                s, g_type_name (gtype), g_type_name
(query.return_type));
            g_free (s);
            continue; /* FIXME: these could be listed as methods ?
*/
        }

        closure = dbus_g_signal_closure_new (object, signame,
(char*) iface);
        g_closure_set_marshal (closure, signal_emitter_marshaller);

        g_signal_connect_closure_by_id (object,
            id,
            0,
            closure,
            FALSE);

        g_free (s);
    }
}

static gint
dbus_error_to_gerror_code (const char *derr)
{
    if (0) ;
    else if (!strcmp (derr, DBUS_ERROR_FAILED ))
        return DBUS_GERROR_FAILED ;
    else if (!strcmp (derr, DBUS_ERROR_NO_MEMORY ))
        return DBUS_GERROR_NO_MEMORY ;
    else if (!strcmp (derr, DBUS_ERROR_SERVICE_UNKNOWN ))
        return DBUS_GERROR_SERVICE_UNKNOWN ;
    else if (!strcmp (derr, DBUS_ERROR_NAME_HAS_NO_OWNER ))
        return DBUS_GERROR_NAME_HAS_NO_OWNER ;
    else if (!strcmp (derr, DBUS_ERROR_NO_REPLY ))
        return DBUS_GERROR_NO_REPLY ;
}

```

```

else if (!strcmp (derr, DBUS_ERROR_IO_ERROR ))
    return DBUS_GERROR_IO_ERROR ;
else if (!strcmp (derr, DBUS_ERROR_BAD_ADDRESS ))
    return DBUS_GERROR_BAD_ADDRESS ;
else if (!strcmp (derr, DBUS_ERROR_NOT_SUPPORTED ))
    return DBUS_GERROR_NOT_SUPPORTED ;
else if (!strcmp (derr, DBUS_ERROR_LIMITS_EXCEEDED ))
    return DBUS_GERROR_LIMITS_EXCEEDED ;
else if (!strcmp (derr, DBUS_ERROR_ACCESS_DENIED ))
    return DBUS_GERROR_ACCESS_DENIED ;
else if (!strcmp (derr, DBUS_ERROR_AUTH_FAILED ))
    return DBUS_GERROR_AUTH_FAILED ;
else if (!strcmp (derr, DBUS_ERROR_NO_SERVER ))
    return DBUS_GERROR_NO_SERVER ;
else if (!strcmp (derr, DBUS_ERROR_TIMEOUT ))
    return DBUS_GERROR_TIMEOUT ;
else if (!strcmp (derr, DBUS_ERROR_NO_NETWORK ))
    return DBUS_GERROR_NO_NETWORK ;
else if (!strcmp (derr, DBUS_ERROR_ADDRESS_IN_USE ))
    return DBUS_GERROR_ADDRESS_IN_USE ;
else if (!strcmp (derr, DBUS_ERROR_DISCONNECTED ))
    return DBUS_GERROR_DISCONNECTED ;
else if (!strcmp (derr, DBUS_ERROR_INVALID_ARGS ))
    return DBUS_GERROR_INVALID_ARGS ;
else if (!strcmp (derr, DBUS_ERROR_FILE_NOT_FOUND ))
    return DBUS_GERROR_FILE_NOT_FOUND ;
else if (!strcmp (derr, DBUS_ERROR_FILE_EXISTS ))
    return DBUS_GERROR_FILE_EXISTS ;
else if (!strcmp (derr, DBUS_ERROR_UNKNOWN_METHOD ))
    return DBUS_GERROR_UNKNOWN_METHOD ;
else if (!strcmp (derr, DBUS_ERROR_TIMED_OUT ))
    return DBUS_GERROR_TIMED_OUT ;
else if (!strcmp (derr, DBUS_ERROR_MATCH_RULE_NOT_FOUND ))
    return DBUS_GERROR_MATCH_RULE_NOT_FOUND ;
else if (!strcmp (derr, DBUS_ERROR_MATCH_RULE_INVALID ))
    return DBUS_GERROR_MATCH_RULE_INVALID ;
else if (!strcmp (derr, DBUS_ERROR_SPAWN_EXEC_FAILED ))
    return DBUS_GERROR_SPAWN_EXEC_FAILED ;
else if (!strcmp (derr, DBUS_ERROR_SPAWN_FORK_FAILED ))
    return DBUS_GERROR_SPAWN_FORK_FAILED ;
else if (!strcmp (derr, DBUS_ERROR_SPAWN_CHILD_EXITED ))
    return DBUS_GERROR_SPAWN_CHILD_EXITED ;
else if (!strcmp (derr, DBUS_ERROR_SPAWN_CHILD_SIGNALED ))
    return DBUS_GERROR_SPAWN_CHILD_SIGNALED ;
else if (!strcmp (derr, DBUS_ERROR_SPAWN_FAILED ))
    return DBUS_GERROR_SPAWN_FAILED ;
else if (!strcmp (derr, DBUS_ERROR_UNIX_PROCESS_ID_UNKNOWN ))
    return DBUS_GERROR_UNIX_PROCESS_ID_UNKNOWN ;
else if (!strcmp (derr, DBUS_ERROR_INVALID_SIGNATURE ))
    return DBUS_GERROR_INVALID_SIGNATURE ;
else if (!strcmp (derr, DBUS_ERROR_INVALID_FILE_CONTENT ))
    return DBUS_GERROR_INVALID_FILE_CONTENT ;

```

```

    else if (!strcmp (derr, DBUS_ERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN
))
        return DBUS_GERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN ;
    else
        return DBUS_GERROR_REMOTE_EXCEPTION;
}

/**
 * dbus_set_g_error:
 * @gerror: an error
 * @derror: a #DBusError
 *
 * Store the information from a Dbus method error return into a
 * GError. For the normal case of an arbitrary remote process,
 * the error code will be DBUS_GERROR_REMOTE_EXCEPTION. Now,
 * Dbus errors have two components; a message and a "name".
 * The former is an arbitrary (normally American English) string.
 * The second is a string like com.example.FooFailure which
 * programs can use as a conditional source. Because a GError
 * only has one string, we use a hack to encode both values:
 *
 * <human readable string>&null&error
name&null&
 *
 * You can use the following code to retrieve both values:
 *
 * |[const char *msg = error->message;
 * size_t len = strlen(msg);
 * const char *error_name = msg+len+1;]|
 */
void
dbus_set_g_error (GError **gerror,
                 DBusError *derror)
{
    int code;

    g_return_if_fail (derror != NULL);
    g_return_if_fail (dbus_error_is_set (derror));
    g_return_if_fail (gerror == NULL || *gerror == NULL);

    code = dbus_error_to_gerror_code (derror->name);
    if (code != DBUS_GERROR_REMOTE_EXCEPTION)
        g_set_error (gerror, DBUS_GERROR,
                    code,
                    "%s",
                    derror->message);
    else
        g_set_error (gerror, DBUS_GERROR,
                    code,
                    "%s%c%s",
                    derror->message ? derror->message : "",
                    '\\0',

```

```

        derror->name);
    }

static void
dbus_g_error_info_free (gpointer p)
{
    DBusGErrorInfo *info;

    info = p;

    g_free (info->default_iface);
    g_free (info);
}

/**
 * SECTION:dbus-gobject
 * @short_description: Exporting a #GObject remotely
 * @see_also: #GObject
 * @stability: Stable
 *
 *
 * FIXME
 */

/**
 * dbus_glib_global_set_disable_legacy_property_access:
 *
 * For historical reasons, Dbus-GLib will allow read-only
 * access to every GObject property of an object exported
 * to the bus, regardless of whether or not the property
 * is listed in the type info installed with
 * dbus_g_object_type_install_info(). (Write access is
 * denied however).
 *
 * If you wish to restrict even read-only access, you
 * can call this method to globally change the behavior
 * for the entire process.
 *
 * Since: 0.88
 */
void
dbus_glib_global_set_disable_legacy_property_access (void)
{
    disable_legacy_property_access = TRUE;
}

/**
 * dbus_g_object_type_install_info:
 * @object_type: #GType for the object
 * @info: introspection data generated by #dbus-glib-tool
 *
 * Install introspection information about the given object #GType
 * sufficient to allow methods on the object to be invoked by name.

```

```

* The introspection information is normally generated by
* dbus-glib-tool, then this function is called in the
* class_init() for the object class.
*
* Once introspection information has been installed, instances of the
* object registered with dbus_g_connection_register_g_object() can
have
* their methods invoked remotely.
*/
void
dbus_g_object_type_install_info (GType                object_type,
                                const DBusGObjectInfo *info)
{
    g_return_if_fail (G_TYPE_IS_CLASSED (object_type) ||
G_TYPE_IS_INTERFACE (object_type));

    _dbus_g_value_types_init ();

    g_type_set_qdata (object_type,
                    dbus_g_object_type_dbus_metadata_quark (),
                    (gpointer) info);
}

/**
 * dbus_g_error_domain_register:
 * @domain: the #GError domain
 * @default_iface: the prefix used for error values, or %NULL
 * @code_enum: a #GType for a #GEnum of the error codes
 *
 * Register a #GError domain and set of codes with D-Bus. When an
object
 * raises a #GError in the domain @domain from one of its D-Bus
methods,
 * the D-Bus error name used will be @default_iface, followed by a
dot,
 * followed by the #GEnumValue.value_nick corresponding to the
#GError.code.
 * For D-Bus, it's conventional to use an error name (value_nick) that
is
 * in CamelCase.
 *
 * (For instance, if a D-Bus method
<code>com.example.MyObject.GetThings</code>
 * can raise a #GError with domain <code>MY_ERROR</code> and code
 * <code>MY_ERROR_NOT_HAPPY</code>, you could call
 * <code>dbus_g_error_domain_register (MY_ERROR,
"com.example.MyError",
 * MY_TYPE_ERROR)</code>, and set up the value_nick for
 * <code>MY_ERROR_NOT_HAPPY</code> to be <code>NotHappy</code>,
 * resulting in the D-Bus error string
 * <code>com.example.MyError.NotHappy</code>.)
 *
 */

```



```

* If @default_iface is %NULL, the D-Bus interface of the method that
failed
* will be used.
*
* (For instance, if the above example had called
* <code>dbus_g_error_domain_register (MY_ERROR, NULL,
MY_TYPE_ERROR)</code>
* instead, then the D-Bus error string would be
* <code>com.example.MyObject.NotHappy</code>.)
*/
void
dbus_g_error_domain_register (GQuark                domain,
                             const char            *default_iface,
                             GType                code_enum)
{
    DBusGErrorInfo *info;

    g_return_if_fail (g_quark_to_string (domain) != NULL);
    g_return_if_fail (code_enum != G_TYPE_INVALID);
    g_return_if_fail (G_TYPE_FUNDAMENTAL (code_enum) == G_TYPE_ENUM);

    g_static_rw_lock_writer_lock (&globals_lock);

    if (error_metadata == NULL)
        g_datalist_init (&error_metadata);

    info = g_datalist_id_get_data (&error_metadata, domain);

    if (info != NULL)
    {
        g_warning ("Metadata for error domain \"%s\" already
registered\n",
                  g_quark_to_string (domain));
    }
    else
    {
        info = g_new0 (DBusGErrorInfo, 1);
        info->default_iface = g_strdup (default_iface);
        info->code_enum = code_enum;

        g_datalist_id_set_data_full (&error_metadata,
                                     domain,
                                     info,
                                     dbus_g_error_info_free);
    }

    g_static_rw_lock_writer_unlock (&globals_lock);
}

/* Called when the object is destroyed */
static void
object_export_object_died (gpointer user_data, GObject *dead)

```

```

{
    ObjectExport *oe = user_data;

    g_assert (dead == oe->object);

    /* this prevents the weak unref from taking place, which would cause
an
    * assertion failure since the object has already gone... */
    oe->object = NULL;

    /* ... while this results in a call to
object_registration_unregister
    * for each contained registration */
    object_export_unregister_all (oe);

    /* We deliberately don't remove the ObjectExport yet, in case the
object is
    * resurrected and re-registered: if that happens, we wouldn't want
to call
    * export_signals() again. */
}

/**
 * dbus_g_connection_unregister_g_object:
 * @connection: the D-BUS connection
 * @object: the object
 *
 * Removes @object from any object paths at which it is exported on
 * @connection. Properties, methods, and signals
 * of the object can no longer be accessed remotely.
 */
void
dbus_g_connection_unregister_g_object (DBusGConnection *connection,
                                       GObject *object)
{
    ObjectExport *oe;
    GSList *registrations;

    g_return_if_fail (connection != NULL);
    g_return_if_fail (G_IS_OBJECT (object));

    oe = g_object_get_data (object, "dbus_glib_object_registrations");

    g_return_if_fail (oe != NULL);
    g_return_if_fail (oe->registrations != NULL);

    /* Copy the list before iterating it: it will be modified in
    * object_registration_free() each time an object path is
unregistered.
    */
    for (registrations = g_slist_copy (oe->registrations);
         registrations != NULL;

```

```

        registrations = g_slist_delete_link (registrations,
registrations))
    {
        ObjectRegistration *o = registrations->data;

        if (o->connection != connection)
            continue;

        dbus_connection_unregister_object_path
(DBUS_CONNECTION_FROM_G_CONNECTION (o->connection),
        o->object_path);
    }
}

/**
 * dbus_g_connection_register_g_object:
 * @connection: the D-BUS connection
 * @at_path: the path where the object will live (the object's name)
 * @object: the object
 *
 * Registers a #GObject at the given path. Properties, methods, and
signals
 * of the object can then be accessed remotely. Methods are only
available
 * if method introspection data has been added to the object's class
 * with dbus_g_object_type_install_info().
 *
 * The registration will be cancelled if either the #DBusConnection or
 * the #GObject gets finalized, or if
dbus_g_connection_unregister_g_object()
 * is used.
 *
 * Note: If an object is registered multiple times, the first
registration
 * takes priority for cases such as turning an object into an object
path.
 */
void
dbus_g_connection_register_g_object (DBusGConnection
*connection,
                                     const char
                                     GObject
                                     *at_path,
                                     *object)
{
    ObjectExport *oe;
    GSList *iter;
    ObjectRegistration *o;
    DBusError error;

    g_return_if_fail (connection != NULL);
    g_return_if_fail (g_variant_is_object_path (at_path));
    g_return_if_fail (G_IS_OBJECT (object));

```

```

oe = g_object_get_data (object, "dbus_glib_object_registrations");

if (oe == NULL)
{
    GList *info_list = lookup_object_info (object);

    if (info_list == NULL)
    {
        g_warning ("No introspection data registered for object
class \"%s\"",
                    g_type_name (G_TYPE_FROM_INSTANCE (object)));
        return;
    }

    /* This adds a hook into every signal for the object. Only do
this
    * on the first registration, because inside the signal
marshaller
    * we emit a signal for each registration.
    */
    export_signals (info_list, object);
    g_list_free (info_list);

    oe = object_export_new ();
    g_object_set_data_full (object,
"dbus_glib_object_registrations", oe,
(GDestroyNotify) object_export_free);
}

if (oe->object == NULL)
{
    /* Either the ObjectExport is newly-created, or it already
existed but
    * the object was disposed and resurrected, causing the weak ref
to
    * fall off */
    oe->object = object;
    g_object_weak_ref (object, object_export_object_died, oe);
}

for (iter = oe->registrations; iter; iter = iter->next)
{
    o = iter->data;

    /* Silently ignore duplicate registrations */
    if (strcmp (o->object_path, at_path) == 0 && o->connection ==
connection)
        return;
}

o = object_registration_new (connection, at_path, oe);

```

```

    dbus_error_init (&error);
    if (!dbus_connection_try_register_object_path
(DBUS_CONNECTION_FROM_G_CONNECTION (connection),
                                                    at_path,
                                                    &gobject_dbus_vtable,
                                                    o,
                                                    &error))
    {
        g_error ("Failed to register GObject with DBusConnection: %s
%s",
                error.name, error.message);
        dbus_error_free (&error);
        object_registration_free (o);
        return;
    }

    oe->registrations = g_slist_append (oe->registrations, o);
}

/**
 * dbus_g_connection_lookup_g_object:
 * @connection: a #DBusGConnection
 * @at_path: path
 *
 * FIXME
 *
 * Returns: the object at path @at_path
 */
GObject *
dbus_g_connection_lookup_g_object (DBusGConnection *connection,
                                   const char *at_path)
{
    gpointer p;
    ObjectRegistration *o;

    g_return_val_if_fail (connection != NULL, NULL);
    g_return_val_if_fail (g_variant_is_object_path (at_path), NULL);

    if (!dbus_connection_get_object_path_data
(DBUS_CONNECTION_FROM_G_CONNECTION (connection), at_path, &p))
        return NULL;

    if (p == NULL)
        return NULL;

    o = p;

    if (o->export->object == NULL)
        return NULL;

    return G_OBJECT (o->export->object);
}

```

```

typedef struct {
    GType    rettype;
    guint    n_params;
    GType    *params;
} DBusGFuncSignature;

static guint
funcsig_hash (gconstpointer key)
{
    const DBusGFuncSignature *sig = key;
    GType *types;
    guint ret;
    guint i;

    ret = sig->rettype;
    types = sig->params;

    for (i = 0; i < sig->n_params; i++)
    {
        ret += (int) (*types);
        types++;
    }

    return ret;
}

static gboolean
funcsig_equal (gconstpointer aval,
               gconstpointer bval)
{
    const DBusGFuncSignature *a = aval;
    const DBusGFuncSignature *b = bval;
    const GType *atypes;
    const GType *btypes;
    guint i;

    if (a->rettype != b->rettype
        || a->n_params != b->n_params)
        return FALSE;

    atypes = a->params;
    btypes = b->params;

    for (i = 0; i < a->n_params; i++)
    {
        if (*btypes != *atypes)
            return FALSE;
        atypes++;
        btypes++;
    }
}

```

```

    return TRUE;
}

static void
funcsig_free (DBusGFuncSignature *sig)
{
    g_free (sig->params);
    g_free (sig);
}

GClosureMarshal
_dbus_gobject_lookup_marshall (GType          rettype,
                               guint          n_params,
                               const GType *param_types)
{
    GClosureMarshal ret;
    DBusGFuncSignature sig;
    GType *params;
    guint i;

    /* Convert to fundamental types */
    rettype = G_TYPE_FUNDAMENTAL (rettype);
    params = g_new (GType, n_params);
    for (i = 0; i < n_params; i++)
        params[i] = G_TYPE_FUNDAMENTAL (param_types[i]);

    sig.rettype = rettype;
    sig.n_params = n_params;
    sig.params = params;

    g_static_rw_lock_reader_lock (&globals_lock);

    if (marshal_table)
        ret = g_hash_table_lookup (marshal_table, &sig);
    else
        ret = NULL;

    g_static_rw_lock_reader_unlock (&globals_lock);

    if (ret == NULL)
    {
        if (rettype == G_TYPE_NONE)
        {
            if (n_params == 0)
                ret = g_cclosure_marshal_VOID__VOID;
            else if (n_params == 1)
            {
                switch (params[0])
                {
                    case G_TYPE_BOOLEAN:
                        ret = g_cclosure_marshal_VOID__BOOLEAN;
                        break;
                }
            }
        }
    }
}

```



```

GArray *types;
GType gtype;

va_start (args, rettype);

types = g_array_new (TRUE, TRUE, sizeof (GType));

while ((gtype = va_arg (args, GType)) != G_TYPE_INVALID)
    g_array_append_val (types, gtype);

dbus_g_object_register_marshallers_array (marshaller, rettype,
                                           types->len, (GType*) types->data);

g_array_free (types, TRUE);
va_end (args);
}

/**
 * dbus_g_object_register_marshallers_array:
 * @marshaller: a #GClosureMarshal to be used for invocation
 * @rettype: a #GType for the return type of the function
 * @n_types: number of function parameters
 * @types: a C array of GTypes values
 *
 * Register a #GClosureMarshal to be used for signal invocations.
 * @see_also dbus_g_object_register_marshallers()
 */
void
dbus_g_object_register_marshallers_array (GClosureMarshal marshaller,
                                         GType rettype,
                                         guint n_types,
                                         const GType* types)
{
    DBusGFuncSignature *sig;
    guint i;

    g_static_rw_lock_writer_lock (&globals_lock);

    if (marshal_table == NULL)
        marshal_table = g_hash_table_new_full (funcsig_hash,
                                                funcsig_equal,
                                                (GDestroyNotify) funcsig_free,
                                                NULL);

    sig = g_new0 (DBusGFuncSignature, 1);
    sig->rettype = G_TYPE_FUNDAMENTAL (rettype);
    sig->n_params = n_types;
    sig->params = g_new (GType, n_types);
    for (i = 0; i < n_types; i++)
        sig->params[i] = G_TYPE_FUNDAMENTAL (types[i]);

    g_hash_table_insert (marshal_table, sig, marshaller);

```

```

    g_static_rw_lock_writer_unlock (&globals_lock);
}

/**
 * dbus_g_method_get_sender:
 * @context: the method context
 *
 * Get the sender of a message so we can send a
 * "reply" later (i.e. send a message directly
 * to a service which invoked the method at a
 * later time).
 *
 * Returns: the unique name of the sender. It
 * is up to the caller to free the returned string.
 */
gchar *
dbus_g_method_get_sender (DBusGMethodInvocation *context)
{
    const gchar *sender;

    g_return_val_if_fail (context != NULL, NULL);

    sender = dbus_message_get_sender (dbus_g_message_get_message
    (context->message));
    return g_strdup (sender);
}

/**
 * dbus_g_method_get_reply:
 * @context: the method context
 *
 * Get the reply message to append reply values
 * Used as a sidedoor when you can't generate dbus values
 * of the correct type due to glib binding limitations
 *
 * Returns: a #DBusMessage with the reply
 */
DBusMessage *
dbus_g_method_get_reply (DBusGMethodInvocation *context)
{
    g_return_val_if_fail (context != NULL, NULL);

    return reply_or_die (dbus_g_message_get_message (context->message));
}

/**
 * dbus_g_method_send_reply:
 * @context: the method context
 * @reply: the reply message, will be unrefed
 *
 * Send a manually created reply message.
 */

```

```

* Used as a sidedoor when you can't generate dbus values
* of the correct type due to glib binding limitations
*/
void
dbus_g_method_send_reply (DBusGMethodInvocation *context, DBusMessage
*reply)
{
    g_return_if_fail (context != NULL);
    g_return_if_fail (reply != NULL);

    connection_send_or_die (dbus_g_connection_get_connection (context-
>connection),
        reply);
    dbus_message_unref (reply);

    dbus_g_connection_unref (context->connection);
    dbus_g_message_unref (context->message);
    g_free (context);
}

/**
* dbus_g_method_return:
* @context: the method context
* @...: zero or more values to return from the method, with their
number
* and types given by its #DBusGObjectInfo
*
* Send a return message for a given method invocation, with
arguments.
* This function also frees the sending context.
*/
void
dbus_g_method_return (DBusGMethodInvocation *context, ...)
{
    DBusMessage *reply;
    DBusMessageIter iter;
    va_list args;
    char *out_sig;
    GArray *argsig;
    guint i;

    g_return_if_fail (context != NULL);

    /* This field was initialized inside invoke_object_method; we
    * carry it over through the async invocation to here.
    */
    if (!context->send_reply)
        goto out;

    reply = dbus_g_method_get_reply (context);

```

```

    out_sig = method_output_signature_from_object_info (context->object,
context->method);
    argsig = _dbus_gtypes_from_arg_signature (out_sig, FALSE);

    dbus_message_iter_init_append (reply, &iter);

    va_start (args, context);
    for (i = 0; i < argsig->len; i++)
    {
        GValue value = {0,};
        char *error;
        g_value_init (&value, g_array_index (argsig, GType, i));
        error = NULL;
        G_VALUE_COLLECT (&value, args, G_VALUE_NOCOPY_CONTENTS, &error);
        if (error)
        {
            g_warning("%s", error);
            g_free (error);
        }
        else
        {
            if (!_dbus_gvalue_marshal (&iter, &value))
                g_warning ("failed to marshal parameter %d for method %s",
                    i, dbus_message_get_member (
                        dbus_g_message_get_message (context-
>message)));
        }
    }
    va_end (args);

    connection_send_or_die (dbus_g_connection_get_connection (context-
>connection),
        reply);
    dbus_message_unref (reply);

    g_free (out_sig);
    g_array_free (argsig, TRUE);

out:
    dbus_g_connection_unref (context->connection);
    dbus_g_message_unref (context->message);
    g_free (context);
}

/**
 * dbus_g_method_return_error:
 * @context: the method context
 * @error: the error to send
 *
 * Send a error message for a given method invocation.
 * This function also frees the sending context.
 */

```

```

void
dbus_g_method_return_error (DBusGMethodInvocation *context, const
GError *error)
{
    DBusMessage *reply;

    g_return_if_fail (context != NULL);
    g_return_if_fail (error != NULL);

    /* See comment in dbus_g_method_return */
    if (!context->send_reply)
        goto out;

    reply = gerror_to_dbus_error_message (context->object,
dbus_g_message_get_message (context->message), error);
    connection_send_or_die (
        dbus_g_connection_get_connection (context->connection), reply);
    dbus_message_unref (reply);

out:
    dbus_g_connection_unref (context->connection);
    dbus_g_message_unref (context->message);
    g_free (context);
}

const char *
_dbus_gobject_get_path (GObject *obj)
{
    ObjectExport *oe;
    ObjectRegistration *o;

    oe = g_object_get_data (obj, "dbus_glib_object_registrations");

    if (oe == NULL || oe->registrations == NULL)
        return NULL;

    /* First one to have been registered wins */
    o = oe->registrations->data;

    return o->object_path;
}

#ifdef DBUS_BUILD_TESTS
#include <stdlib.h>

static void
_dummy_function (void)
{
}

/* Data structures copied from one generated by current dbus-binding-
tool;

```

```

* we need to support this layout forever
*/
static const DBusGMethodInfo dbus_glib_internal_test_methods[] = {
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 0 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 49 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 117 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 191 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 270 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 320 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 391 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 495 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 623 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 693 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 765 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 838 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 911 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 988 },
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1064
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1140
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1204
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1278
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1347
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1408
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1460
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1533
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1588
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1647
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1730
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1784
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1833
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1895
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1947
},
    { (GCallback) _dummy_function, g_cclosure_marshal_VOID__VOID, 1999
},
};

const DBusGObjectInfo dbus_glib_internal_test_object_info = {

```

```

0,
dbus_glib_internal_test_methods,
30,
"org.freedesktop.DBus.Tests.MyObject\0DoNothing\0S\0\0org.freedesktop.
DBus.Tests.MyObject\0Increment\0S\0x\0I\0u\0arg1\00\0F\0N\0u\0\0org.fr
eedesktop.DBus.Tests.MyObject\0IncrementRetval\0S\0x\0I\0u\0arg1\00\0F
\0R\0u\0\0org.freedesktop.DBus.Tests.MyObject\0IncrementRetValError\0S
\0x\0I\0u\0arg1\00\0F\0E\0u\0\0org.freedesktop.DBus.Tests.MyObject\0Th
rowError\0S\0\0org.freedesktop.DBus.Tests.MyObject\0Uppercase\0S\0arg0
\0I\0s\0arg1\00\0F\0N\0s\0\0org.freedesktop.DBus.Tests.MyObject\0ManyA
rgs\0S\0x\0I\0u\0str\0I\0s\0trouble\0I\0d\0d_ret\00\0F\0N\0d\0str_ret\
00\0F\0N\0s\0\0org.freedesktop.DBus.Tests.MyObject\0ManyReturn\0S\0arg
0\00\0F\0N\0u\0arg1\00\0F\0N\0s\0arg2\00\0F\0N\0i\0arg3\00\0F\0N\0u\0a
rg4\00\0F\0N\0u\0arg5\00\0C\0N\0s\0\0org.freedesktop.DBus.Tests.MyObje
ct\0Stringify\0S\0val\0I\0v\0arg1\00\0F\0N\0s\0\0org.freedesktop.DBus.
Tests.MyObject\0Unstringify\0S\0val\0I\0s\0arg1\00\0F\0N\0v\0\0org.fre
edesktop.DBus.Tests.MyObject\0Recursive1\0S\0arg0\0I\0au\0arg1\00\0F\0
N\0u\0\0org.freedesktop.DBus.Tests.MyObject\0Recursive2\0S\0arg0\0I\0u
\0arg1\00\0F\0N\0au\0\0org.freedesktop.DBus.Tests.MyObject\0ManyUpperc
ase\0S\0arg0\0I\0as\0arg1\00\0F\0N\0as\0\0org.freedesktop.DBus.Tests.M
yObject\0StrHashLen\0S\0arg0\0I\0a{ss}\0arg1\00\0F\0N\0u\0\0org.freede
sktop.DBus.Tests.MyObject\0SendCar\0S\0arg0\0I\0(suv)\0arg1\00\0F\0N\0
(uo)\0\0org.freedesktop.DBus.Tests.MyObject\0GetHash\0S\0arg0\00\0F\0N
\0a{ss}\0\0org.freedesktop.DBus.Tests.MyObject\0RecArrays\0S\0val\0I\0
aas\0arg1\00\0F\0N\0aa\0\0org.freedesktop.DBus.Tests.MyObject\0Objpat
h\0S\0arg0\0I\0o\0arg1\00\0C\0N\0o\0\0org.freedesktop.DBus.Tests.MyObj
ect\0GetObjs\0S\0arg0\00\0F\0N\0ao\0\0org.freedesktop.DBus.Tests.MyObj
ect\0IncrementVal\0S\0\0org.freedesktop.DBus.Tests.MyObject\0AsyncIncr
ement\0A\0x\0I\0u\0arg1\00\0F\0N\0u\0\0org.freedesktop.DBus.Tests.MyOb
ject\0AsyncThrowError\0A\0\0org.freedesktop.DBus.Tests.MyObject\0GetVa
l\0S\0arg0\00\0F\0N\0u\0\0org.freedesktop.DBus.Tests.MyObject\0ManyStr
ingify\0S\0arg0\0I\0a{sv}\0arg1\00\0F\0N\0a{sv}\0\0org.freedesktop.DBu
s.Tests.MyObject\0EmitFrobncate\0S\0\0org.freedesktop.DBus.Tests.MyOb
ject\0Terminate\0S\0\0org.freedesktop.DBus.Tests.FooObject\0GetValue\0
S\0arg0\00\0F\0N\0u\0\0org.freedesktop.DBus.Tests.FooObject\0EmitSigna
ls\0S\0\0org.freedesktop.DBus.Tests.FooObject\0EmitSignal2\0S\0\0org.f
reedesktop.DBus.Tests.FooObject\0Terminate\0S\0\0\0",
"org.freedesktop.DBus.Tests.MyObject\0Frobncate\0org.freedesktop.DBus
.Tests.FooObject\0Sig0\0org.freedesktop.DBus.Tests.FooObject\0Sig1\0or
g.freedesktop.DBus.Tests.FooObject\0Sig2\0\0",
"\0"
};

/*
 * Unit test for GLib GObject integration ("skeletons")
 * Returns: %TRUE on success.
 */
gboolean
_dbus_gobject_test (const char *test_data_dir)
{
    int i;

```

```

const char *arg;
const char *arg_name;
gboolean arg_in;
gboolean constval;
RetValType retval;
const char *arg_signature;
const char *sigdata;
const char *iface;
const char *signame;

static struct { const char *wincaps; const char *uscore; }
name_pairs[] = {
    { "SetFoo", "set_foo" },
    { "Foo", "foo" },
    { "GetFooBar", "get_foo_bar" },
    { "Hello", "hello" }

    /* Impossible-to-handle cases */
    /* { "FrobateUIHandler", "frobate_ui_handler" } */
};

/* Test lookup in our hardcoded object info; if these tests fail
 * then it likely means you changed the generated object info in an
 * incompatible way and broke the lookup functions. In that case
 * you need to bump the version and use a new structure instead. */
/* DoNothing */
arg = method_arg_info_from_object_info
(&dbus_glib_internal_test_object_info,
                                     &(dbus_glib_internal_test_methods[0]));
g_assert (*arg == '\0');

/* Increment */
arg = method_arg_info_from_object_info
(&dbus_glib_internal_test_object_info,
                                     &(dbus_glib_internal_test_methods[1]));
g_assert (*arg != '\0');
arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
g_assert (!strcmp (arg_name, "x"));
g_assert (arg_in == TRUE);
g_assert (!strcmp (arg_signature, "u"));
g_assert (*arg != '\0');
arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
g_assert (arg_in == FALSE);
g_assert (retval == RETVAL_NONE);
g_assert (!strcmp (arg_signature, "u"));
g_assert (*arg == '\0');

/* IncrementRetval */
arg = method_arg_info_from_object_info
(&dbus_glib_internal_test_object_info,

```



```

                                &(dbus_glib_internal_test_methods[2]));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (!strcmp (arg_name, "x"));
    g_assert (arg_in == TRUE);
    g_assert (!strcmp (arg_signature, "u"));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (retval == RETVAL_NOERROR);
    g_assert (arg_in == FALSE);
    g_assert (!strcmp (arg_signature, "u"));
    g_assert (*arg == '\0');

    /* IncrementRetvalError */
    arg = method_arg_info_from_object_info
(&dbus_glib_internal_test_object_info,
                                &(dbus_glib_internal_test_methods[3]));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (!strcmp (arg_name, "x"));
    g_assert (arg_in == TRUE);
    g_assert (!strcmp (arg_signature, "u"));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (retval == RETVAL_ERROR);
    g_assert (arg_in == FALSE);
    g_assert (!strcmp (arg_signature, "u"));
    g_assert (*arg == '\0');

    /* Stringify */
    arg = method_arg_info_from_object_info
(&dbus_glib_internal_test_object_info,
                                &(dbus_glib_internal_test_methods[8]));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (!strcmp (arg_name, "val"));
    g_assert (arg_in == TRUE);
    g_assert (!strcmp (arg_signature, "v"));
    g_assert (*arg != '\0');
    arg = arg_iterate (arg, &arg_name, &arg_in, &constval, &retval,
&arg_signature);
    g_assert (retval == RETVAL_NONE);
    g_assert (arg_in == FALSE);
    g_assert (!strcmp (arg_signature, "s"));
    g_assert (*arg == '\0');

    sigdata = dbus_glib_internal_test_object_info.exported_signals;

```

```

g_assert (*sigdata != '\0');
sigdata = signal_iterate (sigdata, &iface, &signame);
g_assert (!strcmp (iface, "org.freedesktop.DBus.Tests.MyObject"));
g_assert (!strcmp (signame, "Froblicate"));
g_assert (*sigdata != '\0');
sigdata = signal_iterate (sigdata, &iface, &signame);
g_assert (!strcmp (iface, "org.freedesktop.DBus.Tests.FooObject"));
g_assert (!strcmp (signame, "Sig0"));
g_assert (*sigdata != '\0');
sigdata = signal_iterate (sigdata, &iface, &signame);
g_assert (!strcmp (iface, "org.freedesktop.DBus.Tests.FooObject"));
g_assert (!strcmp (signame, "Sig1"));
g_assert (*sigdata != '\0');
sigdata = signal_iterate (sigdata, &iface, &signame);
g_assert (!strcmp (iface, "org.freedesktop.DBus.Tests.FooObject"));
g_assert (!strcmp (signame, "Sig2"));
g_assert (*sigdata == '\0');

i = 0;
while (i < (int) G_N_ELEMENTS (name_pairs))
{
    char *uscore;
    char *wincaps;

    uscore = _dbus_gutils_wincaps_to_uscore (name_pairs[i].wincaps);
    wincaps = uscore_to_wincaps (name_pairs[i].uscore);

    if (strcmp (uscore, name_pairs[i].uscore) != 0)
    {
        g_printerr ("\\"%s\" should have been converted to \"%s\" not
\\"%s\"\\n",
                    name_pairs[i].wincaps, name_pairs[i].uscore,
                    uscore);
        exit (1);
    }

    if (strcmp (wincaps, name_pairs[i].wincaps) != 0)
    {
        g_printerr ("\\"%s\" should have been converted to \"%s\" not
\\"%s\"\\n",
                    name_pairs[i].uscore, name_pairs[i].wincaps,
                    wincaps);
        exit (1);
    }

    g_free (uscore);
    g_free (wincaps);

    ++i;
}

```


G_END_DECLS

#endif

File = dbus-gobject.sgml

<!-- ##### SECTION Title ##### -->
DBus GObject related functions

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->
<para>

</para>

<!-- ##### SECTION See_Also ##### -->
<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### STRUCT DBusGObjectInfo ##### -->
<para>

</para>

@format_version:
@method_infos:
@n_method_infos:
@data:
@exported_signals:
@exported_properties:

<!-- ##### FUNCTION dbus_g_object_type_install_info ##### -->
<para>

</para>

@object_type:

@info:

```
<!-- ##### FUNCTION dbus_g_object_type_register_shadow_property ##### -->
```

<para>

</para>

@iface_type:

@dbus_prop_name:

@shadow_prop_name:

```
<!-- ##### FUNCTION dbus_g_object_path_get_g_type ##### -->
```

<para>

</para>

@void:

@Returns:

```
<!-- ##### FUNCTION dbus_g_object_register_marshallers ##### -->
```

<para>

</para>

@marshaller:

@rettype:

@...:

```
<!-- ##### FUNCTION dbus_g_object_register_marshallers_array ##### -->
```

<para>

</para>

@marshaller:

@rettype:

@n_types:

@types:

```
<!-- ##### FUNCTION
```

```
dbus_glib_global_set_disable_legacy_property_access ##### -->
```

<para>

</para>

@void:

File = dbus-gparser.c

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gparser.c parse Dbus description files
 *
 * Copyright (C) 2003, 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
```

```
#include <config.h>
```

```
#include "dbus-gparser.h"
#include "dbus/dbus-glib-lowlevel.h"
#include "dbus-gidl.h"
#include "dbus-gobject.h"
#include "dbus/dbus-signature.h"
#include <string.h>
```

```
#define ELEMENT_IS(name) (strcmp (element_name, (name)) == 0)
```

```
typedef struct
{
    const char *name;
    const char **retloc;
} LocateAttr;
```

```
static gboolean
locate_attributes (const char *element_name,
                  const char **attribute_names,
```

```

        const char **attribute_values,
        GError      **error,
        const char  *first_attribute_name,
        const char  **first_attribute_retloc,
        ...)
{
    va_list args;
    const char *name;
    const char **retloc;
    int n_attrs;
#define MAX_ATTRS 24
    LocateAttr attrs[MAX_ATTRS];
    gboolean retval;
    int i;

    g_return_val_if_fail (first_attribute_name != NULL, FALSE);
    g_return_val_if_fail (first_attribute_retloc != NULL, FALSE);

    retval = TRUE;

    n_attrs = 1;
    attrs[0].name = first_attribute_name;
    attrs[0].retloc = first_attribute_retloc;
    *first_attribute_retloc = NULL;

    va_start (args, first_attribute_retloc);

    name = va_arg (args, const char*);
    retloc = va_arg (args, const char**);

    while (name != NULL)
    {
        if (retloc == NULL)
        {
            va_end (args);
            return FALSE;
        }

        g_assert (n_attrs < MAX_ATTRS);

        attrs[n_attrs].name = name;
        attrs[n_attrs].retloc = retloc;
        n_attrs += 1;
        *retloc = NULL;

        name = va_arg (args, const char*);
        retloc = va_arg (args, const char**);
    }

    va_end (args);

    i = 0;

```

```

while (attribute_names[i])
{
    int j;
    gboolean found;

    found = FALSE;
    j = 0;
    while (j < n_attrs)
    {
        if (strcmp (attrs[j].name, attribute_names[i]) == 0)
        {
            retloc = attrs[j].retloc;

            if (*retloc != NULL)
            {
                g_set_error (error,
                            G_MARKUP_ERROR,
                            G_MARKUP_ERROR_PARSE,
                            "Attribute \"%s\" repeated twice on the
same <%s> element",
                            attrs[j].name, element_name);
                retval = FALSE;
                goto out;
            }

            *retloc = attribute_values[i];
            found = TRUE;
        }

        ++j;
    }

    if (!found)
    {
        /* We want to passthrough namespaced XML nodes that we don't
know anything about. */
        if (strchr (attribute_names[i], ':') == NULL)
        {
            g_set_error (error,
                        G_MARKUP_ERROR,
                        G_MARKUP_ERROR_PARSE,
                        "Attribute \"%s\" is invalid on <%s>
element in this context",
                        attribute_names[i], element_name);
            retval = FALSE;
            goto out;
        }
    }

    ++i;
}

```



```

out:
    return retval;
}

#if 0
static gboolean
check_no_attributes (const char *element_name,
                    const char **attribute_names,
                    const char **attribute_values,
                    GError **error)
{
    if (attribute_names[0] != NULL)
    {
        g_set_error (error,
                    G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Attribute \"%s\" is invalid on <%s> element in
this context",
                    attribute_names[0], element_name);
        return FALSE;
    }

    return TRUE;
}
#endif

struct Parser
{
    int refcount;

    NodeInfo *result; /* Filled in when we pop the last node */
    GSList *node_stack;
    InterfaceInfo *interface;
    MethodInfo *method;
    SignalInfo *signal;
    PropertyInfo *property;
    ArgInfo *arg;
    gboolean in_annotation;
    quint unknown_namespaced_depth;
};

Parser*
parser_new (void)
{
    Parser *parser;

    parser = g_new0 (Parser, 1);

    parser->refcount = 1;

    return parser;
}

```

```

Parser *
parser_ref (Parser *parser)
{
    parser->refcount += 1;

    return parser;
}

void
parser_unref (Parser *parser)
{
    parser->refcount -= 1;
    if (parser->refcount == 0)
    {
        if (parser->result)
            node_info_unref (parser->result);

        g_free (parser);
    }
}

gboolean
parser_check_doctype (Parser      *parser,
                     const char  *doctype,
                     GError      **error)
{
    g_return_val_if_fail (error == NULL || *error == NULL, FALSE);

    if (strcmp (doctype, "node") != 0)
    {
        g_set_error (error,
                    G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "%s, use node or interface",
                    doctype);

        return FALSE;
    }
    else
        return TRUE;
}

static gboolean
parse_node (Parser      *parser,
            const char  *element_name,
            const char **attribute_names,
            const char **attribute_values,
            GError      **error)
{
    const char *name;
    NodeInfo *node;

```

```

if (parser->interface ||
    parser->method ||
    parser->signal ||
    parser->property ||
    parser->arg ||
    parser->in_annotation)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "Can't put <%s> element here",
                element_name);
    return FALSE;
}

name = NULL;
if (!locate_attributes (element_name, attribute_names,
                       attribute_values, error,
                       "name", &name,
                       NULL))

    return FALSE;

/* Only the root node can have no name */
if (parser->node_stack != NULL && name == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "name", element_name);
    return FALSE;
}

/* Root element name must be absolute */
if (parser->node_stack == NULL && name && *name != '/')
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute on <%s> element must be an
absolute object path, \"%s\" not OK",
                "name", element_name, name);
    return FALSE;
}

/* Other element names must not be absolute */
if (parser->node_stack != NULL && name && *name == '/')
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute on <%s> element must not be an
absolute object path, \"%s\" starts with /",
                "name", element_name, name);
    return FALSE;
}

```

```

    }

    node = node_info_new (name);

    if (parser->node_stack != NULL)
    {
        node_info_add_node (parser->node_stack->data,
                            node);
    }

    parser->node_stack = g_slist_prepend (parser->node_stack,
                                          node);

    return TRUE;
}

static gboolean
parse_interface (Parser      *parser,
                 const char  *element_name,
                 const char **attribute_names,
                 const char **attribute_values,
                 GError      **error)
{
    const char *name;
    InterfaceInfo *iface;
    NodeInfo *top;

    if (parser->interface ||
        parser->method ||
        parser->signal ||
        parser->property ||
        parser->arg ||
        parser->in_annotation ||
        (parser->node_stack == NULL))
    {
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Can't put <%s> element here",
                    element_name);
        return FALSE;
    }

    name = NULL;
    if (!locate_attributes (element_name, attribute_names,
                           attribute_values, error,
                           "name", &name,
                           NULL))

        return FALSE;

    if (name == NULL)
    {
        g_set_error (error, G_MARKUP_ERROR,

```

```

        G_MARKUP_ERROR_PARSE,
        "\"%s\" attribute required on <%s> element ",
        "name", element_name);
    return FALSE;
}

top = parser->node_stack->data;

iface = interface_info_new (name);
node_info_add_interface (top, iface);
interface_info_unref (iface);

parser->interface = iface;

return TRUE;
}

static gboolean
parse_method (Parser      *parser,
              const char  *element_name,
              const char **attribute_names,
              const char **attribute_values,
              GError      **error)
{
    const char *name;
    MethodInfo *method;

    if (parser->interface == NULL ||
        parser->node_stack == NULL ||
        parser->method ||
        parser->signal ||
        parser->property ||
        parser->in_annotation ||
        parser->arg)
    {
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Can't put <%s> element here",
                    element_name);
        return FALSE;
    }

    name = NULL;
    if (!locate_attributes (element_name, attribute_names,
                          attribute_values, error,
                          "name", &name,
                          NULL))

        return FALSE;

    if (name == NULL)
    {
        g_set_error (error, G_MARKUP_ERROR,

```

```

        G_MARKUP_ERROR_PARSE,
        "\\\"%s\" attribute required on <%s> element ",
        "name", element_name);
    return FALSE;
}

method = method_info_new (name);
interface_info_add_method (parser->interface, method);
method_info_unref (method);

parser->method = method;

return TRUE;
}

static gboolean
parse_signal (Parser      *parser,
              const char  *element_name,
              const char  **attribute_names,
              const char  **attribute_values,
              GError      **error)
{
    const char *name;
    SignalInfo *signal;

    if (parser->interface == NULL ||
        parser->node_stack == NULL ||
        parser->signal ||
        parser->method ||
        parser->property ||
        parser->in_annotation ||
        parser->arg)
    {
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Can't put <%s> element here",
                    element_name);
        return FALSE;
    }

    name = NULL;
    if (!locate_attributes (element_name, attribute_names,
                           attribute_values, error,
                           "name", &name,
                           NULL))

        return FALSE;

    if (name == NULL)
    {
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "\\\"%s\" attribute required on <%s> element ",

```

```

        "name", element_name);
    return FALSE;
}

signal = signal_info_new (name);
interface_info_add_signal (parser->interface, signal);
signal_info_unref (signal);

parser->signal = signal;

return TRUE;
}

static gboolean
validate_signature (const char *str,
                   const char *element_name,
                   GError **error)
{
    DBusError derror;

    dbus_error_init (&derror);

    if (!dbus_signature_validate (str, &derror))
    {
        dbus_set_g_error (error, &derror);
        return FALSE;
    }
    return TRUE;
}

static gboolean
parse_property (Parser *parser,
                const char *element_name,
                const char **attribute_names,
                const char **attribute_values,
                GError **error)
{
    const char *name;
    const char *access;
    const char *type;
    PropertyInfo *property;
    PropertyAccessFlags access_flags;

    if (parser->interface == NULL ||
        parser->node_stack == NULL ||
        parser->signal ||
        parser->method ||
        parser->property ||
        parser->in_annotation ||
        parser->arg)
    {
        g_set_error (error, G_MARKUP_ERROR,

```

```

        G_MARKUP_ERROR_PARSE,
        "Can't put <%s> element here",
        element_name);
    return FALSE;
}

name = NULL;
if (!locate_attributes (element_name, attribute_names,
                        attribute_values, error,
                        "name", &name,
                        "access", &access,
                        "type", &type,
                        NULL))

    return FALSE;

if (name == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "name", element_name);
    return FALSE;
}

if (access == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "access", element_name);
    return FALSE;
}

if (type == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "type", element_name);
    return FALSE;
}

if (!validate_signature (type, element_name, error))
    return FALSE;

access_flags = 0;
if (strcmp (access, "readwrite") == 0)
    access_flags = PROPERTY_READ | PROPERTY_WRITE;
else if (strcmp (access, "read") == 0)
    access_flags = PROPERTY_READ;
else if (strcmp (access, "write") == 0)
    access_flags = PROPERTY_WRITE;

```



```

        NULL))
    return FALSE;

/* name can be null for args */
if (type == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "type", element_name);
    return FALSE;
}

if (direction == NULL)
{
    /* methods default to in, signal to out */
    if (parser->method)
        direction = "in";
    else if (parser->signal)
        direction = "out";
    else
        g_assert_not_reached ();
}

dir = ARG_INVALID;

if (strcmp (direction, "in") == 0)
    dir = ARG_IN;
else if (strcmp (direction, "out") == 0)
    dir = ARG_OUT;

if (dir == ARG_INVALID ||
    (parser->signal && dir == ARG_IN))
{
    if (parser->signal)
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Signals must have direction=\"out\" (just omit
the direction attribute)");
    else
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "\"%s\" attribute on <%s> has value \"in\" or
\"out\"",
                    "direction", element_name);
    return FALSE;
}

if (!validate_signature (type, element_name, error))
    return FALSE;

```

```

generated_name = NULL;
if (name == NULL)
    generated_name = g_strdup_printf ("arg%d",
                                     parser->method ?
                                     method_info_get_n_args (parser-
>method) :
                                     signal_info_get_n_args (parser-
>signal));

arg = arg_info_new (name ? name : generated_name, dir, type);
if (parser->method)
    method_info_add_arg (parser->method, arg);
else if (parser->signal)
    signal_info_add_arg (parser->signal, arg);
else
    g_assert_not_reached ();

g_free (generated_name);

arg_info_unref (arg);

parser->arg = arg;

return TRUE;
}

static gboolean
parse_annotation (Parser      *parser,
                  const char  *element_name,
                  const char **attribute_names,
                  const char **attribute_values,
                  GError      **error)
{
    const char *name;
    const char *value;

    if (!(parser->method || parser->interface || parser->arg ||
          parser->property || parser->signal) ||
        parser->node_stack == NULL ||
        parser->in_annotation)
    {
        g_set_error (error, G_MARKUP_ERROR,
                    G_MARKUP_ERROR_PARSE,
                    "Can't put <%s> element here",
                    element_name);
        return FALSE;
    }

    name = NULL;
    if (!locate_attributes (element_name, attribute_names,
                          attribute_values, error,
                          "name", &name,

```

```

        "value", &value,
        NULL))

return FALSE;

if (name == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "name", element_name);
    return FALSE;
}
if (value == NULL)
{
    g_set_error (error, G_MARKUP_ERROR,
                G_MARKUP_ERROR_PARSE,
                "\"%s\" attribute required on <%s> element ",
                "value", element_name);
    return FALSE;
}

if (parser->arg)
    arg_info_add_annotation (parser->arg, name, value);
else if (parser->method)
    method_info_add_annotation (parser->method, name, value);
else if (parser->interface)
    interface_info_add_annotation (parser->interface, name, value);
else
    g_assert_not_reached ();

parser->in_annotation = TRUE;

return TRUE;
}

gboolean
parser_start_element (Parser      *parser,
                    const char  *element_name,
                    const char **attribute_names,
                    const char **attribute_values,
                    GError      **error)
{
    g_return_val_if_fail (error == NULL || *error == NULL, FALSE);

    if (ELEMENT_IS ("node"))
    {
        if (!parse_node (parser, element_name, attribute_names,
                        attribute_values, error))
            return FALSE;
    }
    else if (ELEMENT_IS ("interface"))
    {

```

```

        if (!parse_interface (parser, element_name, attribute_names,
                             attribute_values, error))
            return FALSE;
    }
else if (ELEMENT_IS ("method"))
    {
        if (!parse_method (parser, element_name, attribute_names,
                           attribute_values, error))
            return FALSE;
    }
else if (ELEMENT_IS ("signal"))
    {
        if (!parse_signal (parser, element_name, attribute_names,
                            attribute_values, error))
            return FALSE;
    }
else if (ELEMENT_IS ("property"))
    {
        if (!parse_property (parser, element_name, attribute_names,
                              attribute_values, error))
            return FALSE;
    }
else if (ELEMENT_IS ("arg"))
    {
        if (!parse_arg (parser, element_name, attribute_names,
                        attribute_values, error))
            return FALSE;
    }
else if (ELEMENT_IS ("annotation"))
    {
        if (!parse_annotation (parser, element_name, attribute_names,
                                attribute_values, error))
            return FALSE;
    }
else
    {
        if (strchr (element_name, ':') != NULL)
            /* Passthrough XML-namespaced nodes */
            parser->unknown_namespaced_depth += 1;
        else if (parser->unknown_namespaced_depth == 0)
            {
                g_set_error (error, G_MARKUP_ERROR,
                             G_MARKUP_ERROR_PARSE,
                             "Element <%s> not recognized",
                             element_name);
                return FALSE;
            }
    }

return TRUE;
}

```

```

gboolean
parser_end_element (Parser      *parser,
                   const char  *element_name,
                   GError      **error)
{
    g_return_val_if_fail (error == NULL || *error == NULL, FALSE);

    if (ELEMENT_IS ("interface"))
    {
        parser->interface = NULL;
    }
    else if (ELEMENT_IS ("method"))
    {
        parser->method = NULL;
    }
    else if (ELEMENT_IS ("signal"))
    {
        parser->signal = NULL;
    }
    else if (ELEMENT_IS ("property"))
    {
        parser->property = NULL;
    }
    else if (ELEMENT_IS ("arg"))
    {
        parser->arg = NULL;
    }
    else if (ELEMENT_IS ("annotation"))
    {
        parser->in_annotation = FALSE;
    }
    else if (ELEMENT_IS ("node"))
    {
        NodeInfo *top;

        g_assert (parser->node_stack != NULL);
        top = parser->node_stack->data;

        parser->node_stack = g_slist_remove (parser->node_stack,
                                             top);

        if (parser->node_stack == NULL)
            parser->result = top; /* We are done, store the result */
    }
    else if (strchr (element_name, ':') != NULL)
    {
        /* Passthrough XML-namespaced nodes */
        parser->unknown_namespaced_depth -= 1;
    }
    else if (parser->unknown_namespaced_depth > 0)
    {
        /* pass through unknown elements underneath a namespace */
    }
}

```

```

    }
    else
        g_assert_not_reached (); /* should have had an error on
start_element */

    return TRUE;
}

gboolean
parser_content (Parser      *parser,
                const char *content,
                int         len,
                GError      **error)
{
    g_return_val_if_fail (error == NULL || *error == NULL, FALSE);

    /* FIXME check that it's all whitespace */

    return TRUE;
}

gboolean
parser_finished (Parser      *parser,
                 GError      **error)
{
    g_return_val_if_fail (error == NULL || *error == NULL, FALSE);

    return TRUE;
}

NodeInfo*
parser_get_nodes (Parser *parser)
{
    return parser->result;
}

```

File = dbus-gparser.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gparser.h parse Dbus description files
 *
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or

```

```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_GLIB_PARSER_H
#define DBUS_GLIB_PARSER_H

#include <dbus/dbus.h>
#include <glib.h>
#include "dbus-gidl.h"

G_BEGIN_DECLS

typedef struct Parser Parser;

Parser* parser_new          (void);
Parser* parser_ref         (Parser *parser);
void    parser_unref       (Parser *parser);
gboolean parser_check_doctype (Parser *parser,
                                const char *doctype,
                                GError **error);
gboolean parser_start_element (Parser *parser,
                                const char *element_name,
                                const char **attribute_names,
                                const char **attribute_values,
                                GError **error);
gboolean parser_end_element  (Parser *parser,
                                const char *element_name,
                                GError **error);
gboolean parser_content      (Parser *parser,
                                const char *content,
                                int len,
                                GError **error);
gboolean parser_finished     (Parser *parser,
                                GError **error);

NodeInfo* description_load_from_file (const char *filename,
                                      GError **error);
NodeInfo* description_load_from_string (const char *str,
                                        int len,
                                        GError **error);

NodeInfo* parser_get_nodes (Parser *parser);

```



```
G_END_DECLS
```

```
#endif /* DBUS_GLIB_GPARSER_H */
```

```
File = dbus-gproxy.c
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
```

```
/* dbus-gproxy.c Proxy for remote objects
```

```
*
```

```
* Copyright (C) 2003, 2004, 2005 Red Hat, Inc.
```

```
* Copyright (C) 2005 Nokia
```

```
*
```

```
* Licensed under the Academic Free License version 2.1
```

```
*
```

```
* This program is free software; you can redistribute it and/or  
modify
```

```
* it under the terms of the GNU General Public License as published  
by
```

```
* the Free Software Foundation; either version 2 of the License, or
```

```
* (at your option) any later version.
```

```
*
```

```
* This program is distributed in the hope that it will be useful,
```

```
* but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
```

```
* GNU General Public License for more details.
```

```
*
```

```
* You should have received a copy of the GNU General Public License
```

```
* along with this program; if not, write to the Free Software
```

```
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
```

```
02110-1301 USA
```

```
*
```

```
*/
```

```
#include <config.h>
```

```
#include <dbus/dbus-glib.h>
```

```
#include <dbus/dbus-glib-lowlevel.h>
```

```
#include <dbus/dbus-signature.h>
```

```
#include "dbus-gutils.h"
```

```
#include "dbus-gsignature.h"
```

```
#include "dbus-gvalue.h"
```

```
#include "dbus-gvalue-utils.h"
```

```
#include "dbus-gobject.h"
```

```
#include <string.h>
```

```
#include <gobject/gvaluecollector.h>
```

```
#include <gio/gio.h>
```

```
#define DBUS_G_PROXY_CALL_TO_ID(x) (GPOINTER_TO_UINT(x))
```

```
#define DBUS_G_PROXY_ID_TO_CALL(x) (GUINTE_TO_POINTER(x))
```

```

#define DBUS_G_PROXY_GET_PRIVATE(o) \
    (G_TYPE_INSTANCE_GET_PRIVATE ((o), DBUS_TYPE_G_PROXY,
DBusGProxyPrivate))

static void oom (void) G_GNUC_NORETURN;
static void
oom (void)
{
    g_error ("no memory");
}

typedef struct _DBusGProxyManager DBusGProxyManager;

typedef struct _DBusGProxyPrivate DBusGProxyPrivate;

struct _DBusGProxyPrivate
{
    DBusGProxyManager *manager; /**< Proxy manager */
    char *name;                /**< Name messages go to or NULL */
    char *path;                /**< Path messages go to or NULL */
    char *interface;          /**< Interface messages go to or NULL */

    DBusGProxyCall *name_call; /**< Pending call id to retrieve name
owner */
    guint for_owner : 1;      /**< Whether or not this proxy is for a
name owner */
    guint associated : 1;     /**< Whether or not this proxy is
associated (for name proxies) */

    /* FIXME: make threadsafe? */
    guint call_id_counter;   /**< Integer counter for pending calls
*/

    GData *signal_signatures; /**< D-BUS signatures for each signal */

    GHashTable *pending_calls; /**< Calls made on this proxy which have
not yet returned */

    int default_timeout; /**< Default timeout to use, see
dbus_g_proxy_set_default_timeout */
};

static void dbus_g_proxy_init          (DBusGProxy      *proxy);
static void dbus_g_proxy_class_init   (DBusGProxyClass *klass);
static GObject *dbus_g_proxy_constructor (GType
type,
                                guint
n_construct_properties,
                                GObjectConstructParam
*construct_properties);
static void      dbus_g_proxy_set_property (GObject
*object,

```

```

                                guint                prop_id,
                                const GValue          *value,
                                GParamSpec           *pspec);
static void    dbus_g_proxy_get_property      (GObject
*object,
                                guint                prop_id,
                                GValue             *value,
                                GParamSpec         *pspec);

static void dbus_g_proxy_finalize            (GObject      *object);
static void dbus_g_proxy_dispose            (GObject      *object);
static void dbus_g_proxy_destroy            (DBusGProxy   *proxy);
static void dbus_g_proxy_emit_remote_signal (DBusGProxy   *proxy,
                                DBusMessage
*message);

static DBusGProxyCall *manager_begin_bus_call (DBusGProxyManager
*manager,
                                const char          *method,
                                DBusGProxyCallNotify notify,
                                gpointer            data,
                                GDestroyNotify      destroy,
                                GType               first_arg_type,
                                ...);
static guint dbus_g_proxy_begin_call_internal (DBusGProxy
*proxy,
                                const char          *method,
                                DBusGProxyCallNotify notify,
                                gpointer            data,
                                GDestroyNotify      destroy,
                                GValueArray         *args,
                                int timeout );
static gboolean dbus_g_proxy_end_call_internal (DBusGProxy
*proxy,
                                guint                call_id,
                                GError              **error,
                                GType               first_arg_type,
                                va_list             args);

/*
 * A list of proxies with a given name+path+interface, used to
 * route incoming signals.
 */
typedef struct
{
    GSList *proxies; /**< The list of proxies */

    char name[4]; /**< name (empty string for none), nul byte,
 * path, nul byte,
 * interface, nul byte
 */

```

```

} DBusGProxyList;

/*
 * DBusGProxyManager's primary task is to route signals to the proxies
 * those signals are emitted on. In order to do this it also has to
 * track the owners of the names proxies are bound to.
 */
struct _DBusGProxyManager
{
    GStaticMutex lock; /**< Thread lock */
    int refcount;      /**< Reference count */
    DBusConnection *connection; /**< Connection we're associated with.
*/

    DBusGProxy *bus_proxy; /**< Special internal proxy used to talk to
the bus */

    GHashTable *proxy_lists; /**< Hash used to route incoming signals
* and iterate over proxies
* tristring -> DBusGProxyList
*/

    GHashTable *owner_match_rules; /**< Hash to keep track of match
rules of
* NameOwnerChanged.
* gchar *name -> guint *refcount
*/

    GHashTable *owner_names; /**< Hash to keep track of mapping from
* char * -> GSList of
DBusGProxyNameOwnerInfo
* base name -> [name,name,...] for proxies which
* are for names.
*/

    GSList *unassociated_proxies; /**< List of name proxies for
which
* there was no result for
* GetNameOwner
*/
};

static DBusGProxyManager *dbus_g_proxy_manager_ref
(DBusGProxyManager *manager);
static DBusHandlerResult dbus_g_proxy_manager_filter (DBusConnection
*connection,
                                                    DBusMessage
*message,
                                                    void
*user_data);

/** Lock the DBusGProxyManager */
#define LOCK_MANAGER(mgr) (g_static_mutex_lock (&(mgr)->lock))
/** Unlock the DBusGProxyManager */

```

```

#define UNLOCK_MANAGER(mgr) (g_static_mutex_unlock (&(mgr)->lock))

static int g_proxy_manager_slot = -1;

/* Lock controlling get/set manager as data on each connection */
static GStaticMutex connection_g_proxy_lock = G_STATIC_MUTEX_INIT;

static DBusGProxyManager*
dbus_g_proxy_manager_get (DBusConnection *connection)
{
    DBusGProxyManager *manager;

    dbus_connection_allocate_data_slot (&g_proxy_manager_slot);
    if (g_proxy_manager_slot < 0)
        g_error ("out of memory");

    g_static_mutex_lock (&connection_g_proxy_lock);

    manager = dbus_connection_get_data (connection,
g_proxy_manager_slot);
    if (manager != NULL)
    {
        dbus_connection_free_data_slot (&g_proxy_manager_slot);
        dbus_g_proxy_manager_ref (manager);
        g_static_mutex_unlock (&connection_g_proxy_lock);
        return manager;
    }

    manager = g_new0 (DBusGProxyManager, 1);

    manager->refcount = 1;
    manager->connection = connection;

    g_static_mutex_init (&manager->lock);

    /* Proxy managers keep the connection alive, which means that
     * DBusGProxy indirectly does. To free a connection you have to free
     * all the proxies referring to it.
     */
    dbus_connection_ref (manager->connection);

    dbus_connection_set_data (connection, g_proxy_manager_slot,
                             manager, NULL);

    dbus_connection_add_filter (connection, dbus_g_proxy_manager_filter,
                               manager, NULL);

    g_static_mutex_unlock (&connection_g_proxy_lock);

    return manager;
}

```

```

static DBusGProxyManager *
dbus_g_proxy_manager_ref (DBusGProxyManager *manager)
{
    g_assert (manager != NULL);
    g_assert (manager->refcount > 0);

    LOCK_MANAGER (manager);

    manager->refcount += 1;

    UNLOCK_MANAGER (manager);

    return manager;
}

static void
dbus_g_proxy_manager_unref (DBusGProxyManager *manager)
{
    g_assert (manager != NULL);
    g_assert (manager->refcount > 0);

    LOCK_MANAGER (manager);
    manager->refcount -= 1;
    if (manager->refcount == 0)
    {
        UNLOCK_MANAGER (manager);

        if (manager->bus_proxy)
            g_object_unref (manager->bus_proxy);

        if (manager->proxy_lists)
        {
            /* can't have any proxies left since they hold
             * a reference to the proxy manager.
             */
            g_assert (g_hash_table_size (manager->proxy_lists) == 0);

            g_hash_table_destroy (manager->proxy_lists);
            manager->proxy_lists = NULL;
        }

        if (manager->owner_match_rules)
        {
            /* Since we destroyed all proxies, none can be tracking
             * name owners
             */
            g_assert (g_hash_table_size (manager->owner_match_rules) ==
0);

            g_hash_table_destroy (manager->owner_match_rules);
            manager->owner_match_rules = NULL;
        }
    }
}

```

```

    if (manager->owner_names)
    {
        /* Since we destroyed all proxies, none can be tracking
         * name owners
         */
        g_assert (g_hash_table_size (manager->owner_names) == 0);

        g_hash_table_destroy (manager->owner_names);
        manager->owner_names = NULL;
    }

    g_assert (manager->unassociated_proxies == NULL);

    g_static_mutex_free (&manager->lock);

    g_static_mutex_lock (&connection_g_proxy_lock);

    dbus_connection_remove_filter (manager->connection,
dbus_g_proxy_manager_filter,
                                manager);

    dbus_connection_set_data (manager->connection,
                              g_proxy_manager_slot,
                              NULL, NULL);

    g_static_mutex_unlock (&connection_g_proxy_lock);

    dbus_connection_unref (manager->connection);
    g_free (manager);

    dbus_connection_free_data_slot (&g_proxy_manager_slot);
}
else
{
    UNLOCK_MANAGER (manager);
}
}

static guint
tristring_hash (gconstpointer key)
{
    const char *p = key;
    guint h = *p;

    if (h)
    {
        for (p += 1; *p != '\0'; p++)
            h = (h << 5) - h + *p;
    }

    /* skip nul and do the next substring */

```

```

for (p += 1; *p != '\0'; p++)
    h = (h << 5) - h + *p;

/* skip nul again and another substring */
for (p += 1; *p != '\0'; p++)
    h = (h << 5) - h + *p;

return h;
}

static gboolean
strequal_len (const char *a,
              const char *b,
              size_t *lenp)
{
    size_t a_len;
    size_t b_len;

    a_len = strlen (a);
    b_len = strlen (b);

    if (a_len != b_len)
        return FALSE;

    if (memcmp (a, b, a_len) != 0)
        return FALSE;

    *lenp = a_len;

    return TRUE;
}

static gboolean
tristring_equal (gconstpointer a,
                gconstpointer b)
{
    const char *ap = a;
    const char *bp = b;
    size_t len;

    if (!strequal_len (ap, bp, &len))
        return FALSE;

    ap += len + 1;
    bp += len + 1;

    if (!strequal_len (ap, bp, &len))
        return FALSE;

    ap += len + 1;
    bp += len + 1;

```



```

    if (strcmp (ap, bp) != 0)
        return FALSE;

    return TRUE;
}

static char*
tristring_alloc_from_strings (size_t padding_before,
                             const char *name,
                             const char *path,
                             const char *interface)
{
    size_t name_len, iface_len, path_len, len;
    char *tri;

    if (name)
        name_len = strlen (name);
    else
        name_len = 0;

    path_len = strlen (path);

    iface_len = strlen (interface);

    tri = g_malloc (padding_before + name_len + path_len + iface_len +
3);

    len = padding_before;

    if (name)
        memcpy (&tri[len], name, name_len);

    len += name_len;
    tri[len] = '\0';
    len += 1;

    g_assert (len == (padding_before + name_len + 1));

    memcpy (&tri[len], path, path_len);
    len += path_len;
    tri[len] = '\0';
    len += 1;

    g_assert (len == (padding_before + name_len + path_len + 2));

    memcpy (&tri[len], interface, iface_len);
    len += iface_len;
    tri[len] = '\0';
    len += 1;

    g_assert (len == (padding_before + name_len + path_len + iface_len +
3));
}

```

```

    return tri;
}

static char*
tristring_from_proxy (DBusGProxy *proxy)
{
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    return tristring_alloc_from_strings (0,
                                        priv->name,
                                        priv->path,
                                        priv->interface);
}

static char*
tristring_from_message (DBusMessage *message)
{
    const char *path;
    const char *interface;

    path = dbus_message_get_path (message);
    interface = dbus_message_get_interface (message);

    g_assert (path);
    g_assert (interface);

    return tristring_alloc_from_strings (0,
                                        dbus_message_get_sender
(message),
                                        path, interface);
}

static DBusGProxyList*
g_proxy_list_new (DBusGProxy *first_proxy)
{
    DBusGProxyList *list;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(first_proxy);

    list = (void*) tristring_alloc_from_strings (G_STRUCT_OFFSET
(DBusGProxyList, name),
                                        priv->name,
                                        priv->path,
                                        priv->interface);

    list->proxies = NULL;

    return list;
}

static void
g_proxy_list_free (DBusGProxyList *list)
{

```

```

/* we don't hold a reference to the proxies in the list,
 * as they ref the GProxyManager
 */
g_slist_free (list->proxies);

g_free (list);
}

static char*
g_proxy_get_signal_match_rule (DBusGProxy *proxy)
{
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    /* FIXME Escaping is required here */

    if (priv->name)
        return g_strdup_printf
("type='signal',sender='%s',path='%s',interface='%s'",
                                priv->name, priv->path, priv->interface);
    else
        return g_strdup_printf ("type='signal',path='%s',interface='%s'",
                                priv->path, priv->interface);
}

static char *
get_owner_match_rule (const gchar *name)
{
    return g_strdup_printf ("type='signal',sender='"
DBUS_SERVICE_DBUS
                            "',path='" DBUS_PATH_DBUS
                            "',interface='" DBUS_INTERFACE_DBUS
                            "',member='NameOwnerChanged'"
                            ",arg0='%s'", name);
}

typedef struct
{
    char *name;
    guint refcount;
} DBusGProxyNameOwnerInfo;

static gint
find_name_in_info (gconstpointer a, gconstpointer b)
{
    const DBusGProxyNameOwnerInfo *info = a;
    const char *name = b;

    return strcmp (info->name, name);
}

typedef struct
{
    const char *name;

```

```

    const char *owner;
    DBusGProxyNameOwnerInfo *info;
} DBusGProxyNameOwnerForeachData;

static void
name_owner_foreach (gpointer key, gpointer val, gpointer data)
{
    const char *owner;
    DBusGProxyNameOwnerForeachData *foreach_data;
    GSList *names;
    GSList *link;

    owner = key;
    names = val;
    foreach_data = data;

    if (foreach_data->owner != NULL)
        return;

    g_assert (foreach_data->info == NULL);

    link = g_slist_find_custom (names, foreach_data->name,
find_name_in_info);
    if (link)
    {
        foreach_data->owner = owner;
        foreach_data->info = link->data;
    }
}

static gboolean
dbus_g_proxy_manager_lookup_name_owner (DBusGProxyManager
*manager,
                                     const char          *name,
                                     DBusGProxyNameOwnerInfo **info,
                                     const char          **owner)
{
    DBusGProxyNameOwnerForeachData foreach_data;

    foreach_data.name = name;
    foreach_data.owner = NULL;
    foreach_data.info = NULL;

    g_hash_table_foreach (manager->owner_names, name_owner_foreach,
&foreach_data);

    *info = foreach_data.info;
    *owner = foreach_data.owner;
    return *info != NULL;
}

static void

```

```

insert_nameinfo (DBusGProxyManager      *manager,
                 const char              *owner,
                 DBusGProxyNameOwnerInfo *info)
{
    GSList *names;
    gboolean insert;

    names = g_hash_table_lookup (manager->owner_names, owner);

    /* Only need to g_hash_table_insert the first time */
    insert = (names == NULL);

    names = g_slist_append (names, info);

    if (insert)
        g_hash_table_insert (manager->owner_names, g_strdup (owner),
names);
}

static void
dbus_g_proxy_manager_monitor_name_owner (DBusGProxyManager *manager,
                                         const char          *owner,
                                         const char          *name)
{
    GSList *names;
    GSList *link;
    DBusGProxyNameOwnerInfo *nameinfo;

    names = g_hash_table_lookup (manager->owner_names, owner);
    link = g_slist_find_custom (names, name, find_name_in_info);

    if (!link)
        {
            nameinfo = g_new0 (DBusGProxyNameOwnerInfo, 1);
            nameinfo->name = g_strdup (name);
            nameinfo->refcount = 1;

            insert_nameinfo (manager, owner, nameinfo);
        }
    else
        {
            nameinfo = link->data;
            nameinfo->refcount++;
        }
}

static void
dbus_g_proxy_manager_unmonitor_name_owner (DBusGProxyManager
*manager,
                                         const char          *name)
{
    DBusGProxyNameOwnerInfo *info;

```

```

const char *owner;
gboolean ret;

ret = dbus_g_proxy_manager_lookup_name_owner (manager, name, &info,
&owner);
g_assert (ret);
g_assert (info != NULL);
g_assert (owner != NULL);

info->refcount--;
if (info->refcount == 0)
{
    GSList *names;
    GSList *link;

    names = g_hash_table_lookup (manager->owner_names, owner);
    link = g_slist_find_custom (names, name, find_name_in_info);
    names = g_slist_delete_link (names, link);
    if (names != NULL)
g_hash_table_insert (manager->owner_names, g_strdup (owner),
names);
    else
g_hash_table_remove (manager->owner_names, owner);

    g_free (info->name);
    g_free (info);
}
}

typedef struct
{
    const char *name;
    GSList *destroyed;
} DBusGProxyUnassociateData;

static void
unassociate_proxies (gpointer key, gpointer val, gpointer user_data)
{
    DBusGProxyList *list;
    const char *name;
    GSList *tmp;
    DBusGProxyUnassociateData *data;

    list = val;
    data = user_data;
    name = data->name;

    for (tmp = list->proxies; tmp; tmp = tmp->next)
    {
        DBusGProxy *proxy = DBUS_G_PROXY (tmp->data);
        DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
        DBusGProxyManager *manager;

```

```

manager = priv->manager;

if (priv->name != NULL && !strcmp (priv->name, name))
{
    if (!priv->for_owner)
    {
        /* If a service appeared and then vanished very quickly,
        * it's conceivable we have an inflight request for
        * GetNameOwner here.  Cancel it.
        * https://bugs.freedesktop.org/show_bug.cgi?id=18573
        */
        if (priv->name_call)
            dbus_g_proxy_cancel_call (manager->bus_proxy, priv-
>name_call);

        priv->name_call = NULL;

        priv->associated = FALSE;
        manager->unassociated_proxies = g_slist_prepend (manager-
>unassociated_proxies, proxy);
    }
    else
    {
        data->destroyed = g_slist_prepend (data->destroyed, proxy);
        /* make contents of list into weak pointers in case the
objects
        * unref each other when disposing */
        g_object_add_weak_pointer (G_OBJECT (proxy),
            &(data->destroyed->data));
    }
}
}

static void
dbus_g_proxy_manager_replace_name_owner (DBusGProxyManager *manager,
                                         const char *name,
                                         const char *prev_owner,
                                         const char *new_owner)
{
    GSList *names;

    if (prev_owner[0] == '\\0')
    {
        GSList *tmp;
        GSList *removed;

        /* We have a new service, look at unassociated proxies */

        removed = NULL;

```

```

for (tmp = manager->unassociated_proxies; tmp ; tmp = tmp->next)
{
    DBusGProxy *proxy = tmp->data;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    if (!strcmp (priv->name, name))
    {
        removed = g_slist_prepend (removed, tmp);

        dbus_g_proxy_manager_monitor_name_owner (manager,
new_owner, name);
        priv->associated = TRUE;
    }
}

for (tmp = removed; tmp; tmp = tmp->next)
manager->unassociated_proxies = g_slist_delete_link (manager-
>unassociated_proxies, tmp->data);
g_slist_free (removed);
}
else
{
    DBusGProxyNameOwnerInfo *info;
    GSList *link;

    /* Name owner changed or deleted */

    names = g_hash_table_lookup (manager->owner_names, prev_owner);

    info = NULL;
    if (names != NULL)
    {
        link = g_slist_find_custom (names, name, find_name_in_info);

        if (link != NULL)
        {
            info = link->data;

            names = g_slist_delete_link (names, link);

            if (names == NULL)
            {
                g_hash_table_remove (manager->owner_names,
prev_owner);
            }
            else
            {
                g_hash_table_insert (manager->owner_names,
g_strdup (prev_owner), names);
            }
        }
    }
}
}

```



```

if (new_owner[0] == '\0')
{
    DBusGProxyUnassociateData data;
    GSList *tmp;

    data.name = name;
    data.destroyed = NULL;

    /* A service went away, we need to unassociate proxies */
    g_hash_table_foreach (manager->proxy_lists,
        unassociate_proxies, &data);

    UNLOCK_MANAGER (manager);

    /* the destroyed list's data pointers are weak pointers, so
that we
    * don't end up calling destroy on proxies which have
already been
    * freed up as a result of other ones being destroyed */
    for (tmp = data.destroyed; tmp; tmp = tmp->next)
        if (tmp->data != NULL)
            {
                g_object_remove_weak_pointer (G_OBJECT (tmp->data),
                    &(tmp->data));
                dbus_g_proxy_destroy (tmp->data);
            }
    g_slist_free (data.destroyed);

    LOCK_MANAGER (manager);

    if (info)
        {
            g_free (info->name);
            g_free (info);
        }
    else if (info)
        {
            insert_nameinfo (manager, new_owner, info);
        }
    }
}

static void
got_name_owner_cb (DBusGProxy          *bus_proxy,
                  DBusGProxyCall      *call,
                  void                 *user_data)
{
    DBusGProxy *proxy = user_data;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    GError *error;

```

```

char *owner;

error = NULL;
owner = NULL;

LOCK_MANAGER (priv->manager);

if (!dbus_g_proxy_end_call (bus_proxy, call, &error,
                            G_TYPE_STRING, &owner,
                            G_TYPE_INVALID))
{
    if (error->domain == DBUS_GERROR && error->code ==
        DBUS_GERROR_NAME_HAS_NO_OWNER)
    {
        priv->manager->unassociated_proxies = g_slist_prepend (priv->manager->unassociated_proxies, proxy);
    }
    else if (error->domain == DBUS_GERROR && error->code ==
        DBUS_GERROR_REMOTE_EXCEPTION)
        g_warning ("Couldn't get name owner (%s): %s",
                    dbus_g_error_get_name (error),
                    error->message);
    else
        g_warning ("Couldn't get name owner (code %d): %s",
                    error->code, error->message);
    g_clear_error (&error);
    goto out;
}
else
{
    dbus_g_proxy_manager_monitor_name_owner (priv->manager, owner,
        priv->name);
    priv->associated = TRUE;
}

out:
priv->name_call = NULL;
UNLOCK_MANAGER (priv->manager);
g_free (owner);
}

static char *
get_name_owner (DBusConnection *connection,
                const char *name,
                GError **error)
{
    DBusError derror;
    DBusMessage *request, *reply;
    char *base_name;

    dbus_error_init (&derror);

```

```

base_name = NULL;
reply = NULL;

request = dbus_message_new_method_call (DBUS_SERVICE_DBUS,
                                        DBUS_PATH_DBUS,
                                        DBUS_INTERFACE_DBUS,
                                        "GetNameOwner");

if (request == NULL)
    g_error ("Out of memory");

if (!dbus_message_append_args (request,
                               DBUS_TYPE_STRING, &name,
                               DBUS_TYPE_INVALID))
    g_error ("Out of memory");

reply =
    dbus_connection_send_with_reply_and_block (connection,
                                              request,
                                              2000, &derror);

if (reply == NULL)
    goto error;

if (dbus_set_error_from_message (&derror, reply))
    goto error;

if (!dbus_message_get_args (reply, &derror,
                            DBUS_TYPE_STRING, &base_name,
                            DBUS_TYPE_INVALID))
    goto error;

base_name = g_strdup (base_name);
goto out;

error:
g_assert (dbus_error_is_set (&derror));
dbus_set_g_error (error, &derror);
dbus_error_free (&derror);

out:
if (request)
    dbus_message_unref (request);
if (reply)
    dbus_message_unref (reply);

return base_name;
}

static void
guint_slice_free (gpointer data)
{
    g_slice_free (guint, data);
}

```



```

if (list->proxies == NULL && priv->name)
{
    /* We have to add match rules to the server,
     * but only if the server is a message bus,
     * not if it's a peer.
     */
    char *rule;
    guint *refcount;

    rule = g_proxy_get_signal_match_rule (proxy);
    /* We don't check for errors; it's not like anyone would handle
them, and
     * we don't want a round trip here.
     */
    dbus_bus_add_match (manager->connection, rule, NULL);
    g_free (rule);

    refcount = g_hash_table_lookup (manager->owner_match_rules,
priv->name);

    if (refcount != NULL)
    {
        g_assert (*refcount != 0);
        g_assert (*refcount < G_MAXUINT);
        (*refcount)++;
    }
    else
    {
        char *rule;
        rule = get_owner_match_rule (priv->name);
        dbus_bus_add_match (manager->connection,
                             rule, NULL);

        g_free (rule);

        refcount = g_slice_new (guint);
        *refcount = 1;
        g_hash_table_insert (manager->owner_match_rules,
                             g_strdup (priv->name), refcount);
    }
}

g_assert (g_slist_find (list->proxies, proxy) == NULL);

list->proxies = g_slist_prepend (list->proxies, proxy);

if (!priv->for_owner)
{
    const char *owner;
    DBusGProxyNameOwnerInfo *info;

    if (!dbus_g_proxy_manager_lookup_name_owner (manager, priv-
>name, &info, &owner))

```

```

        {
            priv->name_call = manager_begin_bus_call (manager,
"GetNameOwner",

                                                    got_name_owner_cb,
                                                    proxy, NULL,
                                                    G_TYPE_STRING,
                                                    priv->name,
                                                    G_TYPE_INVALID);

            priv->associated = FALSE;
        }
        else
        {
            info->refcount++;
            priv->associated = TRUE;
        }
    }
}

UNLOCK_MANAGER (manager);
}

static void
dbus_g_proxy_manager_unregister (DBusGProxyManager *manager,
                                DBusGProxy         *proxy)
{
    DBusGProxyList *list;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    char *tri;

    LOCK_MANAGER (manager);

#ifdef G_DISABLE_CHECKS
    if (manager->proxy_lists == NULL)
    {
        g_warning ("Trying to unregister a proxy but there aren't any
registered");
        return;
    }
#endif

    tri = tristring_from_proxy (proxy);

    list = g_hash_table_lookup (manager->proxy_lists, tri);

#ifdef G_DISABLE_CHECKS
    if (list == NULL)
    {
        g_warning ("Trying to unregister a proxy but it isn't
registered");
        return;
    }
#endif
}

```

```

g_assert (g_slist_find (list->proxies, proxy) != NULL);
list->proxies = g_slist_remove (list->proxies, proxy);
g_assert (g_slist_find (list->proxies, proxy) == NULL);
if (!priv->for_owner)
{
    if (!priv->associated)
    {
        GSList *link;

        if (priv->name_call != 0)
        {
            dbus_g_proxy_cancel_call (manager->bus_proxy, priv-
>name_call);
            priv->name_call = 0;
        }
        else
        {
            link = g_slist_find (manager->unassociated_proxies,
proxy);
            if (link != NULL)
            {
                manager->unassociated_proxies = g_slist_delete_link
(
                manager->unassociated_proxies, link);
            }
        }
    }
    else
    {
        g_assert (priv->name_call == 0);

        dbus_g_proxy_manager_unmonitor_name_owner (manager, priv-
>name);
    }
}

if (list->proxies == NULL)
{
    char *rule;
    g_hash_table_remove (manager->proxy_lists,
                        tri);

    rule = g_proxy_get_signal_match_rule (proxy);
    dbus_bus_remove_match (manager->connection,
                        rule, NULL);
    g_free (rule);

    if (priv->name)

```

```

        {
            guint *refcount;
            refcount = g_hash_table_lookup (manager->owner_match_rules,
priv->name);
            (*refcount)--;

            if (*refcount == 0)
            {
                rule = get_owner_match_rule (priv->name);
                dbus_bus_remove_match (manager->connection,
                                     rule, NULL);

                g_free (rule);
                g_hash_table_remove (manager->owner_match_rules, priv-
>name);
            }
        }
    }

    if (g_hash_table_size (manager->proxy_lists) == 0)
    {
        g_hash_table_destroy (manager->proxy_lists);
        manager->proxy_lists = NULL;
    }

    if (manager->owner_match_rules != NULL &&
        g_hash_table_size (manager->owner_match_rules) == 0)
    {
        g_hash_table_destroy (manager->owner_match_rules);
        manager->owner_match_rules = NULL;
    }

    g_free (tri);

    UNLOCK_MANAGER (manager);
}

static void
list_proxies_foreach (gpointer key,
                     gpointer value,
                     gpointer user_data)
{
    DBusGProxyList *list;
    GSList **ret;
    GSList *tmp;

    list = value;
    ret = user_data;

    tmp = list->proxies;
    while (tmp != NULL)
    {
        DBusGProxy *proxy = DBUS_G_PROXY (tmp->data);

```



```

        g_object_ref (proxy);
        *ret = g_slist_prepend (*ret, proxy);

        tmp = tmp->next;
    }
}

static GSList*
dbus_g_proxy_manager_list_all (DBusGProxyManager *manager)
{
    GSList *ret;

    ret = NULL;

    if (manager->proxy_lists)
    {
        g_hash_table_foreach (manager->proxy_lists,
                              list_proxies_foreach,
                              &ret);
    }

    return ret;
}

static DBusHandlerResult
dbus_g_proxy_manager_filter (DBusConnection      *connection,
                             DBusMessage        *message,
                             void                *user_data)
{
    DBusGProxyManager *manager;

    if (dbus_message_get_type (message) != DBUS_MESSAGE_TYPE_SIGNAL)
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

    manager = user_data;

    dbus_g_proxy_manager_ref (manager);

    LOCK_MANAGER (manager);

    if (dbus_message_is_signal (message,
                                DBUS_INTERFACE_LOCAL,
                                "Disconnected"))
    {
        /* Destroy all the proxies, quite possibly resulting in
unreferencing
        * the proxy manager and the connection as well.
        */
        GSList *all;
        GSList *tmp;
    }
}

```

```

all = dbus_g_proxy_manager_list_all (manager);

tmp = all;
while (tmp != NULL)
{
    DBusGProxy *proxy;

    proxy = DBUS_G_PROXY (tmp->data);

    UNLOCK_MANAGER (manager);
    dbus_g_proxy_destroy (proxy);
    g_object_unref (G_OBJECT (proxy));
    LOCK_MANAGER (manager);

    tmp = tmp->next;
}

g_slist_free (all);

#ifdef G_DISABLE_CHECKS
    if (manager->proxy_lists != NULL)
        g_warning ("Disconnection emitted \"destroy\" on all
DBusGProxy, but somehow new proxies were created in response to one of
those destroy signals. This will cause a memory leak.");
#endif
}
else
{
    char *tri;
    GSList *full_list;
    GSList *owned_names;
    GSList *tmp;
    const char *sender;

    sender = dbus_message_get_sender (message);

    /* First we handle NameOwnerChanged internally */
    if (g_strcmp0 (sender, DBUS_SERVICE_DBUS) == 0 &&
        dbus_message_is_signal (message,
                                DBUS_INTERFACE_DBUS,
                                "NameOwnerChanged"))
    {
        const char *name;
        const char *prev_owner;
        const char *new_owner;
        DBusError derr;

        dbus_error_init (&derr);
        if (!dbus_message_get_args (message,
                                    &derr,
                                    DBUS_TYPE_STRING,
                                    &name,

```

```

        DBUS_TYPE_STRING,
        &prev_owner,
        DBUS_TYPE_STRING,
        &new_owner,
        DBUS_TYPE_INVALID))
    {
        /* Ignore this error */
        dbus_error_free (&derr);
    }
    else if (manager->owner_names != NULL)
    {
        dbus_g_proxy_manager_replace_name_owner (manager, name,
prev_owner, new_owner);
    }
}

/* dbus spec requires these, libdbus validates */
g_assert (dbus_message_get_path (message) != NULL);
g_assert (dbus_message_get_interface (message) != NULL);
g_assert (dbus_message_get_member (message) != NULL);

tri = tristring_from_message (message);

if (manager->proxy_lists)
{
    DBusGProxyList *owner_list;
    owner_list = g_hash_table_lookup (manager->proxy_lists, tri);
    if (owner_list)
        full_list = g_slist_copy (owner_list->proxies);
    else
        full_list = NULL;
}
else
full_list = NULL;

g_free (tri);

if (manager->owner_names && sender)
{
    owned_names = g_hash_table_lookup (manager->owner_names,
sender);
    for (tmp = owned_names; tmp; tmp = tmp->next)
    {
        DBusGProxyList *owner_list;
        DBusGProxyNameOwnerInfo *nameinfo;

        nameinfo = tmp->data;
        g_assert (nameinfo->refcount > 0);
        tri = tristring_alloc_from_strings (0, nameinfo->name,
dbus_message_get_path (message),
dbus_message_get_interface
(message));

```

```

        owner_list = g_hash_table_lookup (manager->proxy_lists,
tri);
        if (owner_list != NULL)
            {
                GSList *elt;

                /* Ignore duplicates when adding to full_list */
                for (elt = owner_list->proxies; elt; elt = g_slist_next
(elt))
                    {
                        if (!g_slist_find (full_list, elt->data))
                            full_list = g_slist_append (full_list, elt-
>data);
                    }
            }
        g_free (tri);
    }
}

#ifdef 0
    g_print ("proxy got %s,%s,%s = list %p\n",
        tri,
        tri + strlen (tri) + 1,
        tri + strlen (tri) + 1 + strlen (tri + strlen (tri) +
1) + 1,
        list);
#endif

    /* Emit the signal */

    g_slist_foreach (full_list, (GFunc) g_object_ref, NULL);

    for (tmp = full_list; tmp; tmp = tmp->next)
    {
        DBusGProxy *proxy;

        proxy = DBUS_G_PROXY (tmp->data);

        UNLOCK_MANAGER (manager);
        dbus_g_proxy_emit_remote_signal (proxy, message);
        g_object_unref (G_OBJECT (proxy));
        LOCK_MANAGER (manager);
    }
    g_slist_free (full_list);
}

UNLOCK_MANAGER (manager);
dbus_g_proxy_manager_unref (manager);

/* "Handling" signals doesn't make sense, they are for everyone
* who cares

```

```

    */
    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}

/* ----- DbusGProxy ----- */
#define DBUS_G_PROXY_DESTROYED(proxy)
(DBUS_G_PROXY_GET_PRIVATE(proxy)->manager == NULL)

static void
marshal_dbus_message_to_g_marshaller (GClosure      *closure,
                                       GValue        *return_value,
                                       guint         n_param_values,
                                       const GValue *param_values,
                                       gpointer      invocation_hint,
                                       gpointer      marshal_data);

enum
{
    PROP_0,
    PROP_NAME,
    PROP_PATH,
    PROP_INTERFACE,
    PROP_CONNECTION
};

enum
{
    DESTROY,
    RECEIVED,
    LAST_SIGNAL
};

static void *parent_class;
static guint signals[LAST_SIGNAL] = { 0 };

static void
dbus_g_proxy_init (DBusGProxy *proxy)
{
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    g_datalist_init (&priv->signal_signatures);
    priv->pending_calls = g_hash_table_new_full (NULL, NULL, NULL,
                                                (GDestroyNotify) dbus_pending_call_unref);
    priv->name_call = 0;
    priv->associated = FALSE;
    priv->default_timeout = -1;
}

static GObject *
dbus_g_proxy_constructor (GType          type,
                          guint          n_construct_properties,

```

```

        GObjectConstructParam *construct_properties)
{
    DBusGProxy *proxy;
    DBusGProxyClass *klass;
    GObjectClass *parent_class;
    DBusGProxyPrivate *priv;

    klass = DBUS_G_PROXY_CLASS (g_type_class_peek (DBUS_TYPE_G_PROXY));

    parent_class = G_OBJECT_CLASS (g_type_class_peek_parent (klass));

    proxy = DBUS_G_PROXY (parent_class->constructor (type,
n_construct_properties,
                                construct_properties));

    priv = DBUS_G_PROXY_GET_PRIVATE (proxy);

    /* if these assertions fail, a deriving class has not set our
required
    * parameters - our own public constructors do return_if_fail checks
    * on these parameters being provided. unfortunately we can't assert
    * for manager because it's allowed to be NULL when the manager is
    * setting up a bus proxy for its own calls */
    g_assert (priv->path != NULL);
    g_assert (priv->interface != NULL);

    if (priv->manager != NULL)
    {
        dbus_g_proxy_manager_register (priv->manager, proxy);
    }

    return G_OBJECT (proxy);
}

static void
dbus_g_proxy_class_init (DBusGProxyClass *klass)
{
    GObjectClass *object_class = G_OBJECT_CLASS (klass);

    parent_class = g_type_class_peek_parent (klass);

    g_type_class_add_private (klass, sizeof (DBusGProxyPrivate));

    object_class->set_property = dbus_g_proxy_set_property;
    object_class->get_property = dbus_g_proxy_get_property;

    g_object_class_install_property (object_class,
PROP_NAME,
        g_param_spec_string ("name",
                                "name",
                                "name",
                                NULL,

```

```

                                G_PARAM_READWRITE |
G_PARAM_CONSTRUCT_ONLY));

    g_object_class_install_property (object_class,
                                     PROP_PATH,
                                     g_param_spec_string ("path",
                                                         "path",
                                                         "path",
                                                         NULL,
                                                         G_PARAM_READWRITE |
G_PARAM_CONSTRUCT_ONLY));

    g_object_class_install_property (object_class,
                                     PROP_INTERFACE,
                                     g_param_spec_string ("interface",
                                                         "interface",
                                                         "interface",
                                                         NULL,
                                                         G_PARAM_READWRITE |
G_PARAM_CONSTRUCT_ONLY));

    g_object_class_install_property (object_class,
                                     PROP_CONNECTION,
                                     g_param_spec_boxed ("connection",
                                                         "connection",
                                                         "connection",
                                                         DBUS_TYPE_G_CONNECTION,
                                                         G_PARAM_READWRITE |
G_PARAM_CONSTRUCT_ONLY));

    object_class->finalize = dbus_g_proxy_finalize;
    object_class->dispose = dbus_g_proxy_dispose;
    object_class->constructor = dbus_g_proxy_constructor;

    signals[DESTROY] =
        g_signal_new ("destroy",
                     G_OBJECT_CLASS_TYPE (object_class),
                     G_SIGNAL_RUN_CLEANUP | G_SIGNAL_NO_RECURSE |
G_SIGNAL_NO_HOOKS,
                     0,
                     NULL, NULL,
                     g_cclosure_marshal_VOID__VOID,
                     G_TYPE_NONE, 0);

    signals[RECEIVED] =
        g_signal_new ("received",
                     G_OBJECT_CLASS_TYPE (object_class),
                     G_SIGNAL_RUN_LAST | G_SIGNAL_DETAILED,
                     0,
                     NULL, NULL,
                     marshal_dbus_message_to_g_marshaller,
                     G_TYPE_NONE, 2, DBUS_TYPE_MESSAGE, G_TYPE_POINTER);

```

```

}

static gboolean
cancel_pending_call (gpointer key, gpointer val, gpointer data)
{
    DBusPendingCall *pending = val;

    dbus_pending_call_cancel (pending);

    return TRUE;
}

static void
dbus_g_proxy_dispose (GObject *object)
{
    DBusGProxy *proxy = DBUS_G_PROXY (object);
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    if (priv->pending_calls == NULL)
    {
        return;
    }

    /* Cancel outgoing pending calls */
    g_hash_table_foreach_remove (priv->pending_calls,
cancel_pending_call, NULL);
    g_hash_table_destroy (priv->pending_calls);
    priv->pending_calls = NULL;

    if (priv->manager && proxy != priv->manager->bus_proxy)
    {
        dbus_g_proxy_manager_unregister (priv->manager, proxy);
        dbus_g_proxy_manager_unref (priv->manager);
    }
    priv->manager = NULL;

    g_datalist_clear (&priv->signal_signatures);

    g_signal_emit (object, signals[DESTROY], 0);

    G_OBJECT_CLASS (parent_class)->dispose (object);
}

static void
dbus_g_proxy_finalize (GObject *object)
{
    DBusGProxy *proxy = DBUS_G_PROXY (object);
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    g_return_if_fail (DBUS_G_PROXY_DESTROYED (proxy));

    g_free (priv->name);
}

```



```

g_free (priv->path);
g_free (priv->interface);

G_OBJECT_CLASS (parent_class)->finalize (object);
}

static void
dbus_g_proxy_destroy (DBusGProxy *proxy)
{
    /* FIXME do we need the GTK_IN_DESTRUCTION style flag
     * from GtkWidget?
     */
    g_object_run_dispose (G_OBJECT (proxy));
}

static void
dbus_g_proxy_set_property (GObject *object,
                           guint prop_id,
                           const GValue *value,
                           GParamSpec *pspec)
{
    DBusGProxy *proxy = DBUS_G_PROXY (object);
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    DBusGConnection *connection;

    switch (prop_id)
    {
        {
        case PROP_NAME:
            priv->name = g_strdup (g_value_get_string (value));
            if (priv->name)
                priv->for_owner = (priv->name[0] == ':');
            else
                priv->for_owner = TRUE;
            break;
        case PROP_PATH:
            priv->path = g_strdup (g_value_get_string (value));
            break;
        case PROP_INTERFACE:
            priv->interface = g_strdup (g_value_get_string (value));
            break;
        case PROP_CONNECTION:
            connection = g_value_get_boxed (value);
            if (connection != NULL)
            {
                priv->manager = dbus_g_proxy_manager_get
(DBUS_CONNECTION_FROM_G_CONNECTION (connection));
            }
            break;
        default:
            G_OBJECT_WARN_INVALID_PROPERTY_ID (object, prop_id, pspec);
            break;
        }
    }
}

```

```

}

static void
dbus_g_proxy_get_property (GObject *object,
                           guint prop_id,
                           GValue *value,
                           GParamSpec *pspec)
{
    DBusGProxy *proxy = DBUS_G_PROXY (object);
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    switch (prop_id)
    {
        case PROP_NAME:
            g_value_set_string (value, priv->name);
            break;
        case PROP_PATH:
            g_value_set_string (value, priv->path);
            break;
        case PROP_INTERFACE:
            g_value_set_string (value, priv->interface);
            break;
        case PROP_CONNECTION:
            g_value_set_boxed (value,
                DBUS_G_CONNECTION_FROM_CONNECTION(priv->manager->connection));
            break;
        default:
            G_OBJECT_WARN_INVALID_PROPERTY_ID (object, prop_id, pspec);
            break;
    }
}

/* this is to avoid people using g_signal_connect() directly,
 * to avoid confusion with local signal names, and because
 * of the horribly broken current setup (signals are added
 * globally to all proxies)
 */
static char*
create_signal_name (const char *interface,
                   const char *signal)
{
    GString *str;
    char *p;

    str = g_string_new (interface);

    g_string_append (str, "-");

    g_string_append (str, signal);

    /* GLib will silently barf on '.' in signal names */
    p = str->str;

```

```

while (*p)
{
    if (*p == '.')
        *p = '-';
    ++p;
}

return g_string_free (str, FALSE);
}

static void
marshal_dbus_message_to_g_marshaller (GClosure      *closure,
                                       GValue        *return_value,
                                       guint          n_param_values,
                                       const GValue *param_values,
                                       gpointer       invocation_hint,
                                       gpointer       marshal_data)
{
    /* Incoming here we have three params, the instance (Proxy), the
     * DBusMessage, the signature. We want to convert that to an
     * expanded GValue array, then call an appropriate normal GLib
     * marshaller.
     */
#define MAX_SIGNATURE_ARGS 20
    GValueArray *value_array;
    GSignalCMarshaller c_marshaller;
    DBusGProxy *proxy;
    DBusMessage *message;
    GArray *gsignature;
    const GType *types;
    DBusGProxyPrivate *priv;

    g_assert (n_param_values == 3);

    proxy = g_value_get_object (&param_values[0]);
    message = g_value_get_boxed (&param_values[1]);
    gsignature = g_value_get_pointer (&param_values[2]);

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (message != NULL);
    g_return_if_fail (gsignature != NULL);

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    c_marshaller = _dbus_gobject_lookup_marshaller (G_TYPE_NONE,
                                                    gsignature->len,
                                                    (GType*) gsignature->data);

    g_return_if_fail (c_marshaller != NULL);

    {
        DBusGValueMarshalCtx context;

```

```

    context.recursion_depth = 0;
    context.gconnection = DBUS_G_CONNECTION_FROM_CONNECTION (priv-
>manager->connection);
    context.proxy = proxy;

    types = (const GType*) gsignature->data;
    value_array = _dbus_gvalue_demarshal_message (&context, message,
                                                gsignature->len, types, NULL);
}

if (value_array == NULL)
    return;

g_value_array_prepend (value_array, NULL);
g_value_init (g_value_array_get_nth (value_array, 0),
G_TYPE_FROM_INSTANCE (proxy));
g_value_set_instance (g_value_array_get_nth (value_array, 0),
proxy);

(* c_marshall) (closure, return_value, value_array->n_values,
                value_array->values, invocation_hint, marshal_data);

g_value_array_free (value_array);
}

static void
dbus_g_proxy_emit_remote_signal (DBusGProxy *proxy,
                                DBusMessage *message)
{
    const char *interface;
    const char *signal;
    char *name;
    GQuark q;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    GArray *msg_gsignature = NULL;

    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));

    interface = dbus_message_get_interface (message);
    signal = dbus_message_get_member (message);

    g_assert (interface != NULL);
    g_assert (signal != NULL);

    name = create_signal_name (interface, signal);

    /* If the quark isn't preexisting, there's no way there
     * are any handlers connected. We don't want to create
     * extra quarks for every possible signal.
     */
    q = g_quark_try_string (name);

```

```

if (q != 0)
{
    GArray *gsignature;
    guint i;

    gsignature = g_datalist_id_get_data (&priv->signal_signatures,
q);
    if (gsignature == NULL)
        goto out;

    msg_gsignature = _dbus_gtypes_from_arg_signature
(dbus_message_get_signature (message),
                                TRUE);
    for (i = 0; i < gsignature->len; i++)
    {
        if (msg_gsignature->len == i
            || g_array_index (gsignature, GType, i) != g_array_index
(msg_gsignature, GType, i))
            goto mismatch;
    }
    if (msg_gsignature->len != i)
        goto mismatch;

    g_signal_emit (proxy,
                    signals[RECEIVED],
                    q,
                    message,
                    msg_gsignature);
}

out:
    g_free (name);
    if (msg_gsignature)
        g_array_free (msg_gsignature, TRUE);
    return;
mismatch:
#ifdef 0
    /* Don't spew on remote errors */
    g_warning ("Unexpected message signature '%s' for signal '%s'\n",
                dbus_message_get_signature (message),
                name);
#endif
    goto out;
}

/**
 * DBusGProxyCallNotify:
 * @proxy: the proxy on which the method was called
 * @call_id: the call in progress
 * @user_data: data passed to dbus_g_proxy_begin_call() or similar
 *
 * Called when a reply to the call represented by @call_id arrives.

```

```

* Use dbus_g_proxy_end_call() to see whether @call_id succeeded or
* failed, and get the arguments returned (if any) on success.
*/

typedef struct
{
    DBusGProxy *proxy;
    guint call_id;
    DBusGProxyCallNotify func;
    void *data;
    GDestroyNotify free_data_func;
} GPendingNotifyClosure;

static void
d_pending_call_notify (DBusPendingCall *dcall,
                      void *data)
{
    GPendingNotifyClosure *closure = data;

    (* closure->func) (closure->proxy, DBUS_G_PROXY_ID_TO_CALL (closure->call_id), closure->data);
}

static void
d_pending_call_free (void *data)
{
    GPendingNotifyClosure *closure = data;

    if (closure->free_data_func)
        (* closure->free_data_func) (closure->data);

    g_free (closure);
}

#define DBUS_G_VALUE_ARRAY_COLLECT_ALL(VALARRAY, FIRST_ARG_TYPE, ARGS)
\
G_STMT_START { \
    GType valtype; \
    guint i = 0; \
    \
    VALARRAY = g_value_array_new (6); \
    valtype = FIRST_ARG_TYPE; \
    \
    while (valtype != G_TYPE_INVALID) \
    { \
        gchar *collect_err; \
        GValue *val; \
        \
        g_value_array_append (VALARRAY, NULL); \
        val = g_value_array_get_nth (VALARRAY, i); \
        g_value_init (val, valtype); \
        collect_err = NULL; \
    }

```

```

        G_VALUE_COLLECT (val, ARGS, G_VALUE_NOCOPY_CONTENTS,
&collect_err); \
        \
        if (collect_err) \
        { \
            g_critical ("%s: unable to collect argument %u: %s", \
                G_STRFUNC, i, collect_err); \
            g_free (collect_err); \
            g_value_array_free (VALARRAY); \
            VALARRAY = NULL; \
            break; \
        } \
        \
        valtype = va_arg (ARGS, GType); \
        i++; \
    } \
} G_STMT_END

DBusGProxyCall *
manager_begin_bus_call (DBusGProxyManager *manager,
                        const char *method,
                        DBusGProxyCallNotify notify,
                        gpointer user_data,
                        GDestroyNotify destroy,
                        GType first_arg_type,
                        ...)
{
    guint call_id = 0;
    DBusGProxyPrivate *priv;
    va_list args;
    GValueArray *arg_values;

    va_start (args, first_arg_type);

    if (!manager->bus_proxy)
    {
        manager->bus_proxy = g_object_new (DBUS_TYPE_G_PROXY,
            "name", DBUS_SERVICE_DBUS,
            "path", DBUS_PATH_DBUS,
            "interface", DBUS_INTERFACE_DBUS,
            NULL);
        priv = DBUS_G_PROXY_GET_PRIVATE(manager->bus_proxy);
        priv->manager = manager;
    }

    DBUS_G_VALUE_ARRAY_COLLECT_ALL (arg_values, first_arg_type, args);

    if (arg_values != NULL)
    {
        call_id = dbus_g_proxy_begin_call_internal (manager->bus_proxy,
method,
            notify, user_data, destroy, arg_values, -1);
    }
}

```

```

        g_value_array_free (arg_values);
    }

    va_end (args);

    return DBUS_G_PROXY_ID_TO_CALL (call_id);
}

/**
 * SECTION:dbus-gproxy
 * @short_description: D-Bus Proxy
 * @see_also: #DBusGProxy
 * @stability: Stable
 *
 * A #DBusGProxy is a #GObject representing a remote object in a D-Bus
 * service.
 */

/**
 * DBusGProxy:
 *
 * A #GObject representing a remote object in a D-Bus service.
 */

/**
 * DBusGProxyCall:
 *
 * An opaque pointer representing an asynchronous call in progress.
 */

/*
 * dbus_g_proxy_get_type:
 * Standard GObject get_type() function for DBusGProxy.
 *
 * Returns: type ID for DBusGProxy class
 */
GType
dbus_g_proxy_get_type (void)
{
    static GType object_type = 0;

    if (!object_type)
    {
        static const GTypeInfo object_info =
        {
            sizeof (DBusGProxyClass),
            (GBaseInitFunc) NULL,
            (GBaseFinalizeFunc) NULL,
            (GClassInitFunc) dbus_g_proxy_class_init,
            NULL, /* class_finalize */
            NULL, /* class_data */
        };
    }
}

```



```

        sizeof (DBusGProxy),
        0, /* n_preallocs */
        (GInstanceInitFunc) dbus_g_proxy_init,
    };

    object_type = g_type_register_static (G_TYPE_OBJECT,
                                         "DBusGProxy",
                                         &object_info, 0);
}

return object_type;
}

static DBusGProxy*
dbus_g_proxy_new (DBusGConnection *connection,
                 const char      *name,
                 const char      *path_name,
                 const char      *interface_name)
{
    DBusGProxy *proxy;

    g_assert (connection != NULL);

    proxy = g_object_new (DBUS_TYPE_G_PROXY,
                         "name", name,
                         "path", path_name,
                         "interface", interface_name,
                         "connection", connection, NULL);

    return proxy;
}

/**
 * dbus_g_proxy_new_for_name:
 * @connection: the connection to the remote bus
 * @name: any name on the message bus
 * @path: name of the object instance to call methods on
 * @iface: name of the interface to call methods on
 *
 * Creates a new proxy for a remote interface exported by a connection
 * on a message bus. Method calls and signal connections over this
 * proxy will go to the name owner; the name's owner is expected to
 * support the given interface name. THE NAME OWNER MAY CHANGE OVER
 * TIME, for example between two different method calls, unless the
 * name is a unique name. If you need a fixed owner, you need to
 * request the current owner and bind a proxy to its unique name
 * rather than to the generic name; see
 * dbus_g_proxy_new_for_name_owner().
 *
 * A name-associated proxy only makes sense with a message bus, not
 * for app-to-app direct dbus connections.
 */

```



```

                                const char      *path,
                                const char      *iface,
                                GError          **error)
{
    DBusGProxy *proxy;
    char *unique_name;

    g_return_val_if_fail (connection != NULL, NULL);
    g_return_val_if_fail (g_dbus_is_name (name), NULL);
    g_return_val_if_fail (g_variant_is_object_path (path), NULL);
    g_return_val_if_fail (g_dbus_is_interface_name (iface), NULL);

    if (!(unique_name = get_name_owner
(DBUS_CONNECTION_FROM_G_CONNECTION (connection), name, error)))
        return NULL;

    proxy = dbus_g_proxy_new (connection, unique_name, path, iface);
    g_free (unique_name);
    return proxy;
}

/**
 * dbus_g_proxy_new_from_proxy:
 * @proxy: the proxy to use as a template
 * @iface: name of the interface to call methods on
 * @path: of the object inside the peer to call methods on
 *
 * Creates a proxy using an existing proxy as a template, substituting
 * the specified interface and path.  Either or both may be NULL.
 *
 * Returns: new proxy object
 */
DBusGProxy*
dbus_g_proxy_new_from_proxy (DBusGProxy      *proxy,
                             const char      *iface,
                             const char      *path)
{
    DBusGProxyPrivate *priv;

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), NULL);
    g_return_val_if_fail (path == NULL || g_variant_is_object_path
(path), NULL);
    g_return_val_if_fail (iface == NULL ||
        g_dbus_is_interface_name (iface), NULL);

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    if (iface == NULL)
        iface = priv->interface;
    if (path == NULL)
        path = priv->path;

```

```

    return dbus_g_proxy_new (DBUS_G_CONNECTION_FROM_CONNECTION (priv-
>manager->connection),
                            priv->name,
                            path, iface);
}

/**
 * dbus_g_proxy_new_for_peer:
 * @connection: the connection to the peer
 * @path: name of the object inside the peer to call methods on
 * @iface: name of the interface to call methods on
 *
 * Creates a proxy for an object in peer application (one
 * we're directly connected to). That is, this function is
 * intended for use when there's no message bus involved,
 * we're doing a simple 1-to-1 communication between two
 * applications.
 *
 * Returns: new proxy object
 */
DBusGProxy*
dbus_g_proxy_new_for_peer (DBusGConnection      *connection,
                           const char          *path,
                           const char          *iface)
{
    DBusGProxy *proxy;

    g_return_val_if_fail (connection != NULL, NULL);
    g_return_val_if_fail (g_variant_is_object_path (path), NULL);
    g_return_val_if_fail (g_dbus_is_interface_name (iface), NULL);

    proxy = dbus_g_proxy_new (connection, NULL, path, iface);

    return proxy;
}

/**
 * dbus_g_proxy_get_bus_name:
 * @proxy: the proxy
 *
 * Gets the bus name a proxy is bound to (may be %NULL in some cases).
 * If you created the proxy with dbus_g_proxy_new_for_name(), then
 * the name you passed to that will be returned.
 * If you created it with dbus_g_proxy_new_for_name_owner(), then the
 * unique connection name will be returned. If you created it
 * with dbus_g_proxy_new_for_peer() then %NULL will be returned.
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 *
 * Returns: the bus name the proxy sends messages to
 */

```



```

method);

if (message == NULL)
    return NULL;

dbus_message_iter_init_append (message, &msgiter);
for (i = 0; i < args->n_values; i++)
{
    GValue *gvalue;

    gvalue = g_value_array_get_nth (args, i);

    if (!_dbus_gvalue_marshal (&msgiter, gvalue))
    {
        /* This is a programming error by the caller, most likely */
        gchar *contents = g_strdup_value_contents (gvalue);

        g_critical ("Could not marshal argument %u for %s: type %s,
value %s",
                    i, method, G_VALUE_TYPE_NAME (gvalue), contents);
        g_free (contents);
        dbus_message_unref (message);
        return NULL;
    }
}

return message;
}

static guint
dbus_g_proxy_begin_call_internal (DBusGProxy *proxy,
                                const char *method,
                                DBusGProxyCallNotify notify,
                                gpointer user_data,
                                GDestroyNotify destroy,
                                GValueArray *args,
                                int timeout)
{
    DBusMessage *message;
    DBusPendingCall *pending;
    GPendingNotifyClosure *closure;
    guint call_id;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE (proxy);

    pending = NULL;

    message = dbus_g_proxy_marshal_args_to_message (proxy, method,
args);

    /* can only happen on a programming error or OOM; we already
critical'd */
    if (!message)
        return 0;

```

```

if (!dbus_connection_send_with_reply (priv->manager->connection,
                                     message,
                                     &pending,
                                     timeout))

    oom ();

dbus_message_unref (message);

/* If we got a NULL pending, that means the connection was
disconnected,
 * and we need to abort this call.
 * https://bugs.freedesktop.org/show_bug.cgi?id=12675
 */
if (pending == NULL)
    return 0;

call_id = ++priv->call_id_counter;

if (notify != NULL)
{
    closure = g_new (GPendingNotifyClosure, 1);
    closure->proxy = proxy; /* No need to ref as the lifecycle is
tied to proxy */
    closure->call_id = call_id;
    closure->func = notify;
    closure->data = user_data;
    closure->free_data_func = destroy;
    dbus_pending_call_set_notify (pending, d_pending_call_notify,
                                closure,
                                d_pending_call_free);
}

g_hash_table_insert (priv->pending_calls, GUINT_TO_POINTER
(call_id), pending);

return call_id;
}

static gboolean
dbus_g_proxy_end_call_internal (DBusGProxy      *proxy,
                              guint            call_id,
                              GError          **error,
                              GType           first_arg_type,
                              va_list         args)
{
    DBusMessage *reply;
    DBusMessageIter msgiter;
    DBusError derror;
    va_list args_unwind;
    guint over;
    int n_retvals_processed;

```



```

gboolean ret;
GType valtype;
DBusPendingCall *pending;
DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

if (call_id == 0)
{
    /* Being disconnected is the only reason this can happen, except
    * for programmer error; if it was programmer error, we already
    * emitted a critical warning. */
    g_set_error (error, DBUS_G_ERROR, DBUS_G_ERROR_DISCONNECTED,
                "Disconnected from D-Bus (or argument error during call)");
    return FALSE;
}

reply = NULL;
ret = FALSE;
n_retvals_processed = 0;
over = 0;

/* Keep around a copy of output arguments so we can free on error.
*/
G_VA_COPY(args_unwind, args);

pending = g_hash_table_lookup (priv->pending_calls, GUINT_TO_POINTER
(call_id));

dbus_pending_call_block (pending);
reply = dbus_pending_call_steal_reply (pending);

g_assert (reply != NULL);

dbus_error_init (&derror);

switch (dbus_message_get_type (reply))
{
    case DBUS_MESSAGE_TYPE_METHOD_RETURN:
        dbus_message_iter_init (reply, &msgiter);
        valtype = first_arg_type;
        while (valtype != G_TYPE_INVALID)
        {
            int arg_type;
            gpointer return_storage;
            GValue gvalue = { 0, };
            DBusGValueMarshalCtx context;

            context.recursion_depth = 0;
            context.gconnection = DBUS_G_CONNECTION_FROM_CONNECTION (priv-
>manager->connection);
            context.proxy = proxy;

            arg_type = dbus_message_iter_get_arg_type (&msgiter);

```

```

if (arg_type == DBUS_TYPE_INVALID)
{
    g_set_error (error, DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Too few arguments in reply");
    goto out;
}

return_storage = va_arg (args, gpointer);
if (return_storage == NULL)
    goto next;

/* We handle variants specially; the caller is expected
 * to have already allocated storage for them.
 */
if (arg_type == DBUS_TYPE_VARIANT
    && g_type_is_a (valtype, G_TYPE_VALUE))
{
    if (!_dbus_gvalue_demarshal_variant (&context, &msgiter,
(GValue*) return_storage, NULL))
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Couldn't convert argument, expected \"%s\"",
                    g_type_name (valtype));

        goto out;
    }
}
else
{
    g_value_init (&gvalue, valtype);

    if (!_dbus_gvalue_demarshal (&context, &msgiter, &gvalue,
error))
        goto out;

    /* Anything that can be demarshaled must be storable */
    if (!_dbus_gvalue_store (&gvalue, return_storage))
        g_assert_not_reached ();
    /* Ownership of the value passes to the client, don't unset
*/
}

next:
    n_retvals_processed++;
    dbus_message_iter_next (&msgiter);
    valtype = va_arg (args, GType);
}

while (dbus_message_iter_get_arg_type (&msgiter) !=
DBUS_TYPE_INVALID)

```

```

    {
        over++;
        dbus_message_iter_next (&msgiter);
    }

    if (over > 0)
    {
        g_set_error (error, DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Too many arguments in reply; expected %d, got %d",
                    n_retvals_processed, over);
        goto out;
    }
    break;
case DBUS_MESSAGE_TYPE_ERROR:
    dbus_set_error_from_message (&derror, reply);
    dbus_set_g_error (error, &derror);
    dbus_error_free (&derror);
    goto out;
break;
default:
    dbus_set_error (&derror, DBUS_ERROR_FAILED,
                    "Reply was neither a method return nor an
exception");
    dbus_set_g_error (error, &derror);
    dbus_error_free (&derror);
    goto out;
break;
}

ret = TRUE;
out:
if (ret == FALSE)
{
    int i;

    valtype = first_arg_type;
    for (i = 0; i < n_retvals_processed; i++)
    {
        GValue value = {0,};
        gpointer retval;

        g_value_init (&value, valtype);

        retval = va_arg (args_unwind, gpointer);
        if (retval == NULL)
        {
            i--;
            continue;
        }

        _dbus_gvalue_take (&value, retval);
    }
}

```



```

        GDestroyNotify    destroy,
        GType              first_arg_type,
        ...)
{
    guint call_id = 0;
    va_list args;
    GValueArray *arg_values;
    DBusGProxyPrivate *priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), NULL);
    g_return_val_if_fail (!DBUS_G_PROXY_DESTROYED (proxy), NULL);
    g_return_val_if_fail (g_dbus_is_member_name (method), NULL);

    va_start (args, first_arg_type);

    DBUS_G_VALUE_ARRAY_COLLECT_ALL (arg_values, first_arg_type, args);

    if (arg_values != NULL)
    {
        call_id = dbus_g_proxy_begin_call_internal (proxy, method,
notify,
        user_data, destroy, arg_values, priv->default_timeout);

        g_value_array_free (arg_values);
    }

    va_end (args);

    return DBUS_G_PROXY_ID_TO_CALL (call_id);
}

/**
 * dbus_g_proxy_begin_call_with_timeout:
 * @proxy: a proxy for a remote interface
 * @method: the name of the method to invoke
 * @notify: callback to be invoked when method returns
 * @user_data: user data passed to callback
 * @destroy: function called to destroy user_data
 * @timeout: the timeout in milliseconds, or -1 to use a default
 * @first_arg_type: type of the first argument, or %G_TYPE_INVALID if
there
 *   are no arguments
 * @...: first argument, followed by any further type/value pairs,
followed
 *   by %G_TYPE_INVALID
 *
 * Asynchronously invokes a method on a remote interface. The method
 * call will not be sent over the wire until the application returns
 * to the main loop, or blocks in dbus_g_connection_flush() to write
out
 * pending data. The call will be completed after a timeout, or when
 * a reply is received. When the call returns, the callback specified

```

```

* will be invoked; you can then collect the results of the call
* (which may be an error, or a reply), use dbus_g_proxy_end_call().
*
* It is an error to call this method on a proxy that has emitted
* the #DBusGProxy::destroy signal.
*
* TODO this particular function shouldn't die on out of memory,
* since you should be able to do a call with large arguments.
*
* Returns: call identifier.
*/
DBusGProxyCall *
dbus_g_proxy_begin_call_with_timeout (DBusGProxy          *proxy,
                                     const char          *method,
                                     DBusGProxyCallNotify notify,
                                     gpointer             user_data,
                                     GDestroyNotify       destroy,
                                     int timeout,
                                     GType                first_arg_type,
                                     ...)
{
    guint call_id = 0;
    va_list args;
    GValueArray *arg_values;

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), NULL);
    g_return_val_if_fail (!DBUS_G_PROXY_DESTROYED (proxy), NULL);
    g_return_val_if_fail (g_dbus_is_member_name (method), NULL);
    g_return_val_if_fail (timeout >= 0 || timeout == -1, NULL);

    va_start (args, first_arg_type);

    DBUS_G_VALUE_ARRAY_COLLECT_ALL (arg_values, first_arg_type, args);

    if (arg_values != NULL)
    {
        call_id = dbus_g_proxy_begin_call_internal (proxy, method,
notify,
        user_data, destroy, arg_values, timeout);

        g_value_array_free (arg_values);
    }

    va_end (args);

    return DBUS_G_PROXY_ID_TO_CALL (call_id);
}

/**
 * dbus_g_proxy_end_call:
 * @proxy: a proxy for a remote interface
 * @call: the pending call ID from dbus_g_proxy_begin_call()

```

```

* @error: return location for an error
* @first_arg_type: type of first "out" argument, or %G_TYPE_INVALID
if
*     there are no "out" arguments
* @...: return location for first "out" argument, followed by any
further
*     type/location pairs, followed by %G_TYPE_INVALID
*
* Collects the results of a method call. The method call was normally
* initiated with dbus_g_proxy_end_call(). You may use this function
* outside of the callback given to dbus_g_proxy_begin_call; in that
* case this function will block if the results haven't yet been
* received.
*
* All D-Bus method calls can fail with a remote error. If this
occurs,
* the @error will be set and this function will return %FALSE.
*
* Otherwise, the "out" parameters and return value of the
* method are stored in the provided varargs list.
* The list should be terminated with G_TYPE_INVALID.
*
* Returns: %TRUE on success
*/
gboolean
dbus_g_proxy_end_call (DBusGProxy          *proxy,
                      DBusGProxyCall      *call,
                      GError              **error,
                      GType               first_arg_type,
                      ...)
{
    gboolean ret;
    va_list args;

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), FALSE);

    va_start (args, first_arg_type);

    ret = dbus_g_proxy_end_call_internal (proxy, GPOINTER_TO_UINT
    (call), error, first_arg_type, args);

    va_end (args);

    return ret;
}

/**
* dbus_g_proxy_call:
* @proxy: a proxy for a remote interface
* @method: method to invoke
* @error: return location for an error

```

```

* @first_arg_type: type of first "in" argument, or %G_TYPE_INVALID if
none
* @...: value of first "in" argument, any further type/value pairs,
*   %G_TYPE_INVALID, type/location pairs for "out" arguments,
*   and %G_TYPE_INVALID again
*
* Function for synchronously invoking a method and receiving reply
* values. This function is equivalent to dbus_g_proxy_begin_call
* followed by dbus_g_proxy_end_call. All of the input arguments are
* specified first, followed by G_TYPE_INVALID, followed by all of the
* output values, followed by a second G_TYPE_INVALID. Note that
* this means you must always specify G_TYPE_INVALID twice.
*
* It is an error to call this method on a proxy that has emitted
* the #DBusGProxy::destroy signal.
*
* Returns: %TRUE if the method succeeds, %FALSE if it fails
*/
gboolean
dbus_g_proxy_call (DBusGProxy      *proxy,
                  const char      *method,
                  GError          **error,
                  GType           first_arg_type,
                  ...)
{
    gboolean ret;
    guint call_id = 0;
    va_list args;
    GValueArray *in_args;
    DBusGProxyPrivate *priv;

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), FALSE);
    g_return_val_if_fail (!DBUS_G_PROXY_DESTROYED (proxy), FALSE);

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    va_start (args, first_arg_type);

    DBUS_G_VALUE_ARRAY_COLLECT_ALL (in_args, first_arg_type, args);

    if (in_args != NULL)
    {
        call_id = dbus_g_proxy_begin_call_internal (proxy, method, NULL,
NULL,
        NULL, in_args, priv->default_timeout);

        g_value_array_free (in_args);
    }

    first_arg_type = va_arg (args, GType);
    ret = dbus_g_proxy_end_call_internal (proxy, call_id, error,
first_arg_type,

```



```

        args);

    va_end (args);

    return ret;
}

/**
 * dbus_g_proxy_call_with_timeout:
 * @proxy: a proxy for a remote interface
 * @method: method to invoke
 * @timeout: the timeout in milliseconds, or -1 to use a default
 * @error: return location for an error
 * @first_arg_type: type of first "in" argument
 * @...: as for dbus_g_proxy_call()
 *
 * Function for synchronously invoking a method and receiving reply
 * values. This function is equivalent to dbus_g_proxy_begin_call
 * followed by dbus_g_proxy_end_call. All of the input arguments are
 * specified first, followed by G_TYPE_INVALID, followed by all of the
 * output values, followed by a second G_TYPE_INVALID. Note that
 * this means you must always specify G_TYPE_INVALID twice.
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 *
 * Returns: %TRUE if the method succeeds, %FALSE if it fails
 */
gboolean
dbus_g_proxy_call_with_timeout (DBusGProxy      *proxy,
                               const char      *method,
                               int timeout,
                               GError          **error,
                               GType           first_arg_type,
                               ...)
{
    gboolean ret;
    guint call_id = 0;
    va_list args;
    GValueArray *in_args;

    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), FALSE);
    g_return_val_if_fail (!DBUS_G_PROXY_DESTROYED (proxy), FALSE);
    g_return_val_if_fail (g_dbus_is_member_name (method), FALSE);
    g_return_val_if_fail (timeout >= 0 || timeout == -1, FALSE);

    va_start (args, first_arg_type);

    DBUS_G_VALUE_ARRAY_COLLECT_ALL (in_args, first_arg_type, args);

    if (in_args != NULL)
    {

```

```

        call_id = dbus_g_proxy_begin_call_internal (proxy, method, NULL,
NULL,
        NULL, in_args, timeout);

        g_value_array_free (in_args);
    }

    first_arg_type = va_arg (args, GType);
    ret = dbus_g_proxy_end_call_internal (proxy, call_id, error,
        first_arg_type, args);

    va_end (args);

    return ret;
}

/**
 * dbus_g_proxy_call_no_reply:
 * @proxy: a proxy for a remote interface
 * @method: the name of the method to invoke
 * @first_arg_type: type of the first argument, or %G_TYPE_INVALID to
call
 *     the method without arguments
 * @...: the first argument and any remaining type/argument pairs,
followed by
 *     %G_TYPE_INVALID to terminate the list
 *
 * Sends a method call message as with dbus_g_proxy_begin_call(), but
 * does not ask for a reply or allow you to receive one.
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 *
 * TODO: this particular function shouldn't die on out of memory,
 * since you should be able to do a call with large arguments.
 */
void
dbus_g_proxy_call_no_reply (DBusGProxy          *proxy,
                           const char          *method,
                           GType              first_arg_type,
                           ...)
{
    DBusMessage *message = NULL;
    va_list args;
    GValueArray *in_args;
    DBusGProxyPrivate *priv;

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (g_dbus_is_member_name (method));
    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

```

```

va_start (args, first_arg_type);
DBUS_G_VALUE_ARRAY_COLLECT_ALL (in_args, first_arg_type, args);

if (in_args != NULL)
{
    message = dbus_g_proxy_marshal_args_to_message (proxy, method,
in_args);

    g_value_array_free (in_args);
}

va_end (args);

/* can only happen on a programming error or OOM; we already
critical'd */
if (!message)
    return;

dbus_message_set_no_reply (message, TRUE);

if (!dbus_connection_send (priv->manager->connection,
                           message,
                           NULL))

    oom ();

dbus_message_unref (message);
}

/**
 * dbus_g_proxy_cancel_call
 * @proxy: a proxy for a remote interface
 * @call: the pending call ID from dbus_g_proxy_begin_call()
 *
 * Cancels a pending method call. The method call was normally
 * initiated with dbus_g_proxy_begin_call(). This function
 * may not be used on pending calls that have already been
 * ended with dbus_g_proxy_end_call.
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 */
void
dbus_g_proxy_cancel_call (DBusGProxy      *proxy,
                          DBusGProxyCall *call)
{
    guint call_id;
    DBusPendingCall *pending;
    DBusGProxyPrivate *priv;

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));

```

```

priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

call_id = DBUS_G_PROXY_CALL_TO_ID (call);

if (call_id == 0)
{
    /* nothing to cancel */
    return;
}

pending = g_hash_table_lookup (priv->pending_calls, GUINT_TO_POINTER
(call_id));
g_hash_table_remove (priv->pending_calls, GUINT_TO_POINTER
(call_id));
g_return_if_fail (pending != NULL);

dbus_pending_call_cancel (pending);
}

/**
 * dbus_g_proxy_send:
 * @proxy: a proxy for a remote interface
 * @message: the message to address and send
 * @client_serial: return location for message's serial, or %NULL
 *
 * Sends a message to the interface we're proxying for. Does not
 * block or wait for a reply. The message is only actually written out
 * when you return to the main loop or block in
 * dbus_g_connection_flush().
 *
 * The message is modified to be addressed to the target interface.
 * That is, a destination name field or whatever is needed will be
 * added to the message. The basic point of this function is to add
 * the necessary header fields, otherwise it's equivalent to
 * dbus_connection_send().
 *
 * This function adds a reference to the message, so the caller
 * still owns its original reference.
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 */
void
dbus_g_proxy_send (DBusGProxy          *proxy,
                  DBusMessage          *message,
                  dbus_uint32_t        *client_serial)
{
    DBusGProxyPrivate *priv;

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));

```

```

priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

if (priv->name)
{
    if (!dbus_message_set_destination (message, priv->name))
        g_error ("Out of memory");
}
if (priv->path)
{
    if (!dbus_message_set_path (message, priv->path))
        g_error ("Out of memory");
}
if (priv->interface)
{
    if (!dbus_message_set_interface (message, priv->interface))
        g_error ("Out of memory");
}

if (!dbus_connection_send (priv->manager->connection, message,
client_serial))
    g_error ("Out of memory\n");
}

static void
array_free_all (gpointer array)
{
    g_array_free (array, TRUE);
}

/**
 * dbus_g_proxy_add_signal:
 * @proxy: the proxy for a remote interface
 * @signal_name: the name of the signal
 * @first_type: the first argument type, or %G_TYPE_INVALID if none
 * @...: any subsequent argument types, followed by %G_TYPE_INVALID
 *
 * Specifies the argument signature of a D-Bus signal. When the signal
is
 * emitted by the remote object, if the GTypes corresponding to its
arguments'
 * types do not match the types given here, the signal will be
ignored.
 *
 * It is an error to add the same @signal_name to the same @proxy more
than
 * once, even if the argument types given are the same.
 *
 * It is also an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 */
void

```

```

dbus_g_proxy_add_signal (DBusGProxy      *proxy,
                        const char      *signal_name,
                        GType           first_type,
                        ...)
{
    GQuark q;
    char *name;
    GArray *gtypesig;
    GType gtype;
    va_list args;
    DBusGProxyPrivate *priv;

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));
    g_return_if_fail (g_dbus_is_member_name (signal_name));

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);

    name = create_signal_name (priv->interface, signal_name);

    q = g_quark_from_string (name);

    g_return_if_fail (g_datalist_id_get_data (&priv->signal_signatures,
q) == NULL);

    gtypesig = g_array_new (FALSE, TRUE, sizeof (GType));

    va_start (args, first_type);
    gtype = first_type;
    while (gtype != G_TYPE_INVALID)
    {
        g_array_append_val (gtypesig, gtype);
        gtype = va_arg (args, GType);
    }
    va_end (args);

#ifdef G_DISABLE_CHECKS
    if (_dbus_gobject_lookup_marshall (G_TYPE_NONE, gtypesig->len,
(const GType*) gtypesig->data) == NULL)
        g_warning ("No marshaller for signature of signal '%s'",
signal_name);
#endif

    g_datalist_id_set_data_full (&priv->signal_signatures,
                                q, gtypesig,
                                array_free_all);

    g_free (name);
}

/**

```



```

        {
            g_warning ("Attempt to disconnect from signal '%s' which is not
registered\n",
                    name);
        }

    g_free (name);
}

/**
 * dbus_g_proxy_set_default_timeout:
 * @proxy: a proxy for a remote interface
 * @timeout: the timeout in milliseconds, or -1 to reset to the
libdbus default
 *
 * Sets the default timeout to use for a proxy. This timeout will be
 * used in calls where the timeout is not specified, or is specified
to be -1.
 * If this timeout is also set to -1, libdbus will use a reasonable
default
 * value.
 *
 * This is useful for long-running operations that takes longer than
 * the default timeout (which is a on the order of magnitude of tens
 * of seconds). For some applications, consider using a pattern where
 * the method returns once the operation is underway
 * (e.g. immediately) and emits a signal when the operation terminates
 * (though beware of leaking information with/in the signal return
value).
 *
 * It is an error to call this method on a proxy that has emitted
 * the #DBusGProxy::destroy signal.
 *
 * Since: 0.75
 */
void
dbus_g_proxy_set_default_timeout (DBusGProxy      *proxy,
                                int               timeout)
{
    DBusGProxyPrivate *priv;

    g_return_if_fail (DBUS_IS_G_PROXY (proxy));
    g_return_if_fail (!DBUS_G_PROXY_DESTROYED (proxy));
    g_return_if_fail (timeout >= 0 || timeout == -1);

    priv = DBUS_G_PROXY_GET_PRIVATE(proxy);
    priv->default_timeout = timeout;
}

```

File = dbus-gproxy.sgml

```
<!-- ##### SECTION Title ##### -->
DBusGProxy

<!-- ##### SECTION Short_Description ##### -->

<!-- ##### SECTION Long_Description ##### -->
<para>

</para>

<!-- ##### SECTION See_Also ##### -->
<para>

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### STRUCT DBusGProxy ##### -->
<para>

</para>

<!-- ##### STRUCT DBusGProxyCall ##### -->
<para>

</para>

<!-- ##### USER_FUNCTION DBusGProxyCallNotify ##### -->
<para>

</para>

@proxy:
@call_id:
@user_data:

<!-- ##### FUNCTION dbus_g_proxy_new_for_name ##### -->
<para>

</para>
```

@connection:
@name:
@path:
@iface:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_new_for_name_owner ##### -->
<para>

</para>

@connection:
@name:
@path:
@iface:
@error:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_new_from_proxy ##### -->
<para>

</para>

@proxy:
@iface:
@path:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_new_for_peer ##### -->
<para>

</para>

@connection:
@path:
@iface:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_set_interface ##### -->
<para>

</para>

@proxy:
@interface_name:

<!-- ##### FUNCTION dbus_g_proxy_get_path ##### -->

<para>

</para>

@proxy:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_get_bus_name ##### -->

<para>

</para>

@proxy:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_get_interface ##### -->

<para>

</para>

@proxy:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_add_signal ##### -->

<para>

</para>

@proxy:
@signal_name:
@first_type:
@...:

<!-- ##### FUNCTION dbus_g_proxy_connect_signal ##### -->

<para>

</para>

@proxy:
@signal_name:
@handler:
@data:
@free_data_func:

<!-- ##### FUNCTION dbus_g_proxy_disconnect_signal ##### -->

<para>

</para>

@proxy:
@signal_name:
@handler:
@data:

<!-- ##### FUNCTION dbus_g_proxy_send ##### -->

<para>

</para>

@proxy:
@message:
@client_serial:

<!-- ##### FUNCTION dbus_g_proxy_call ##### -->

<para>

</para>

@proxy:
@method:
@error:
@first_arg_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_call_with_timeout ##### -->

<para>

</para>

@proxy:
@method:
@timeout:
@error:
@first_arg_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_call_no_reply ##### -->

<para>

</para>

@proxy:
@method:

@first_arg_type:
@...:

<!-- ##### FUNCTION dbus_g_proxy_begin_call ##### -->
<para>

</para>

@proxy:
@method:
@notify:
@user_data:
@destroy:
@first_arg_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_begin_call_with_timeout ##### -->
<para>

</para>

@proxy:
@method:
@notify:
@user_data:
@destroy:
@timeout:
@first_arg_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_end_call ##### -->
<para>

</para>

@proxy:
@call:
@error:
@first_arg_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_proxy_cancel_call ##### -->
<para>

</para>

```
@proxy:
@call:
```

```
<!-- ##### FUNCTION dbus_g_proxy_set_default_timeout ##### -->
<para>

</para>
```

```
@proxy:
@timeout:
```

```
File = dbus-gsignature.c
```

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gsignature.c Mapping from dbus type signatures to GType
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */
```

```
#include "config.h"
#include "dbus-gtest.h"
#include "dbus-gsignature.h"
#include "dbus-gvalue-utils.h"
#include <string.h>
#include <glib.h>
```

```

#define MAP_BASIC(d_t, g_t) \
    case DBUS_TYPE_##d_t: \
        return G_TYPE_##g_t;
static GType
typecode_to_gtype (int type)
{
    switch (type)
    {
        MAP_BASIC (BOOLEAN, BOOLEAN);
        MAP_BASIC (BYTE, UCHAR);
        MAP_BASIC (INT16, INT);
        MAP_BASIC (INT32, INT);
        MAP_BASIC (UINT16, UINT);
        MAP_BASIC (UINT32, UINT);
        MAP_BASIC (INT64, INT64);
        MAP_BASIC (UINT64, UINT64);
        MAP_BASIC (DOUBLE, DOUBLE);
        MAP_BASIC (STRING, STRING);
    default:
        return G_TYPE_INVALID;
    }
}
#undef MAP_BASIC

static gboolean
dbus_typecode_maps_to_basic (int typecode)
{
    return typecode_to_gtype (typecode) != G_TYPE_INVALID;
}

GType
_dbus_gtype_from_basic_typecode (int typecode)
{
    g_assert (dbus_type_is_basic (typecode));
    g_assert (dbus_typecode_maps_to_basic (typecode));
    return typecode_to_gtype (typecode);
}

static GType
signature_iter_to_g_type_dict (const DBusSignatureIter *subiter,
gboolean is_client)
{
    DBusSignatureIter iter;
    GType key_gtype;
    GType value_gtype;

    g_assert (dbus_signature_iter_get_current_type (subiter) ==
DBUS_TYPE_DICT_ENTRY);

    dbus_signature_iter_recurse (subiter, &iter);

    key_gtype = _dbus_gtype_from_signature_iter (&iter, is_client);

```



```

if (key_gtype == G_TYPE_INVALID)
    return G_TYPE_INVALID;

dbus_signature_iter_next (&iter);
value_gtype = _dbus_gtype_from_signature_iter (&iter, is_client);
if (value_gtype == G_TYPE_INVALID)
    return G_TYPE_INVALID;

if (!_dbus_gtype_is_valid_hash_key (key_gtype)
    || !_dbus_gtype_is_valid_hash_value (value_gtype))
    /* Later we need to return DBUS_TYPE_G_VALUE */
    return G_TYPE_INVALID;

return dbus_g_type_get_map ("GHashTable", key_gtype, value_gtype);
}

static GType
signature_iter_to_g_type_array (DBusSignatureIter *iter, gboolean
is_client)
{
    GType elt_gtype;

    elt_gtype = _dbus_gtype_from_signature_iter (iter, is_client);
    if (elt_gtype == G_TYPE_INVALID)
        return G_TYPE_INVALID;

    if (elt_gtype == G_TYPE_OBJECT)
        return DBUS_TYPE_G_OBJECT_ARRAY;
    if (elt_gtype == G_TYPE_STRING)
        return G_TYPE_STRV;
    if (_dbus_g_type_is_fixed (elt_gtype))
        return dbus_g_type_get_collection ("GArray", elt_gtype);
    else if (g_type_is_a (elt_gtype, G_TYPE_OBJECT)
        || g_type_is_a (elt_gtype, G_TYPE_BOXED))
        return dbus_g_type_get_collection ("GPtrArray", elt_gtype);

    /* Later we need to return DBUS_TYPE_G_VALUE */
    return G_TYPE_INVALID;
}

static GType
signature_iter_to_g_type_struct (DBusSignatureIter *iter, gboolean
is_client)
{
    GArray *types;
    GType ret;
    types = g_array_new (FALSE, FALSE, sizeof (GType));
    do
    {
        GType curtype;
        curtype = _dbus_gtype_from_signature_iter (iter, is_client);
        g_array_append_val (types, curtype);
    }

```

```

    }
    while (dbus_signature_iter_next (iter));

    ret = dbus_g_type_get_structv ("GValueArray", types->len, (GType*)
types->data);
    g_array_free (types, TRUE);
    return ret;
}

GType
_dbus_gtype_from_signature_iter (DBusSignatureIter *iter, gboolean
is_client)
{
    int current_type;

    current_type = dbus_signature_iter_get_current_type (iter);
    /* TODO: handle type 0? */
    if (dbus_typecode_maps_to_basic (current_type))
        return _dbus_gtype_from_basic_typecode (current_type);
    else if (current_type == DBUS_TYPE_OBJECT_PATH)
        return DBUS_TYPE_G_OBJECT_PATH;
    else if (current_type == DBUS_TYPE_SIGNATURE)
        return DBUS_TYPE_G_SIGNATURE;
    else
    {
        DBusSignatureIter subiter;

        g_assert (dbus_type_is_container (current_type));

        if (current_type == DBUS_TYPE_VARIANT)
            return G_TYPE_VALUE;

        dbus_signature_iter_recurse (iter, &subiter);

        if (current_type == DBUS_TYPE_ARRAY)
        {
            int elt_type = dbus_signature_iter_get_current_type (&subiter);
            if (elt_type == DBUS_TYPE_DICT_ENTRY)
                return signature_iter_to_g_type_dict (&subiter, is_client);
            else
                return signature_iter_to_g_type_array (&subiter, is_client);
        }
        else if (current_type == DBUS_TYPE_STRUCT)
        {
            return signature_iter_to_g_type_struct (&subiter,
is_client);
        }
        else
        {
            g_assert_not_reached ();
            return G_TYPE_INVALID;
        }
    }
}

```



```

GType          _dbus_gtype_from_signature_iter (DBusSignatureIter
*sigiter,
                                gboolean          is_client);

GArray *       _dbus_gtypes_from_arg_signature (const char
*signature,
                                gboolean          is_client);

#endif

```

File = dbus-gtest-main.c

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gtest-main.c  Program to run all libdbus-glib tests
 *
 * Copyright (C) 2003  Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */
#include <config.h>

#ifdef DBUS_BUILD_TESTS

#include "dbus-gtest.h"
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int
main (int   argc,
      char **argv)

```

```

{
  const char *test_data_dir;

  setlocale(LC_ALL, "");

  if (argc > 1)
    test_data_dir = argv[1];
  else
    test_data_dir = NULL;

  dbus_glib_internal_do_not_use_run_tests (test_data_dir);

  return 0;
}

#endif

```

File = dbus-gtest.c

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-test.c  Program to run all tests
 *
 * Copyright (C) 2002, 2003  Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */

```

```

#include <config.h>
#include "dbus-gtest.h"
#include <stdio.h>
#include <stdlib.h>

```

```

#include <glib.h>

#ifdef DBUS_BUILD_TESTS
static void
die (const char *failure)
{
    fprintf (stderr, "Unit test failed: %s\n", failure);
    exit (1);
}
#endif /* DBUS_BUILD_TESTS */

/*
 * An exported symbol to be run in order to execute
 * unit tests. Should not be used by
 * any app other than our test app, this symbol
 * won't exist in some builds of the library.
 * (with --enable-tests=no)
 *
 * @param test_data_dir the directory with test data (test/data
normally)
 */
void
dbus_glib_internal_do_not_use_run_tests (const char *test_data_dir)
{
#ifdef DBUS_BUILD_TESTS
    if (test_data_dir == NULL)
        test_data_dir = g_getenv ("DBUS_TEST_DATA");

    if (test_data_dir != NULL)
        printf ("Test data in %s\n", test_data_dir);
    else
        printf ("No test data!\n");

    g_type_init ();

    printf ("%s: running GValue util tests\n", "dbus-glib-test");
    if (!_dbus_gvalue_utils_test (test_data_dir))
        die ("gvalue utils");

    printf ("%s: running GValue tests\n", "dbus-glib-test");
    if (!_dbus_gvalue_test (test_data_dir))
        die ("gvalue utils");

    printf ("%s: running glib tests\n", "dbus-glib-test");
    if (!_dbus_glib_test (test_data_dir))
        die ("glib");

    printf ("%s: running mainloop integration tests\n", "dbus-glib-
test");
    if (!_dbus_gmain_test (test_data_dir))
        die ("gmain");

```

```

printf ("%s: running GObject tests\n", "dbus-glib-test");
if (!_dbus_gobject_test (test_data_dir))
    die ("gobject");

printf ("%s: completed successfully\n", "dbus-glib-test");
#else
printf ("Not compiled with unit tests, not running any\n");
#endif
}

```

File = dbus-gtest.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gtest.h  Declarations of test functions.
 *
 * Copyright (C) 2003  Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */

#ifndef DBUS_GLIB_TEST_H
#define DBUS_GLIB_TEST_H

#include <dbus/dbus-glib.h>

gboolean _dbus_gmain_test    (const char *test_data_dir);
gboolean _dbus_gobject_test  (const char *test_data_dir);
gboolean _dbus_gutils_test   (const char *test_data_dir);
gboolean _dbus_glib_test     (const char *test_data_dir);
gboolean _dbus_gvalue_test   (const char *test_data_dir);

```

```

gboolean _dbus_gvalue_utils_test    (const char *test_data_dir);

void dbus_glib_internal_do_not_use_run_tests (const char
*test_data_dir);

#endif /* DBUS_GLIB_TEST_H */

File = dbus-gthread.c

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gthread.c  GThread integration
 *
 * Copyright (C) 2002  CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */

#include <config.h>

/* #define G_DEBUG_LOCKS 1 */

#include <glib.h>
#include <dbus/dbus.h>
#include "dbus-glib.h"

/**
 * dbus_g_thread_init:
 *
 * Initializes the D-BUS thread system.
 * This function may only be called
 * once and must be called prior to calling any
 * other function in the D-BUS API.

```



```

*/
void
dbus_g_thread_init (void)
{
    if (!g_thread_supported ())
        g_error ("g_thread_init() must be called before
dbus_threads_init()");

    dbus_threads_init_default ();
}

```

File = dbus-gtype-specialized-priv.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gtype-specialized-priv.h: internals of specialized GTypes
*
* Copyright (C) 2009 Collabora Ltd.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/

```

```

#ifndef DBUS_GOBJECT_TYPE_SPECIALIZED_PRIV_H
#define DBUS_GOBJECT_TYPE_SPECIALIZED_PRIV_H

```

```

#include "dbus-gtype-specialized.h"

```

```

G_BEGIN_DECLS

```

```

G_GNUC_INTERNAL

```

```

void          _dbus_g_type_register_collection      (const char
*name,

```

```

                                const
DBusGTypeSpecializedCollectionVtable  *vtable,
                                quint
flags);

G_GNUC_INTERNAL
void      _dbus_g_type_register_map      (const char
*name,

                                const
DBusGTypeSpecializedMapVtable  *vtable,
                                quint
flags);

G_GNUC_INTERNAL
void      _dbus_g_type_register_struct  (const char
*name,

                                const
DBusGTypeSpecializedStructVtable  *vtable,
                                quint
flags);

G_END_DECLS

#endif

```

File = dbus-gtype-specialized.c

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gtype-specialized.c: Non-DBus-specific functions for
specialized GTypes
*
* Copyright (C) 2005 Red Hat, Inc.
* Copyright (C) 2007 Codethink Ltd.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software

```

```

* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>

#include "dbus-glib.h"
#include "dbus-gtype-specialized-priv.h"
#include "dbus-gvalue-utils.h"
#include <glib.h>
#include <string.h>
#include <gobject/gvaluecollector.h>

/**
 * SECTION:dbus-gtype-specialized
 * @short_description: Specialized GTypes
 * @stability: Unstable
 *
 * Specialized gtypes are basically a way to allow the definition of
 * recursive #GTypes. It allows the definition of 'containers' which
 is
 * basically a user defined structure capable of holding other data,
 and a
 * set of functions defining how to access that structure. Containers
 come in
 * 3 flavors: collections, maps and structs.
 *
 * A collection is a container that holds an ordered set of items, all
 * of which must be the same type. (This is an
<emphasis>array</emphasis>
 * in standard D-Bus terminology.) dbus-glib specialized collections
 can be
 * #GArray (for numeric elements), #GPtrArray (for string, object or
 * boxed elements), #GSLList (for boxed elements, not recommended), or
 a
 * user-defined type.
 *
 * A map is a container that holds a set of key/value pairs.
 * The keys have one type, and the values another; the type of the
 keys
 * must be a numeric or string-like type. (This is a
<emphasis>dict</emphasis>
 * (dictionary) or <emphasis>array of dict entry</emphasis> in
 standard D-Bus
 * terminology.) dbus-glib specialized maps can be #GHashTable or a
 * user-defined type.
 *
 * A struct is a container that holds a fixed number of members, each
 member
 * having a predefined type. (This is a <emphasis>struct</emphasis> in

```

```

* standard D-Bus terminology.) It is analogous to the C @struct
keyword, but
* dbus-glib does not generally represent D-Bus structs in C structs.
* dbus-glib specialized structs can be #GValueArray or a user-defined
type.
*
* A specialization is a GType detailing a particular container with
* particular types (a type specialization).
*
* Functions are provided for constructing and manipulating
specializations.
*
* This documentation needs splitting into two pages, one for defining
new
* containers and using existing containers. I expect most users to
only do
* the latter. I also need to add some examples.
*/

/**
* DBUS_TYPE_G_BOOLEAN_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#gboolean
* (corresponding to the D-Bus signature "ab").
*/

/**
* DBUS_TYPE_G_INT_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#gint
* (corresponding to the D-Bus signature "ai").
*/

/**
* DBUS_TYPE_G_UINT_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#guint
* (corresponding to the D-Bus signature "au").
*/

/**
* DBUS_TYPE_G_INT64_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#gint64
* (corresponding to the D-Bus signature "ax").
*/

/**

```

```

* DBUS_TYPE_G_UINT64_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#guint64
* (corresponding to the D-Bus signature "at").
*/

/**
* DBUS_TYPE_G_UCHAR_ARRAY:
*
* Expands to a function call returning the #GType of a #GArray of
#guchar
* (corresponding to the D-Bus signature "ay").
*
* Note that this is not the same thing as a #GByteArray! dbus-glib
does not
* know about the #GByteArray type.
*/

/**
* DBUS_TYPE_G_OBJECT_ARRAY:
*
* Expands to a function call returning the #GType of a #GPtrArray of
#GObject.
*
* Use this type with caution: it can sometimes be used as a
representation
* of arrays whose D-Bus signature is "ao" (transferred as an array of
object
* paths), but the conventional type for such arrays is
* <literal>(dbus_g_type_get_collection ("GPtrArray",
* DBUS_TYPE_G_OBJECT_PATH)</literal>.
*/

/**
* DBUS_TYPE_G_STRING_STRING_HASHTABLE:
*
* Expands to a function call returning the #GType of a #GHashTable
where the
* keys are strings and the values are also strings (corresponding to
the
* D-Bus signature "a{ss}").
*/

/**
* DBusGTypeSpecializedVtable:
* @constructor: returns a new, blank instance of the @type
* @free_func: if not %NULL, frees the @type instance @val
* @copy_func: returns a "deep copy" of the @type instance @val
* @simple_free_func: if not %NULL, frees its argument
*

```

```

* A table of methods used to implement specialized container
behaviour on
* user-defined collections, maps and structs. Exactly one of
@free_func and
* @simple_free_func must be implemented; the other must be %NULL.
* @constructor and @copy_func must always be implemented.
*
* There are additional members, which are reserved for future
expansion and
* must be %NULL.
*/

/**
* DBusGTypeSpecializedConstructor:
* @type: a specialized boxed type
*
* <!-- -->
*
* Returns: a new instance of @type
*/

/**
* DBusGTypeSpecializedFreeFunc:
* @type: a specialized boxed type
* @val: an instance of @type
*
* Frees @val according to @type. This is analogous to
#GBoxedFreeFunc, but
* can use information from @type (for instance to free the contents
of a
* container before freeing the actual container).
*/

/**
* DBusGTypeSpecializedCopyFunc:
* @type: a specialized boxed type
* @src: an instance of @type
*
* Copies @src according to @type. This is analogous to
#GBoxedCopyFunc, but
* can use information from @type (for instance to copy each element
of a
* collection).
*
* Returns: a deep copy of @src
*/

/**
* DBusGTypeSpecializedCollectionFixedAccessorFunc:
* @type: a specialized collection boxed type
* @instance: an instance of @type
* @values: used to return a pointer to the contents of @instance

```

```

* @len: used to return the number of elements in @instance
*
* Implements dbus_g_type_collection_get_fixed() for a #GValue with
type
* @type, containing @instance.
*
* Returns: %TRUE on success
*/

/**
* DBusGTypeSpecializedCollectionIterator:
* @value: an element of the collection
* @user_data: the data supplied when calling
*   dbus_g_type_collection_value_iterate()
*
* A library-user-supplied function, called for each element in the
* collection when dbus_g_type_collection_value_iterate() is called.
*/

/**
* DBusGTypeSpecializedCollectionIteratorFunc:
* @type: a specialized collection boxed type
* @instance: an instance of @type
* @iterator: the function to call for each element
* @user_data: data to pass to @iterator
*
* Implements dbus_g_type_collection_value_iterate() for a #GValue
with
* type @type, containing @instance.
*/

/**
* DBusGTypeSpecializedCollectionAppendFunc:
* @ctx: an appending context returned by
dbus_g_type_specialized_init_append()
* @val: a value to copy into the collection
*
* Implements dbus_g_type_specialized_collection_append().
*
* This function should use the @val and @specialization_type members
of @ctx.
*/

/**
* DBusGTypeSpecializedCollectionEndAppendFunc:
* @ctx: an appending context returned by
dbus_g_type_specialized_init_append()
*
* Implements dbus_g_type_specialized_collection_end_append().
*
* This function should use the @val and @specialization_type members
of @ctx.

```

```

*/

/**
 * DBusGTypeSpecializedCollectionVtable:
 * @base_vtable: base methods shared between collections and other
types
 * @fixed_accessor: if not %NULL, provides access to the contents of
this
 *   collection, as documented for dbus_g_type_collection_get_fixed()
 * @iterator: iterates through the members of @instance
 * @append_func: appends a new member to @instance
 * @end_append_func: if not %NULL, called after each group of calls to
 *   the @append_func
 *
 * A table of methods used to implement specialized collection
behaviour
 * on user-defined types. At least @iterator and @append_func must be
 * implemented.
 */

/**
 * DBusGTypeSpecializedMapIterator:
 * @key_val: a key from the map
 * @value_val: a value from the map
 * @user_data: the data supplied when calling
 *   dbus_g_type_map_value_iterate()
 *
 * A library-user-supplied function, called for each key/value pair in
the
 * collection when dbus_g_type_map_value_iterate() is called.
 */

/**
 * DBusGTypeSpecializedMapIteratorFunc:
 * @type: a specialized map boxed type
 * @instance: an instance of @type
 * @iterator: the function to call for each key/value pair
 * @user_data: data to pass to @iterator
 *
 * Implements dbus_g_type_map_value_iterate() for a #GValue with
 * type @type, containing @instance.
 */

/**
 * DBusGTypeSpecializedMapAppendFunc:
 * @ctx: an appending context returned by
dbus_g_type_specialized_init_append()
 * @key: a key to add to the collection
 * @val: a value to add to the collection
 *
 * Implements dbus_g_type_specialized_map_append().
 */

```



```

* This function should use the @val and @specialization_type members
of @ctx,
* and replace any existing value with key equal to @key.
*/

/**
* DbusGTypeSpecializedMapVtable:
* @base_vtable: base methods shared between maps and other types
* @iterator: iterates through the members of @instance
* @append_func: adds a new key/value pair to @instance
*
* A table of methods used to implement specialized collection
behaviour
* on user-defined types. Both methods must be implemented.
*/

/**
* DbusGTypeSpecializedStructGetMember:
* @type: a specialized struct boxed type
* @instance: an instance of @type
* @member: the index of the member, starting from 0
* @ret_value: an initialized #GValue of the appropriate type for the
given
* member of @type
*
* Implements dbus_g_type_struct_get_member() for a #GValue with type
@type,
* containing @instance.
*
* Returns: %TRUE on success
*/

/**
* DbusGTypeSpecializedStructSetMember:
* @type: a specialized struct boxed type
* @instance: an instance of @type
* @member: the index of the member, starting from 0
* @new_value: an initialized #GValue of the appropriate type for the
given
* member of @type
*
* Implements dbus_g_type_struct_set_member() for a #GValue with type
@type,
* containing @instance.
*
* Returns: %TRUE on success
*/

/**
* DbusGTypeSpecializedStructVtable:
* @base_vtable: base methods shared between maps and other types
* @get_member: returns a member by its index

```

```

* @set_member: sets a member by its index
*
* A table of methods used to implement specialized collection
behaviour
* on user-defined types. Both methods must be implemented.
*/

typedef enum {
    DBUS_G_SPECTYPE_COLLECTION,
    DBUS_G_SPECTYPE_MAP,
    DBUS_G_SPECTYPE_STRUCT
} DBusGTypeSpecializedType;

typedef struct {
    DBusGTypeSpecializedType type;
    const DBusGTypeSpecializedVtable *vtable;
} DBusGTypeSpecializedContainer;

typedef struct {
    guint num_types;
    GType *types;
    const DBusGTypeSpecializedContainer *klass;
} DBusGTypeSpecializedData;

static GHashTable /* char * -> data* */ *specialized_containers;

static GQuark
specialized_type_data_quark ()
{
    static GQuark quark;
    if (!quark)
        quark = g_quark_from_static_string ("DBusGTypeSpecializedData");

    return quark;
}

static gpointer
specialized_init (gpointer arg G_GNUC_UNUSED)
{
    g_assert (specialized_containers == NULL);

    specialized_containers = g_hash_table_new_full (g_str_hash,
g_str_equal,
    g_free, NULL);

    _dbus_g_type_specialized_builtins_init ();
    return NULL;
}

/**
* dbus_g_type_specialized_init:
*

```

```

* Initialize dbus-glib specialized #GType<!-- -->s.
*
* In older versions of dbus-glib, it was necessary to do this before
* instantiating or registering any specialized type. It is now done
* automatically whenever necessary.
*/
void
dbus_g_type_specialized_init (void)
{
    static GOnce once = G_ONCE_INIT;

    g_once (&once, specialized_init, NULL);
}

static DBusGTypeSpecializedData *
lookup_specialization_data (GType type)
{
    return g_type_get_qdata (type, specialized_type_data_quark ());
}

/* Copied from gboxed.c */
static void
proxy_value_init (GValue *value)
{
    value->data[0].v_pointer = NULL;
}

/* Adapted from gboxed.c */
static void
proxy_value_free (GValue *value)
{
    if (value->data[0].v_pointer && !(value->data[1].v_uint &
G_VALUE_NOCOPY_CONTENTS))
    {
        DBusGTypeSpecializedData *data;
        GType type;

        type = G_VALUE_TYPE (value);
        data = lookup_specialization_data (type);
        g_assert (data != NULL);

        if (data->klass->vtable->free_func)
        {
            data->klass->vtable->free_func (type, value->data[0].v_pointer);
        }
        else
        {
            g_assert (data->klass->vtable->simple_free_func != NULL);
            data->klass->vtable->simple_free_func (value->data[0].v_pointer);
        }
    }
}

```

```

    }
}

/* Adapted from gboxed.c */
static void
proxy_value_copy (const GValue *src_value,
                  GValue        *dest_value)
{
    if (src_value->data[0].v_pointer)
    {
        DBusGTypeSpecializedData *data;
        GType type;
        type = G_VALUE_TYPE (src_value);
        data = lookup_specialization_data (type);
        g_assert (data != NULL);
        dest_value->data[0].v_pointer = data->klass->vtable->copy_func
(type, src_value->data[0].v_pointer);
    }
    else
        dest_value->data[0].v_pointer = src_value->data[0].v_pointer;
}

/* Copied from gboxed.c */
static gpointer
proxy_value_peek_pointer (const GValue *value)
{
    return value->data[0].v_pointer;
}

/* Adapted from gboxed.c */
static gchar*
proxy_collect_value (GValue        *value,
                    guint          n_collect_values,
                    GTypeCValue *collect_values,
                    guint          collect_flags)
{
    DBusGTypeSpecializedData *data;
    GType type;

    type = G_VALUE_TYPE (value);
    data = lookup_specialization_data (type);

    if (!collect_values[0].v_pointer)
        value->data[0].v_pointer = NULL;
    else
    {
        if (collect_flags & G_VALUE_NOCOPY_CONTENTS)
        {
            value->data[0].v_pointer = collect_values[0].v_pointer;
            value->data[1].v_uint = G_VALUE_NOCOPY_CONTENTS;
        }
    }
}

```

```

        else
        {
            value->data[0].v_pointer = data->klass->vtable->copy_func
(type, collect_values[0].v_pointer);
        }
    }

    return NULL;
}

/* Adapted from gboxed.c */
static gchar*
proxy_lcopy_value (const GValue *value,
                   guint          n_collect_values,
                   GTypeCValue   *collect_values,
                   guint          collect_flags)
{
    gpointer *boxed_p = collect_values[0].v_pointer;

    if (!boxed_p)
        return g_strdup_printf ("value location for `%s' passed as NULL",
G_VALUE_TYPE_NAME (value));

    if (!value->data[0].v_pointer)
        *boxed_p = NULL;
    else if (collect_flags & G_VALUE_NOCOPY_CONTENTS)
        *boxed_p = value->data[0].v_pointer;
    else
    {
        DBusGTypeSpecializedData *data;
        GType type;

        type = G_VALUE_TYPE (value);
        data = lookup_specialization_data (type);

        *boxed_p = data->klass->vtable->copy_func (type, value-
>data[0].v_pointer);
    }

    return NULL;
}

static char *
build_specialization_name (const char *prefix, guint num_types, const
GType *types)
{
    GString *fullname;
    guint i;

    fullname = g_string_new (prefix);

    g_string_append_c (fullname, '_');

```

```

for (i=0; i < num_types; i++)
{
    if (i!=0)
        g_string_append_c (fullname, '+');
    g_string_append (fullname, g_type_name (types[i]));
}
g_string_append_c (fullname, '_');
return g_string_free (fullname, FALSE);
}

static void
register_container (const char *name,
                   DBusGTypeSpecializedType type,
                   const DBusGTypeSpecializedVtable *vtable)
{
    DBusGTypeSpecializedContainer *klass;

    g_warn_if_fail (vtable->constructor != NULL);
    /* must have either free_func or simple_free_func */
    g_warn_if_fail (vtable->free_func != NULL ||
                   vtable->simple_free_func != NULL);
    g_warn_if_fail (vtable->copy_func != NULL);

    klass = g_new0 (DBusGTypeSpecializedContainer, 1);
    klass->type = type;
    klass->vtable = vtable;

    g_hash_table_insert (specialized_containers, g_strdup (name),
klass);
}

/**
 * dbus_g_type_register_collection:
 * @name: The name of a new collection container
 * @vtable: the vtable defining the new container
 * @flags: As yet unused.
 *
 * Defines a new collection container.
 */
void
dbus_g_type_register_collection (const char
*name,
                                const DBusGTypeSpecializedCollectionVtable
*vtable,
                                guint
flags)
{
    dbus_g_type_specialized_init();

    _dbus_g_type_register_collection (name, vtable, flags);
}

```

```

void
_dbus_g_type_register_collection (const char
*name,
                                const DBusGTypeSpecializedCollectionVtable
*vtable,
                                guint
flags)
{
    /* fixed_accessor is optional */
    g_warn_if_fail (vtable->iterator != NULL);
    g_warn_if_fail (vtable->append_func != NULL);
    /* end_append_func is optional */

    register_container (name, DBUS_G_SPECTYPE_COLLECTION, (const
DBusGTypeSpecializedVtable*) vtable);
}

/**
 * dbus_g_type_register_map:
 * @name: The name of a new map container
 * @vtable: the vtable defining the new container
 * @flags: As yet unused.
 *
 * Defines a new map container.
 */
void
dbus_g_type_register_map (const char                                *name,
                          const DBusGTypeSpecializedMapVtable    *vtable,
                          guint                                    flags)
{
    dbus_g_type_specialized_init();

    _dbus_g_type_register_map (name, vtable, flags);
}

void
_dbus_g_type_register_map (const char
*name,
                          const DBusGTypeSpecializedMapVtable    *vtable,
                          guint                                    flags)
{
    g_warn_if_fail (vtable->iterator != NULL);
    g_warn_if_fail (vtable->append_func != NULL);

    register_container (name, DBUS_G_SPECTYPE_MAP, (const
DBusGTypeSpecializedVtable*) vtable);
}

/**
 * dbus_g_type_register_struct:
 * @name: The name of a new struct container
 * @vtable: the vtable defining the new container

```

```

* @flags: As yet unused.
*
* Defines a new struct container.
*/
void
dbus_g_type_register_struct (const char
*name,
                           const DBusGTypeSpecializedStructVtable *vtable,
                           guint flags)
{
    dbus_g_type_specialized_init();

    _dbus_g_type_register_struct (name, vtable, flags);
}

void
_dbus_g_type_register_struct (const char
*name,
                              const DBusGTypeSpecializedStructVtable *vtable,
                              guint flags)
{
    g_warn_if_fail (vtable->get_member != NULL);
    g_warn_if_fail (vtable->set_member != NULL);

    register_container (name, DBUS_G_SPECTYPE_STRUCT, (const
DBusGTypeSpecializedVtable*) vtable);
}

/**
 * dbus_g_type_map_peek_vtable:
 * @map_type: a gtype of a map specialization
 *
 * Peek the vtable for a given map specialization
 *
 * Returns: the vtable
 */
const DBusGTypeSpecializedMapVtable* dbus_g_type_map_peek_vtable
(GType map_type)
{
    DBusGTypeSpecializedData *data;
    g_return_val_if_fail (dbus_g_type_is_map(map_type), NULL);

    data = lookup_specialization_data (map_type);
    g_assert (data != NULL);

    return (DBusGTypeSpecializedMapVtable *) (data->klass->vtable);
}

/**
 * dbus_g_type_collection_peek_vtable:
 * @collection_type: a gtype of a collection specialization
 *

```



```

* Peek the vtable for a given collection specialization
*
* Returns: the vtable
*/
const DBusGTypeSpecializedCollectionVtable*
dbus_g_type_collection_peek_vtable (GType collection_type)
{
    DBusGTypeSpecializedData *data;
    g_return_val_if_fail (dbus_g_type_is_collection(collection_type),
        NULL);

    data = lookup_specialization_data (collection_type);
    g_assert (data != NULL);

    return (DBusGTypeSpecializedCollectionVtable *) (data->klass-
        >vtable);
}

/**
* dbus_g_type_struct_peek_vtable:
* @struct_type: a gtype of a struct specialization
*
* Peek the vtable for a given struct specialization
*
* Returns: the vtable
*/
const DBusGTypeSpecializedStructVtable* dbus_g_type_struct_peek_vtable
(GType struct_type)
{
    DBusGTypeSpecializedData *data;
    g_return_val_if_fail (dbus_g_type_is_struct (struct_type), NULL);

    data = lookup_specialization_data (struct_type);
    g_assert (data != NULL);

    return (DBusGTypeSpecializedStructVtable *) (data->klass->vtable);
}

static GType
register_specialized_instance (const DBusGTypeSpecializedContainer
*klass,
                               const char *name,
                               guint
num_types,
                               const GType *types)
{
    GType ret;

    static const GTypeValueTable vtable =
    {
        proxy_value_init,
        proxy_value_free,

```

```

    proxy_value_copy,
    proxy_value_peek_pointer,
    "p",
    proxy_collect_value,
    "p",
    proxy_lcopy_value,
};
static const GTypeInfo derived_info =
{
    0,          /* class_size */
    NULL,      /* base_init */
    NULL,      /* base_finalize */
    NULL,      /* class_init */
    NULL,      /* class_finalize */
    NULL,      /* class_data */
    0,         /* instance_size */
    0,         /* n_preallocs */
    NULL,      /* instance_init */
    &vtable,   /* value_table */
};

ret = g_type_register_static (G_TYPE_BOXED, name, &derived_info, 0);
/* install proxy functions upon successful registration */
if (ret != G_TYPE_INVALID)
{
    DBusGTypeSpecializedData *data;
    data = g_new0 (DBusGTypeSpecializedData, 1);
    data->num_types = num_types;
    data->types = g_memdup (types, sizeof (GType) * num_types);
    data->klass = klass;
    g_type_set_qdata (ret, specialized_type_data_quark (), data);
}

return ret;
}

static GType
lookup_or_register_specialized (const char *container,
                               guint       num_types,
                               const GType *types)
{
    GType ret;
    char *name;
    const DBusGTypeSpecializedContainer *klass;

    dbus_g_type_specialized_init();

    klass = g_hash_table_lookup (specialized_containers, container);
    g_return_val_if_fail (klass != NULL, G_TYPE_INVALID);

    name = build_specialization_name (container, num_types, types);
    ret = g_type_from_name (name);
}

```

```

if (ret == G_TYPE_INVALID)
{
    /* Take ownership of name */
    ret = register_specialized_instance (klass, name,
                                        num_types,
                                        types);
}
g_free (name);
return ret;
}

/**
 * dbus_g_type_get_collection:
 * @container: a string specifying a registered collection type
 * @specialization: #GType of collection elements
 *
 * Gets a #GType for a particular collection instance,
 * creating the type if not already created.
 *
 * Returns: the #GType of that instance
 */
GType
dbus_g_type_get_collection (const char *container,
                           GType specialization)
{
    return lookup_or_register_specialized (container, 1,
&specialization);
}

/**
 * dbus_g_type_get_map:
 * @container: a string specifying a registered map type
 * @key_specialization: #GType of keys
 * @value_specialization: #GType of values
 *
 * Gets a #GType for a particular map instance,
 * creating the type if not already created.
 *
 * Returns: the #GType of that instance
 */
GType
dbus_g_type_get_map (const char *container,
                    GType key_specialization,
                    GType value_specialization)
{
    GType types[2];
    types[0] = key_specialization;
    types[1] = value_specialization;
    return lookup_or_register_specialized (container, 2, types);
}

```

```

/**
 * dbus_g_type_get_structv:
 * @container: a string specifying a registered struct type
 * @num_members: number of members in the struct
 * @types: an array specifying a GType for each struct element
 *
 * Gets a #GType for a particular struct instance,
 * creating the type if not already created.
 *
 * Returns: the #GType of that instance
 */
GType
dbus_g_type_get_structv (const char *container,
                        guint num_members,
                        GType *types)
{
    return lookup_or_register_specialized (container, num_members,
types);
}

/**
 * dbus_g_type_get_struct:
 * @container: a string specifying a registered struct type
 * @first_type: #GType for the struct's first member
 * @...: more GTypes for the struct's members, terminated by
G_TYPE_INVALID
 *
 * Varargs method to get a #GType for a particular struct instance,
 * creating the type if not already created.
 *
 * Returns: the #GType of that instance
 */
GType
dbus_g_type_get_struct (const char *container,
                        GType first_type,
                        ...)
{
    GArray *types;
    GType curtype, ret;
    va_list args;
    va_start (args, first_type);

    types = g_array_new (FALSE, FALSE, sizeof (GType));
    curtype = first_type;
    while (curtype != G_TYPE_INVALID)
    {
        g_array_append_val (types, curtype);
        curtype = va_arg (args, GType);
    }
    va_end (args);

    ret = lookup_or_register_specialized (container, types->len,

```

```

        (GType *) types->data);

g_array_free (types, TRUE);

return ret;
}

/**
 * dbus_g_type_is_collection:
 * @gtype: a GType to test
 *
 * Tests if a given GType is a collection.
 *
 * Returns: true if the given GType is a collection
 */
gboolean
dbus_g_type_is_collection (GType gtype)
{
    DBusGTypeSpecializedData *data;
    data = lookup_specialization_data (gtype);
    if (data == NULL)
        return FALSE;
    return data->klass->type == DBUS_G_SPECTYPE_COLLECTION;
}

/**
 * dbus_g_type_is_map:
 * @gtype: a GType to test
 *
 * Tests if a given GType is a map,
 * i.e. it was created with dbus_g_type_get_map().
 *
 * Returns: true if the given GType is a map
 */
gboolean
dbus_g_type_is_map (GType gtype)
{
    DBusGTypeSpecializedData *data;
    data = lookup_specialization_data (gtype);
    if (data == NULL)
        return FALSE;
    return data->klass->type == DBUS_G_SPECTYPE_MAP;
}

/**
 * dbus_g_type_is_struct:
 * @gtype: a GType to test
 *
 * Tests if a given GType is a struct,
 * i.e. it was created with dbus_g_type_get_struct()
 */

```

```

* Returns: true if the given GType is a struct
*/
gboolean
dbus_g_type_is_struct (GType gtype)
{
    DBusGTypeSpecializedData *data;
    data = lookup_specialization_data (gtype);
    if (data == NULL)
        return FALSE;
    return data->klass->type == DBUS_G_SPECTYPE_STRUCT;
}

static GType
get_specialization_index (GType gtype, guint i)
{
    DBusGTypeSpecializedData *data;

    data = lookup_specialization_data (gtype);
    if (i < data->num_types)
        return data->types[i];
    else
        return G_TYPE_INVALID;
}

/**
 * dbus_g_type_get_collection_specialization:
 * @gtype: a collection #GType, as created by
dbus_g_type_get_collection()
 *
 * Return the type of each element in collections of type @gtype.
 * It is an error to call this function on a non-collection type.
 *
 * Returns: the element type for a given collection GType.
 */
GType
dbus_g_type_get_collection_specialization (GType gtype)
{
    g_return_val_if_fail (dbus_g_type_is_collection (gtype),
G_TYPE_INVALID);
    return get_specialization_index (gtype, 0);
}

/**
 * dbus_g_type_get_map_key_specialization:
 * @gtype: a map #GType, as created by dbus_g_type_get_map()
 *
 * Return the type of the keys in maps of type @gtype.
 * It is an error to call this function on a non-map type.
 *
 * Returns: the key type for a given map #GType.
 */

```

```

GType
dbus_g_type_get_map_key_specialization (GType gtype)
{
    g_return_val_if_fail (dbus_g_type_is_map (gtype), G_TYPE_INVALID);
    return get_specialization_index (gtype, 0);
}

/**
 * dbus_g_type_get_map_value_specialization:
 * @gtype: a map GType, as created by dbus_g_type_get_map().
 *
 * Return the type of the values in maps of type @gtype.
 * It is an error to call this function on a non-map type.
 *
 * Returns: the value type for a given map GType.
 */
GType
dbus_g_type_get_map_value_specialization (GType gtype)
{
    g_return_val_if_fail (dbus_g_type_is_map (gtype), G_TYPE_INVALID);
    return get_specialization_index (gtype, 1);
}

/**
 * dbus_g_type_get_struct_member_type
 * @gtype: a struct GType, as created with dbus_g_type_get_struct()
 * @member: the index of a struct member
 *
 * Get the type of a member of a specialized struct.
 * It is an error to call this function on a non-struct type.
 *
 * Returns: the type for a given member of a struct #GType,
 *         or %G_TYPE_INVALID if @member >=
dbus_g_type_get_struct_size()
 */
GType
dbus_g_type_get_struct_member_type (GType gtype, guint member)
{
    g_return_val_if_fail (dbus_g_type_is_struct (gtype),
G_TYPE_INVALID);
    return get_specialization_index (gtype, member);
}

/**
 * dbus_g_type_get_struct_size
 * @gtype: a struct GType, as created with dbus_g_type_get_struct().
 *
 * Get the number of members in a specialized struct.
 * It is an error to call this function on a non-struct type.
 *
 * Returns: the number of members in a given struct #GType.
 */

```

```

guint
dbus_g_type_get_struct_size (GType gtype)
{
    DBusGTypeSpecializedData *data;
    g_return_val_if_fail (dbus_g_type_is_struct (gtype),
G_TYPE_INVALID);

    data = lookup_specialization_data (gtype);
    return data->num_types;
}

/**
 * dbus_g_type_specialized_construct:
 * @gtype: a specialized #GType, as created by
dbus_g_type_get_collection(),
 * dbus_g_type_get_map() or dbus_g_type_get_struct()
 *
 * Create an instance of a given specialized type.
 * The structure created and returned will depend on the container
type of the
 * GType. E.g. If the given type was created by
 * dbus_g_type_get_collection("GArray", G_TYPE_INT),
 * then this will return a GArray with element_size of sizeof(int)
 *
 * Returns: a pointer to a newly constructed instance of the given
type.
 */
gpointer
dbus_g_type_specialized_construct (GType gtype)
{
    DBusGTypeSpecializedData *data;

    dbus_g_type_specialized_init();

    data = lookup_specialization_data (gtype);
    g_return_val_if_fail (data != NULL, FALSE);

    return data->klass->vtable->constructor (gtype);
}

/**
 * dbus_g_type_collection_get_fixed:
 * @value: a GValue containing a boxed specialized collection
 * that has a @fixed_accessor in its vtable
 * @data_ret: used to return a pointer to the fixed data, which must
not be
 * modified (for instance, for a #GArray of #gint, this would point
 * to an array of #gint)
 * @len_ret: used to return the length (counting collection elements,
not
 * bytes: in a #GArray containing one #gint, this would be 1)
 *
 */

```



```

* Calling this function is likely to be a bad idea. Consider using
* dbus_g_type_collection_value_iterate() instead.
*
* On success, @data_ret is a pointer to the underlying data in a
collection
* of fixed-length fundamental types. Knowledge of the underlying data
model
* of the collection is needed in order to use @data_ret correctly.
*
* It is an error to call this function on a specialized type that is
not a
* collection, or on a collection that does not have a @fixed_accessor
in its
* #DBusGTypeSpecializedCollectionVtable.
*
* Specialized #GArray<!---->s are the only types provided by dbus-
glib that
* can be used with this function; user-defined types might also work.
*
* Returns: %TRUE on success
*/
gboolean
dbus_g_type_collection_get_fixed (GValue *value,
                                gpointer *data_ret,
                                guint *len_ret)
{
    DBusGTypeSpecializedData *data;
    DBusGTypeSpecializedCollectionVtable *vtable;
    GType gtype;

    dbus_g_type_specialized_init();

    g_return_val_if_fail (G_VALUE_HOLDS_BOXED (value), FALSE);

    gtype = G_VALUE_TYPE (value);
    g_return_val_if_fail (dbus_g_type_is_collection (gtype), FALSE);

    data = lookup_specialization_data (gtype);
    /* dbus_g_type_is_collection() already checked this */
    g_assert (data != NULL);

    vtable = (DBusGTypeSpecializedCollectionVtable *) (data->klass-
>vtable);
    g_return_val_if_fail (vtable->fixed_accessor != NULL, FALSE);

    return vtable->fixed_accessor (gtype, g_value_get_boxed (value),
                                data_ret, len_ret);
}

/**
* dbus_g_type_collection_value_iterate:
* @value: a #GValue holding a collection type.

```

```

* @iterator: a function to call for each element
* @user_data: user data to pass to the @iterator
*
* Calls the given function for each element of the collection.
* The function is passed a #GValue containing the element and the
given
* @user_data parameter. The collection may not be modified while
iterating over
* it.
*/
void
dbus_g_type_collection_value_iterate (const GValue
*value,
                                     DBusGTypeSpecializedCollectionIterator
iterator,
                                     gpointer
user_data)
{
    DBusGTypeSpecializedData *data;
    GType gtype;

    dbus_g_type_specialized_init();

    g_return_if_fail (G_VALUE_HOLDS_BOXED (value));

    gtype = G_VALUE_TYPE (value);
    g_return_if_fail (dbus_g_type_is_collection (gtype));

    data = lookup_specialization_data (gtype);
    /* dbus_g_type_is_collection() already checked this */
    g_assert (data != NULL);

    ((DBusGTypeSpecializedCollectionVtable *) data->klass->vtable)-
>iterator (gtype,
                                     g_value_get_boxed
(value),
                                     iterator,
user_data);
}

/**
 * DBusGTypeSpecializedAppendContext:
 * @val: the #GValue containing the array to which you're appending
 * @specialization_type: the #GType of the array's elements
 *
 * A context for appending. There are more fields, which are private.
 */

typedef struct {
    GValue *val;
    GType specialization_type;
    DBusGTypeSpecializedData *specdata;

```

```

    guint c;
    gpointer d;
} DBusGTypeSpecializedAppendContextReal;

G_STATIC_ASSERT (sizeof (DBusGTypeSpecializedAppendContextReal) ==
                 sizeof (DBusGTypeSpecializedAppendContext));

/**
 * dbus_g_type_specialized_init_append:
 * @value: a #GValue containing an instance of specialized type
 * @ctx: a #DBusGTypeSpecializedAppendContext in which to return a new
 *       appending context.
 *
 * Create a new context for adding elements to a collection or
 * key/value pairs
 * to a map. You generally don't need or want to use this..
 */
void
dbus_g_type_specialized_init_append (GValue *value,
DBusGTypeSpecializedAppendContext *ctx)
{
    DBusGTypeSpecializedAppendContextReal *realctx =
(DBusGTypeSpecializedAppendContextReal *) ctx;
    GType gtype;
    DBusGTypeSpecializedData *specdata;

    dbus_g_type_specialized_init();

    g_return_if_fail (G_VALUE_HOLDS_BOXED (value));
    gtype = G_VALUE_TYPE (value);
    specdata = lookup_specialization_data (gtype);
    g_return_if_fail (specdata != NULL);
    g_return_if_fail (specdata->num_types != 0);

    realctx->val = value;
    realctx->specialization_type = specdata->types[0];
    realctx->specdata = specdata;
}

/**
 * dbus_g_type_specialized_collection_append:
 * @ctx: a context created by dbus_g_type_specialized_init_append()
 *       for a #GValue containing a collection
 * @elt: a #GValue containing an element to append to the collection
 *
 * Appends a given element to the end of a collection.
 */
void
dbus_g_type_specialized_collection_append
(DBusGTypeSpecializedAppendContext *ctx,
                                     GValue
                                     *elt)
{

```

```

    DBusGTypeSpecializedAppendContextReal *realctx =
(DBusGTypeSpecializedAppendContextReal *) ctx;

    g_return_if_fail (dbus_g_type_is_collection (G_VALUE_TYPE (ctx-
>val)));

    ((DBusGTypeSpecializedCollectionVtable *) realctx->specdata->klass-
>vtable)->append_func (ctx, elt);
}

/**
 * dbus_g_type_specialized_collection_end_append:
 * @ctx: a context created by dbus_g_type_specialized_init_append()
 *       for a #GValue containing a collection
 *
 * Finish appending elements to a given collection
 */
void
dbus_g_type_specialized_collection_end_append
(DBusGTypeSpecializedAppendContext *ctx)
{
    DBusGTypeSpecializedAppendContextReal *realctx =
(DBusGTypeSpecializedAppendContextReal *) ctx;

    g_return_if_fail (dbus_g_type_is_collection (G_VALUE_TYPE (ctx-
>val)));

    if (((DBusGTypeSpecializedCollectionVtable *) realctx->specdata-
>klass->vtable)->end_append_func != NULL)
        ((DBusGTypeSpecializedCollectionVtable *) realctx->specdata-
>klass->vtable)->end_append_func (ctx);
}

/**
 * dbus_g_type_specialized_map_append:
 * @ctx: a context created by dbus_g_type_specialized_init_append()
 *       for a #GValue containing a map
 * @key: a GValue containing a key, whose contents will be stolen by
@ctx
 * @val: a GValue containing a value, whose contents will be stolen by
@ctx
 *
 * Inserts the given key/value pair into the map instance.
 */
void
dbus_g_type_specialized_map_append (DBusGTypeSpecializedAppendContext
*ctx,
                                   GValue                                *key,
                                   GValue                                *val)
{
    DBusGTypeSpecializedAppendContextReal *realctx =
(DBusGTypeSpecializedAppendContextReal *) ctx;

```

```

    g_return_if_fail (dbus_g_type_is_map (G_VALUE_TYPE (ctx->val)));

    ((DBusGTypeSpecializedMapVtable *) realctx->specdata->klass->vtable)->append_func (ctx, key, val);
}

/**
 * dbus_g_type_map_value_iterate:
 * @value: a #GValue holding a specialized map
 * @iterator: a function to call for each element
 * @user_data: user data to pass to the @iterator
 *
 * Calls the given function for each key/value pair of the map.
 * The function is passed two GValues containing the key/value pair
and the given
 * @user_data parameter. The map may not be modified while iterating
over
 * it.
 */
void
dbus_g_type_map_value_iterate (const GValue
*value,
                                DBusGTypeSpecializedMapIterator
iterator,
                                gpointer
user_data)
{
    DBusGTypeSpecializedData *data;
    GType gtype;

    dbus_g_type_specialized_init();

    g_return_if_fail (G_VALUE_HOLDS_BOXED (value));

    gtype = G_VALUE_TYPE (value);
    g_return_if_fail (dbus_g_type_is_map (gtype));

    data = lookup_specialization_data (gtype);
    /* already checked by dbus_g_type_is_map() */
    g_assert (data != NULL);

    ((DBusGTypeSpecializedMapVtable *) data->klass->vtable)->iterator
(gtype,
                                g_value_get_boxed
(value),
                                iterator, user_data);
}

/**
 * dbus_g_type_struct_get_member:

```

```

* @value: a #GValue containing a struct instance
* @member: the index of a given member
* @dest: an initialised #GValue in which to return the struct member
*
* Fetches a given member of a given struct instance. @dest must be
initialised
* was the correct type for that member, e.g. as returned by
* @dbus_g_type_get_struct_member_type
*
* Returns: %TRUE if successful
*/
gboolean
dbus_g_type_struct_get_member (const GValue *value,
                               guint        member,
                               GValue      *dest)
{
    DBusGTypeSpecializedData *data;
    GType gtype;

    dbus_g_type_specialized_init();

    g_return_val_if_fail (G_VALUE_HOLDS_BOXED (value), FALSE);

    gtype = G_VALUE_TYPE (value);
    g_return_val_if_fail (dbus_g_type_is_struct (gtype), FALSE);

    data = lookup_specialization_data (gtype);
    /* already checked by dbus_g_type_is_struct() */
    g_assert (data != NULL);

    return ((DBusGTypeSpecializedStructVtable *) (data->klass->vtable))-
>get_member(gtype,
g_value_get_boxed (value),
member,
dest);
}

/**
* dbus_g_type_struct_set_member:
* @value: a #GValue containing a struct instance
* @member: the index of a given member
* @src: an #GValue containing the new value for that struct member
*
* Sets a given member of a struct to a new value. The type of @src
must match
* the existing type of @member member of the struct.
*
* Returns: %TRUE if successful
*/
gboolean
dbus_g_type_struct_set_member (GValue      *value,

```

```

        guint      member,
        const GValue *src)
{
    DBusGTypeSpecializedData *data;
    GType gtype;

    dbus_g_type_specialized_init();

    g_return_val_if_fail (G_VALUE HOLDS_BOXED (value), FALSE);

    gtype = G_VALUE_TYPE (value);
    g_return_val_if_fail (dbus_g_type_is_struct (gtype), FALSE);

    data = lookup_specialization_data (gtype);
    /* already checked by dbus_g_type_is_struct() */
    g_assert (data != NULL);

    return ((DBusGTypeSpecializedStructVtable *) (data->klass->vtable))-
>set_member(gtype,

g_value_get_boxed (value),

member,
src);
}

/**
 * dbus_g_type_struct_get:
 * @value: a #GValue containing a struct instance
 * @member: struct member to get
 * @...: location in which to return the value of this member,
 * followed optionally by more member/return locations pairs,
followed by
 * by %G_MAXUINT
 *
 * Collects the selected values of this struct into the return
locations
 * provided.
 *
 * Returns: %FALSE on failure
 */

gboolean
dbus_g_type_struct_get (const GValue *value,
                        guint first_member,
                        ...)
{
    va_list var_args;
    GType type;
    guint size, i;
    gchar *error;
    GValue val = {0,};

```

```
g_return_val_if_fail (dbus_g_type_is_struct (G_VALUE_TYPE (value)),
FALSE);
```

```
va_start (var_args, first_member);
size = dbus_g_type_get_struct_size (G_VALUE_TYPE (value));
i = first_member;
while (i != G_MAXUINT)
{
    if (i >= size)
        goto error;
```

```
    type = dbus_g_type_get_struct_member_type (G_VALUE_TYPE
(value), i);
```

```
    g_value_init (&val, type);
    dbus_g_type_struct_get_member (value, i, &val);
```

```
    G_VALUE_LCOPY (&val, var_args, 0, &error);
    if (error)
```

```
    {
        g_warning ("%s, %s", G_STRFUNC, error);
        g_free (error);
        g_value_unset (&val);
        goto error;
```

```
    }
    g_value_unset (&val);
    i = va_arg (var_args, guint);
```

```
    }
va_end (var_args);
return TRUE;
```

```
error:
va_end (var_args);
return FALSE;
```

```
}
```

```
/**
```

```
* dbus_g_type_struct_set:
* @value: a #GValue containing a struct instance
* @member: struct member to set
* @...: value for the first member, followed optionally by
*       more member/value pairs, followed by %G_MAXUINT
*
* Sets the selected members of the struct in @value.
*
* Returns: %FALSE on failure
*/
```

```
gboolean
dbus_g_type_struct_set (GValue *value,
                        guint first_member,
                        ...)
```

```
{
```



```

va_list var_args;
GType type;
guint size,i;
gchar *error;
GValue val = {0,};

g_return_val_if_fail (dbus_g_type_is_struct (G_VALUE_TYPE (value)),
FALSE);

va_start (var_args, first_member);
size = dbus_g_type_get_struct_size (G_VALUE_TYPE (value));
i = first_member;
while (i != G_MAXUINT)
{
    if (i >= size)
        goto error;

    type = dbus_g_type_get_struct_member_type (G_VALUE_TYPE
(value),i);

    g_value_init (&val, type);

    G_VALUE_COLLECT (&val, var_args, 0, &error);
    if (error)
    {
        g_warning ("%s, %s", G_STRFUNC, error);
        g_free (error);
        g_value_unset (&val);
        goto error;
    }

    dbus_g_type_struct_set_member (value, i, &val);

    g_value_unset (&val);
    i = va_arg (var_args, guint);
}
va_end (var_args);
return TRUE;
error:
    va_end (var_args);
    return FALSE;
}

static void
_collection_iterator (const GValue *value,
                    gpointer user_data)
{
    GPtrArray *children = user_data;

    g_ptr_array_add (children, dbus_g_value_build_g_variant (value));
}

```

```

static void
_map_iterator (const GValue *kvalue,
              const GValue *vvalue,
              gpointer user_data)
{
    GPtrArray *children = user_data;

    g_ptr_array_add (children,
                    g_variant_new_dict_entry (
                        dbus_g_value_build_g_variant (kvalue),
                        dbus_g_value_build_g_variant (vvalue)));
}

static GVariantType *
dbus_g_value_type_build_g_variant_type (GType type)
{
    if (dbus_g_type_is_collection (type))
    {
        GType element_type = dbus_g_type_get_collection_specialization
            (type);
        GVariantType *element_sig =
            dbus_g_value_type_build_g_variant_type (
                element_type);
        GVariantType *ret = g_variant_type_new_array (element_sig);

        g_variant_type_free (element_sig);
        return ret;
    }
    else if (dbus_g_type_is_map (type))
    {
        GType key_type = dbus_g_type_get_map_key_specialization (type);
        GType value_type = dbus_g_type_get_map_value_specialization
            (type);
        GVariantType *key_sig = dbus_g_value_type_build_g_variant_type (
            key_type);
        GVariantType *value_sig = dbus_g_value_type_build_g_variant_type
            (
                value_type);
        GVariantType *entry_sig = g_variant_type_new_dict_entry
            (key_sig,
             value_sig);
        GVariantType *ret = g_variant_type_new_array (entry_sig);

        g_variant_type_free (key_sig);
        g_variant_type_free (value_sig);
        g_variant_type_free (entry_sig);
        return ret;
    }
    else if (dbus_g_type_is_struct (type))
    {
        guint size = dbus_g_type_get_struct_size (type);
        guint i;

```

```

GVariantType **sigs = g_new0 (GVariantType *, size);
GVariantType *ret;

for (i = 0; i < size; i++)
{
    GType t = dbus_g_type_get_struct_member_type (type, i);

    sigs[i] = dbus_g_value_type_build_g_variant_type (t);
}

ret = g_variant_type_new_tuple ((const GVariantType * const *)
sigs,
    size);

for (i = 0; i < size; i++)
    g_variant_type_free (sigs[i]);

g_free (sigs);
return ret;
}
else if (type == G_TYPE_BOOLEAN)
    return g_variant_type_copy (G_VARIANT_TYPE_BOOLEAN);
else if (type == G_TYPE_UCHAR)
    return g_variant_type_copy (G_VARIANT_TYPE_BYTE);
else if (type == G_TYPE_INT)
    return g_variant_type_copy (G_VARIANT_TYPE_INT32);
else if (type == G_TYPE_UINT)
    return g_variant_type_copy (G_VARIANT_TYPE_UINT32);
else if (type == G_TYPE_INT64)
    return g_variant_type_copy (G_VARIANT_TYPE_INT64);
else if (type == G_TYPE_UINT64)
    return g_variant_type_copy (G_VARIANT_TYPE_UINT64);
else if (type == G_TYPE_DOUBLE)
    return g_variant_type_copy (G_VARIANT_TYPE_DOUBLE);
else if (type == G_TYPE_STRING)
    return g_variant_type_copy (G_VARIANT_TYPE_STRING);
else if (type == G_TYPE_STRV)
    return g_variant_type_copy (G_VARIANT_TYPE_STRING_ARRAY);
else if (type == DBUS_TYPE_G_OBJECT_PATH)
    return g_variant_type_copy (G_VARIANT_TYPE_OBJECT_PATH);
else if (type == DBUS_TYPE_G_SIGNATURE)
    return g_variant_type_copy (G_VARIANT_TYPE_SIGNATURE);
else if (type == G_TYPE_VALUE)
    return g_variant_type_copy (G_VARIANT_TYPE_VARIANT);
else
    g_error ("%s: Unknown type: %s", G_STRFUNC, g_type_name (type));
}

/**
 * dbus_g_value_build_g_variant:
 * @value: a simple or specialized #GValue to convert to a #GVariant
 *

```

```

* Recurses @value and converts its contents to a #GVariant.
*
* The value must either be a simple value (integer, string, boolean,
* object path etc.) or a specialized container registered with
* dbus_g_type_get_collection(), dbus_g_type_get_map() or
* dbus_g_type_get_struct(). Providing any other type is a programming
error
* (including as a child type).
*
* Returns: a new #GVariant containing @value with a floating
reference
*/
GVariant *
dbus_g_value_build_g_variant (const GValue *value)
{
    GType type;

    g_return_val_if_fail (G_IS_VALUE (value), NULL);

    type = G_VALUE_TYPE (value);

    if (dbus_g_type_is_collection (type))
    {
        GVariant *variant;
        GPtrArray *children;
        GVariantType *signature = NULL;

        children = g_ptr_array_new ();
        dbus_g_type_collection_value_iterate (value,
        _collection_iterator,
        children);

        if (children->len == 0)
        {
            /* we can't cheat by saying "the type of the children?
that!" */
            GType element_type =
dbus_g_type_get_collection_specialization (
                type);

            signature = dbus_g_value_type_build_g_variant_type
(element_type);
        }

        variant = g_variant_new_array (signature, (GVariant **)
children->pdata,
        children->len);
        g_ptr_array_free (children, TRUE);
        g_variant_type_free (signature);

        return variant;
    }
}

```

```

else if (dbus_g_type_is_map (type))
{
    GVariant *variant;
    GPtrArray *children;
    GVariantType *signature = NULL;

    children = g_ptr_array_new ();
    dbus_g_type_map_value_iterate (value, _map_iterator, children);

    if (children->len == 0)
    {
        /* we can't cheat by saying "the type of the children?
that!" */
        GType key_type = dbus_g_type_get_map_key_specialization (
            type);
        GType value_type = dbus_g_type_get_map_value_specialization
(
            type);
        GVariantType *k = dbus_g_value_type_build_g_variant_type
(key_type);
        GVariantType *v = dbus_g_value_type_build_g_variant_type (
            value_type);

        signature = g_variant_type_new_dict_entry (k, v);
        g_variant_type_free (k);
        g_variant_type_free (v);
    }

    variant = g_variant_new_array (signature, (GVariant **)
children->pdata,
        children->len);
    g_ptr_array_free (children, TRUE);
    g_variant_type_free (signature);

    return variant;
}
else if (dbus_g_type_is_struct (type))
{
    GVariant *variant, **children;
    guint size, i;

    size = dbus_g_type_get_struct_size (type);
    children = g_new0 (GVariant *, size);

    for (i = 0; i < size; i++)
    {
        GValue cvalue = { 0, };

        g_value_init (&cvalue, dbus_g_type_get_struct_member_type
(type, i));
        dbus_g_type_struct_get_member (value, i, &cvalue);

```

```

        children[i] = dbus_g_value_build_g_variant (&cvalue);
        g_value_unset (&cvalue);
    }

    variant = g_variant_new_tuple (children, size);
    g_free (children);

    return variant;
}
else if (type == G_TYPE_BOOLEAN)
    return g_variant_new_boolean (g_value_get_boolean (value));
else if (type == G_TYPE_UCHAR)
    return g_variant_new_byte (g_value_get_uchar (value));
else if (type == G_TYPE_INT)
    return g_variant_new_int32 (g_value_get_int (value));
else if (type == G_TYPE_UINT)
    return g_variant_new_uint32 (g_value_get_uint (value));
else if (type == G_TYPE_INT64)
    return g_variant_new_int64 (g_value_get_int64 (value));
else if (type == G_TYPE_UINT64)
    return g_variant_new_uint64 (g_value_get_uint64 (value));
else if (type == G_TYPE_DOUBLE)
    return g_variant_new_double (g_value_get_double (value));
else if (type == G_TYPE_STRING)
    return g_variant_new_string (g_value_get_string (value));
else if (type == G_TYPE_STRV)
    return g_variant_new_strv (g_value_get_boxed (value), -1);
else if (type == DBUS_TYPE_G_OBJECT_PATH)
    return g_variant_new_object_path (g_value_get_boxed (value));
else if (type == DBUS_TYPE_G_SIGNATURE)
    return g_variant_new_signature (g_value_get_boxed (value));
else if (type == G_TYPE_VALUE)
    return g_variant_new_variant (
        dbus_g_value_build_g_variant (g_value_get_boxed (value)));
else
    {
        g_error ("%s: Unknown type: %s", G_STRFUNC, g_type_name (type));
    }
}

```

File = dbus-gtype-specialized.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gtype-specialized.h: Non-DBus-specific functions for
specialized GTypes
*
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*

```

```

* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/

```

```

#ifndef DBUS_GOBJECT_TYPE_SPECIALIZED_H
#define DBUS_GOBJECT_TYPE_SPECIALIZED_H

#include <glib.h>
#include <glib-object.h>

G_BEGIN_DECLS

GType          dbus_g_type_get_collection          (const
char *container,
                                                    GType
specialization);
GType          dbus_g_type_get_map                (const
char *container,
                                                    GType
key_specialization,
                                                    GType
value_specialization);
GType          dbus_g_type_get_structv           (const
char *container,
                                                    guint    num_members,
                                                    GType    *types);
GType          dbus_g_type_get_struct            (const
char *container,
                                                    GType
first_type,
                                                    ...);
gboolean       dbus_g_type_is_collection         (GType
gtype);
gboolean       dbus_g_type_is_map                (GType
gtype);
gboolean       dbus_g_type_is_struct             (GType
gtype);

```



```

gboolean      dbus_g_type_collection_get_fixed      (GValue
*value,
                                                    gpointer
*data_ret,
                                                    guint
*len_ret);

void          dbus_g_type_collection_value_iterate  (const
GValue      *value,
DBusGTypeSpecializedCollectionIterator  iterator,
                                                    gpointer
user_data);

void          dbus_g_type_map_value_iterate        (const
GValue      *value,
DBusGTypeSpecializedMapIterator          iterator,
                                                    gpointer
user_data);

gboolean      dbus_g_type_struct_get_member        (const GValue
*value,
                                                    guint member,
GValue
*dest);
gboolean      dbus_g_type_struct_set_member        (GValue
*value,
                                                    guint member,
const GValue
*src);

gboolean      dbus_g_type_struct_get               (const GValue
*value,
                                                    guint member,
...);

gboolean      dbus_g_type_struct_set               (GValue
*value,
                                                    guint member,
...);

typedef gpointer (*DBusGTypeSpecializedConstructor) (GType type);
typedef void     (*DBusGTypeSpecializedFreeFunc)   (GType type,
gpointer val);
typedef gpointer (*DBusGTypeSpecializedCopyFunc)   (GType type,
gpointer src);

typedef struct {
    DBusGTypeSpecializedConstructor  constructor;
    DBusGTypeSpecializedFreeFunc     free_func;
    DBusGTypeSpecializedCopyFunc     copy_func;

```

```

    GDestroyNotify                simple_free_func; /* for type-
independent freeing if possible */
    /*<private>*/
    gpointer                       padding2;
    gpointer                       padding3;
} DBusGTypeSpecializedVtable;

typedef gboolean (*DBusGTypeSpecializedCollectionFixedAccessorFunc)
(GType type, gpointer instance, gpointer *values, guint *len);
typedef void     (*DBusGTypeSpecializedCollectionIteratorFunc)
(GType type, gpointer instance, DBusGTypeSpecializedCollectionIterator
iterator, gpointer user_data);
typedef void     (*DBusGTypeSpecializedCollectionAppendFunc)
(DBusGTypeSpecializedAppendContext *ctx, GValue *val);
typedef void     (*DBusGTypeSpecializedCollectionEndAppendFunc)
(DBusGTypeSpecializedAppendContext *ctx);

typedef struct {
    DBusGTypeSpecializedVtable                base_vtable;
    DBusGTypeSpecializedCollectionFixedAccessorFunc  fixed_accessor;
    DBusGTypeSpecializedCollectionIteratorFunc      iterator;
    DBusGTypeSpecializedCollectionAppendFunc        append_func;
    DBusGTypeSpecializedCollectionEndAppendFunc     end_append_func;
} DBusGTypeSpecializedCollectionVtable;

typedef void (*DBusGTypeSpecializedMapIteratorFunc) (GType type,
gpointer instance, DBusGTypeSpecializedMapIterator iterator, gpointer
user_data);
typedef void (*DBusGTypeSpecializedMapAppendFunc)
(DBusGTypeSpecializedAppendContext *ctx, GValue *key, GValue *val);

typedef struct {
    DBusGTypeSpecializedVtable                base_vtable;
    DBusGTypeSpecializedMapIteratorFunc      iterator;
    DBusGTypeSpecializedMapAppendFunc        append_func;
} DBusGTypeSpecializedMapVtable;

typedef gboolean (*DBusGTypeSpecializedStructGetMember) (GType type,
gpointer instance, guint member, GValue *ret_value);
typedef gboolean (*DBusGTypeSpecializedStructSetMember) (GType type,
gpointer instance, guint member, const GValue *new_value);

typedef struct {
    DBusGTypeSpecializedVtable                base_vtable;
    DBusGTypeSpecializedStructGetMember      get_member;
    DBusGTypeSpecializedStructSetMember      set_member;
} DBusGTypeSpecializedStructVtable;

void                dbus_g_type_specialized_init                (void);

void                dbus_g_type_register_collection              (const char
*name,
```

```

        const
DBusGTypeSpecializedCollectionVtable *vtable,
        guint
flags);

void          dbus_g_type_register_map          (const char
*name,
        const
DBusGTypeSpecializedMapVtable *vtable,
        guint
flags);

void          dbus_g_type_register_struct      (const char
*name,
        const
DBusGTypeSpecializedStructVtable *vtable,
        guint
flags);

const DBusGTypeSpecializedMapVtable* dbus_g_type_map_peek_vtable
(GType map_type);
const DBusGTypeSpecializedCollectionVtable*
dbus_g_type_collection_peek_vtable (GType collection_type);

const DBusGTypeSpecializedStructVtable* dbus_g_type_struct_peek_vtable
(GType struct_type);

GVariant *dbus_g_value_build_g_variant (const GValue *value);

G_END_DECLS

#endif

```

File = dbus-gtype-specialized.sgml

```

<!-- ##### SECTION Title ##### -->
Specializable GType System

```

```

<!-- ##### SECTION Short_Description ##### -->

```

```

<!-- ##### SECTION Long_Description ##### -->

```

```

<para>

```

```

</para>

```

```

<!-- ##### SECTION See_Also ##### -->

```

```

<para>

```

</para>

<!-- ##### SECTION Stability_Level ##### -->

<!-- ##### SECTION Image ##### -->

<!-- ##### USER_FUNCTION DBusGTypeSpecializedCollectionIterator ##### -->

<para>

</para>

@value:
@user_data:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedMapIterator ##### -->

<para>

</para>

@key_val:
@value_val:
@user_data:

<!-- ##### STRUCT DBusGTypeSpecializedAppendContext ##### -->

<para>

</para>

@val:
@specialization_type:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedConstructor ##### -->

<para>

</para>

@type:
@Returns:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedFreeFunc ##### -->

<para>

</para>

@type:

@val:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedCopyFunc ##### -->

<para>

</para>

@type:

@src:

@Returns:

<!-- ##### STRUCT DBusGTypeSpecializedVtable ##### -->

<para>

</para>

@constructor:

@free_func:

@copy_func:

@simple_free_func:

<!-- ##### USER_FUNCTION

DBusGTypeSpecializedCollectionFixedAccessorFunc ##### -->

<para>

</para>

@type:

@instance:

@values:

@len:

@Returns:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedCollectionIteratorFunc

-->

<para>

</para>

@type:

@instance:

@iterator:

@user_data:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedCollectionAppendFunc

-->

<para>

</para>

@ctx:

@val:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedCollectionEndAppendFunc
-->

<para>

</para>

@ctx:

<!-- ##### STRUCT DBusGTypeSpecializedCollectionVtable ##### -->

<para>

</para>

@base_vtable:

@fixed_accessor:

@iterator:

@append_func:

@end_append_func:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedMapIteratorFunc ##### -->

<para>

</para>

@type:

@instance:

@iterator:

@user_data:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedMapAppendFunc ##### -->

<para>

</para>

@ctx:

@key:

@val:

<!-- ##### STRUCT DBusGTypeSpecializedMapVtable ##### -->

<para>

</para>

@base_vtable:
@iterator:
@append_func:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedStructGetMember ##### -->
<para>

</para>

@type:
@instance:
@member:
@ret_value:
@Returns:

<!-- ##### USER_FUNCTION DBusGTypeSpecializedStructSetMember ##### -->
<para>

</para>

@type:
@instance:
@member:
@new_value:
@Returns:

<!-- ##### STRUCT DBusGTypeSpecializedStructVtable ##### -->
<para>

</para>

@base_vtable:
@get_member:
@set_member:

<!-- ##### FUNCTION dbus_g_type_get_collection ##### -->
<para>

</para>

@container:
@specialization:
@Returns:

<!-- ##### FUNCTION dbus_g_type_get_map ##### -->
<para>

</para>

@container:
@key_specialization:
@value_specialization:
@Returns:

<!-- ##### FUNCTION dbus_g_type_get_structv ##### -->
<para>

</para>

@container:
@num_members:
@types:
@Returns:

<!-- ##### FUNCTION dbus_g_type_get_struct ##### -->
<para>

</para>

@container:
@first_type:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_type_is_collection ##### -->
<para>

</para>

@gtype:
@Returns:

<!-- ##### FUNCTION dbus_g_type_is_map ##### -->
<para>

</para>

@gtype:
@Returns:

<!-- ##### FUNCTION dbus_g_type_is_struct ##### -->
<para>

</para>

@gtype:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_get_collection_specialization ##### -->  
<para>
```

</para>

</para>

@gtype:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_get_map_key_specialization ##### -->
```

<para>

</para>

@gtype:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_get_map_value_specialization ##### -->
```

<para>

</para>

@gtype:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_get_struct_member_type ##### -->
```

<para>

</para>

@gtype:

@member:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_get_struct_size ##### -->
```

<para>

</para>

@gtype:

@Returns:

```
<!-- ##### FUNCTION dbus_g_type_specialized_construct ##### -->
```

<para>

</para>

@gtype:
@Returns:

<!-- ##### FUNCTION dbus_g_type_specialized_init_append ##### -->
<para>

</para>

@value:
@ctx:

<!-- ##### FUNCTION dbus_g_type_specialized_collection_append ##### -->
>

<para>

</para>

@ctx:
@elt:

<!-- ##### FUNCTION dbus_g_type_specialized_collection_end_append ##### -->
<para>

</para>

</para>

@ctx:

<!-- ##### FUNCTION dbus_g_type_specialized_map_append ##### -->
<para>

</para>

@ctx:
@key:
@val:

<!-- ##### FUNCTION dbus_g_type_collection_get_fixed ##### -->
<para>

</para>

@value:
@data_ret:

@len_ret:
@Returns:

<!-- ##### FUNCTION dbus_g_type_collection_value_iterate ##### -->
<para>

</para>

@value:
@iterator:
@user_data:

<!-- ##### FUNCTION dbus_g_type_map_value_iterate ##### -->
<para>

</para>

@value:
@iterator:
@user_data:

<!-- ##### FUNCTION dbus_g_type_struct_get_member ##### -->
<para>

</para>

@value:
@member:
@dest:
@Returns:

<!-- ##### FUNCTION dbus_g_type_struct_set_member ##### -->
<para>

</para>

@value:
@member:
@src:
@Returns:

<!-- ##### FUNCTION dbus_g_type_struct_get ##### -->
<para>

</para>

@value:

@member:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_type_struct_set ##### -->

<para>

</para>

@value:
@member:
@...:
@Returns:

<!-- ##### FUNCTION dbus_g_type_specialized_init ##### -->

<para>

</para>

@void:

<!-- ##### FUNCTION dbus_g_type_register_collection ##### -->

<para>

</para>

@name:
@vtable:
@flags:

<!-- ##### FUNCTION dbus_g_type_register_map ##### -->

<para>

</para>

@name:
@vtable:
@flags:

<!-- ##### FUNCTION dbus_g_type_map_peek_vtable ##### -->

<para>

</para>

@map_type:
@Returns:

```
<!-- ##### FUNCTION dbus_g_type_collection_peek_vtable ##### -->
<para>
```

```
</para>
```

```
@collection_type:
@Returns:
```

```
<!-- ##### FUNCTION dbus_g_type_struct_peek_vtable ##### -->
<para>
```

```
</para>
```

```
@struct_type:
@Returns:
```

```
<!-- ##### FUNCTION dbus_g_type_register_struct ##### -->
<para>
```

```
</para>
```

```
@name:
@vtable:
@flags:
```

```
<!-- ##### FUNCTION dbus_g_value_build_g_variant ##### -->
<para>
```

```
</para>
```

```
@value:
@Returns:
```

```
<!-- ##### FUNCTION dbus_g_value_parse_g_variant ##### -->
<para>
```

```
</para>
```

```
@variant:
@value:
```

```
<!-- ##### MACRO DBUS_TYPE_G_BOOLEAN_ARRAY ##### -->
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_UCHAR_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_UINT_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_INT_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_UINT64_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_INT64_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_OBJECT_ARRAY ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### MACRO DBUS_TYPE_G_STRING_STRING_HASHTABLE ##### -->  
<para>
```

```
</para>
```

```
<!-- ##### TYPEDEF DBusGSignature ##### -->
```

<para>

</para>

<!-- ##### MACRO DBUS_TYPE_G_SIGNATURE ##### -->

<para>

</para>

<!-- ##### TYPEDEF DBusGObjectPath ##### -->

<para>

</para>

<!-- ##### MACRO DBUS_TYPE_G_OBJECT_PATH ##### -->

<para>

</para>

File = dbus-gutils.c

/* -*- mode: C; c-file-style: "gnu" -*- */

/* dbus-gutils.c Utils shared between convenience lib and installed lib

*

* Copyright (C) 2003 Red Hat, Inc.

*

* Licensed under the Academic Free License version 2.1

*

* This program is free software; you can redistribute it and/or modify

* it under the terms of the GNU General Public License as published by

* the Free Software Foundation; either version 2 of the License, or

* (at your option) any later version.

*

* This program is distributed in the hope that it will be useful,

* but WITHOUT ANY WARRANTY; without even the implied warranty of

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

* GNU General Public License for more details.

*

* You should have received a copy of the GNU General Public License

* along with this program; if not, write to the Free Software

```
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-gutils.h"
#include "dbus-gtest.h"
#include <string.h>
```

```
char**
_dbus_gutils_split_path (const char *path)
{
    int len;
    char **split;
    int n_components;
    int i, j, comp;

    len = strlen (path);

    n_components = 0;
    if (path[1] != '\0') /* if not "/" */
    {
        i = 0;
        while (i < len)
        {
            if (path[i] == '/')
                n_components += 1;
            ++i;
        }
    }

    split = g_new0 (char*, n_components + 1);

    comp = 0;
    if (n_components == 0)
        i = 1;
    else
        i = 0;
    while (comp < n_components)
    {
        if (path[i] == '/')
            ++i;
        j = i;

        while (j < len && path[j] != '/')
            ++j;

        /* Now [i, j) is the path component */
        g_assert (i < j);
        g_assert (path[i] != '/');
        g_assert (j == len || path[j] == '/');
    }
}
```



```

        split[comp] = g_strdup (&path[i], j - i + 1);

        split[comp][j-i] = '\0';

        ++comp;
        i = j;
    }
    g_assert (i == len);

    return split;
}

char*
_dbus_gutils_wincaps_to_uscore (const char *caps)
{
    const char *p;
    GString *str;

    str = g_string_new (NULL);
    p = caps;
    while (*p)
    {
        if (g_ascii_isupper (*p))
        {
            if (str->len > 0 &&
                (str->len < 2 || str->str[str->len-2] != '_'))
                g_string_append_c (str, '_');
            g_string_append_c (str, g_ascii_tolower (*p));
        }
        else
        {
            g_string_append_c (str, *p);
        }
        ++p;
    }

    return g_string_free (str, FALSE);
}

```

File = dbus-gutils.h

```

/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gutils.h Utils shared between convenience lib and installed
lib
*
* Copyright (C) 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*

```

```

* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/

```

```

#ifndef DBUS_GLIB_UTILS_H
#define DBUS_GLIB_UTILS_H

```

```

#include <dbus/dbus.h>
#include <glib-object.h>

```

```

G_BEGIN_DECLS

```

```

char      **_dbus_gutils_split_path      (const char *path);

```

```

char      *_dbus_gutils_wincaps_to_uscore (const char *uscore);

```

```

/* These munge the pointer to enforce that a plain cast won't work,
* accessor functions must be used; i.e. to ensure the ABI
* reflects our encapsulation.
*/

```

```

#define _DBUS_POINTER_SHIFT(p)      ((void*) (((char*)p) + sizeof
(void*))

```

```

#define _DBUS_POINTER_UNSHIFT(p)    ((void*) (((char*)p) - sizeof
(void*))

```

```

#define DBUS_CONNECTION_FROM_G_CONNECTION(x)      ((DBusConnection*)
_DBUS_POINTER_UNSHIFT(x))

```

```

#define DBUS_MESSAGE_FROM_G_MESSAGE(x)           ((DBusMessage*)
_DBUS_POINTER_UNSHIFT(x))

```

```

#define DBUS_PENDING_CALL_FROM_G_PENDING_CALL(x) ((DBusPendingCall*)
_DBUS_POINTER_UNSHIFT(x))

```

```

#define DBUS_G_CONNECTION_FROM_CONNECTION(x)      ((DBusGConnection*)
_DBUS_POINTER_SHIFT(x))

```

```

#define DBUS_G_MESSAGE_FROM_MESSAGE(x)           ((DBusGMessage*)
_DBUS_POINTER_SHIFT(x))

```

```

#define DBUS_G_PENDING_CALL_FROM_PENDING_CALL(x) ((DBusGPendingCall*)
_DBUS_POINTER_SHIFT(x))

G_END_DECLS

#endif /* DBUS_GLIB_UTILS_H */

File = dbus-gvalue-parse-variant.c

/* GVariant to dbus-glib escape hatch
 *
 * Copyright © 2010 Collabora Ltd. <http://www.collabora.co.uk/>
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * Alternatively, at your option, you can redistribute and/or modify
 * this single file under the terms of the GNU Lesser General Public
License
 * as published by the Free Software Foundation; either version 2.1 of
 * that license, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>

#include <dbus/dbus-gvalue-parse-variant.h>

#include <string.h>

#include <dbus/dbus-glib.h>

/* Static functions in this file are a bit weird: they take a GVariant
as first

```

```
* argument, but it can be NULL. If it is, @value will be initialized
with the
* right type for that GVariant, but not filled in, so it'll contain 0
or NULL
* or whatever. */
```

```
static void dbus_g_value_parse_variant_by_type (GVariant *variant,
        const GVariantType *variant_type,
        GValue *value);
```

```
static void
dbus_g_value_dict_parse_variant (GVariant *variant,
        const GVariantType *member_type,
        GValue *value)
{
    const GVariantType *key_type, *value_type;
    GValue key_parsed = { 0 }, value_parsed = { 0 };

    g_assert (g_variant_type_is_dict_entry (member_type));
    key_type = g_variant_type_key (member_type);
    value_type = g_variant_type_value (member_type);

    /* first get the GTypes, without getting actual values */
    dbus_g_value_parse_variant_by_type (NULL, key_type, &key_parsed);
    dbus_g_value_parse_variant_by_type (NULL, value_type,
    &value_parsed);

    g_value_init (value, dbus_g_type_get_map ("GHashTable",
        G_VALUE_TYPE (&key_parsed),
        G_VALUE_TYPE (&value_parsed)));
    g_value_unset (&key_parsed);
    g_value_unset (&value_parsed);

    if (variant != NULL)
    {
        GVariantIter iter;
        GVariant *child;
        DBusGTypeSpecializedAppendContext ctx;

        g_value_take_boxed (value, dbus_g_type_specialized_construct (
            G_VALUE_TYPE (value)));
        dbus_g_type_specialized_init_append (value, &ctx);

        g_variant_iter_init (&iter, variant);

        for (child = g_variant_iter_next_value (&iter);
            child != NULL;
            child = g_variant_iter_next_value (&iter))
        {
            GVariant *grandchild;

            grandchild = g_variant_get_child_value (child, 0);
```

```

        dbus_g_value_parse_variant_by_type (grandchild, key_type,
                                           &key_parsed);
        g_variant_unref (grandchild);

        grandchild = g_variant_get_child_value (child, 1);
        dbus_g_value_parse_variant_by_type (grandchild, value_type,
                                           &value_parsed);
        g_variant_unref (grandchild);

        /* Here be dragons: this steals the *contents of* key_parsed
and
        * value_parsed, so we can't g_value_unset() them. */
        dbus_g_type_specialized_map_append (&ctx, &key_parsed,
                                           &value_parsed);
        memset (&key_parsed, '\0', sizeof (key_parsed));
        memset (&value_parsed, '\0', sizeof (value_parsed));
        g_variant_unref (child);
    }
}

static void
dbus_g_value_basic_array_parse_variant (GVariant *variant,
                                       gchar type_char,
                                       GValue *value)
{
    GType gtype = G_TYPE_INVALID;
    guint dg_size = 0, gv_size = 0;

    switch ((GVariantClass) type_char)
    {
        case G_VARIANT_CLASS_STRING:
            g_value_init (value, G_TYPE_STRV);

            if (variant != NULL)
                g_value_take_boxed (value, g_variant_dup_strv (variant,
NULL));
            return;

        case G_VARIANT_CLASS_OBJECT_PATH:
        case G_VARIANT_CLASS_SIGNATURE:
            {
                if (type_char == G_VARIANT_CLASS_OBJECT_PATH)
                    gtype = DBUS_TYPE_G_OBJECT_PATH;
                else
                    gtype = DBUS_TYPE_G_SIGNATURE;

                g_value_init (value,
                             dbus_g_type_get_collection ("GPtrArray", gtype));

                if (variant != NULL)
                    {

```

```

        gsize n = g_variant_n_children (variant);
        gsize i;
        GPtrArray *pa = g_ptr_array_sized_new (n);

        for (i = 0; i < n; i++)
        {
            GVariant *child = g_variant_get_child_value
(variant, i);
            gchar *s = g_variant_dup_string (child, NULL);

            g_ptr_array_add (pa, s);
            g_variant_unref (child);
        }

        g_value_take_boxed (value, pa);
    }
}
return;

/* From here down handles fixed-size types. */

case G_VARIANT_CLASS_BYTE:
    gtype = G_TYPE_UCHAR;
    dg_size = dg_size = sizeof (guchar);
    break;

case G_VARIANT_CLASS_BOOLEAN:
    gtype = G_TYPE_BOOLEAN;
    dg_size = sizeof (gboolean);
    gv_size = sizeof (guchar);
    break;

case G_VARIANT_CLASS_INT16:
    gtype = G_TYPE_INT;
    dg_size = sizeof (gint);
    gv_size = sizeof (gint16);
    break;

case G_VARIANT_CLASS_INT32:
    gtype = G_TYPE_INT;
    dg_size = sizeof (gint);
    gv_size = sizeof (gint32);
    break;

case G_VARIANT_CLASS_UINT16:
    gtype = G_TYPE_UINT;
    dg_size = sizeof (guint);
    gv_size = sizeof (guint16);
    break;

case G_VARIANT_CLASS_UINT32:
    gtype = G_TYPE_UINT;

```

```

    dg_size = sizeof (guint);
    gv_size = sizeof (guint32);
    break;

case G_VARIANT_CLASS_INT64:
    gtype = G_TYPE_INT64;
    dg_size = gv_size = sizeof (gint64);
    break;

case G_VARIANT_CLASS_UINT64:
    gtype = G_TYPE_UINT64;
    dg_size = gv_size = sizeof (guint64);
    break;

case G_VARIANT_CLASS_DOUBLE:
    gtype = G_TYPE_DOUBLE;
    dg_size = gv_size = sizeof (gdouble);
    break;

case G_VARIANT_CLASS_HANDLE:
case G_VARIANT_CLASS_VARIANT:
case G_VARIANT_CLASS_MAYBE:
case G_VARIANT_CLASS_ARRAY:
case G_VARIANT_CLASS_TUPLE:
case G_VARIANT_CLASS_DICT_ENTRY:
    g_return_if_reached ();
}

g_assert (gtype != G_TYPE_INVALID);
g_assert (dg_size != 0);
g_assert (gv_size != 0);

g_value_init (value, dbus_g_type_get_collection ("GArray", gtype));

if (variant != NULL)
{
    GArray *arr;
    gsize n, i;
    gconstpointer blob = g_variant_get_fixed_array (variant, &n,
gv_size);

    arr = g_array_sized_new (FALSE, FALSE, dg_size, n);
    g_value_take_boxed (value, arr);

    if (dg_size == gv_size)
    {
        /* fast path: we can just memcpy them in */
        g_array_append_vals (arr, blob, n);
    }
    else
    {
        DBusGTypeSpecializedAppendContext ctx;

```

```

    dbus_g_type_specialized_init_append (value, &ctx);

    for (i = 0; i < n; i++)
    {
        GVariant *child;
        GValue v = { 0 };

        child = g_variant_get_child_value (variant, i);
        dbus_g_value_parse_g_variant (child, &v);
        g_variant_unref (child);
        dbus_g_type_specialized_collection_append (&ctx, &v);
    }

    dbus_g_type_specialized_collection_end_append (&ctx);
}
}

static void
dbus_g_value_tuple_parse_variant (GVariant *variant,
    const GVariantType *variant_type,
    GValue *value)
{
    gsize n = g_variant_type_n_items (variant_type);
    GType *types;
    gsize i;
    GValueArray *va = g_value_array_new (n);
    const GVariantType *inner_type;

    types = g_new0 (GType, n);

    for (i = 0, inner_type = g_variant_type_first (variant_type);
        i < n;
        i++, inner_type = g_variant_type_next (inner_type))
    {
        GVariant *inner_variant;

        if (variant == NULL)
            inner_variant = NULL;
        else
            inner_variant = g_variant_get_child_value (variant, i);

        g_value_array_append (va, NULL);
        dbus_g_value_parse_variant_by_type (inner_variant, inner_type,
            &va->values[i]);
        types[i] = G_VALUE_TYPE (&va->values[i]);

        if (inner_variant != NULL)
            g_variant_unref (inner_variant);
    }
}

```



```

    g_value_init (value, dbus_g_type_get_structv ("GValueArray", n,
types));

    if (variant == NULL)
        g_value_array_free (va);
    else
        g_value_take_boxed (value, va);

    g_free (types);
}

static void
dbus_g_value_array_parse_variant (GVariant *variant,
    const GVariantType *variant_type,
    GValue *value)
{
    const GVariantType *member_type;
    gchar type_char;
    GPtrArray *pa = NULL;
    gsize n = 0, i;

    g_assert (g_variant_type_is_array (variant_type));

    member_type = g_variant_type_element (variant_type);
    type_char = g_variant_type_peek_string (member_type)[0];

    if (g_variant_type_is_dict_entry (member_type))
    {
        dbus_g_value_dict_parse_variant (variant, member_type, value);
        return;
    }

    if (g_variant_type_is_basic (member_type))
    {
        dbus_g_value_basic_array_parse_variant (variant, type_char,
value);
        return;
    }

    /* all the non-basic types end up as a GPtrArray of boxed */
    if (variant != NULL)
    {
        n = g_variant_n_children (variant);
        pa = g_ptr_array_sized_new (n);
    }

    switch ((GVariantClass) type_char)
    {
        case G_VARIANT_CLASS_VARIANT:
            {
                g_value_init (value,

```

```

        dbus_g_type_get_collection ("GPtrArray",
G_TYPE_VALUE));
    }
    break;

    case G_VARIANT_CLASS_ARRAY:
    {
        GValue v = { 0 };

        dbus_g_value_array_parse_variant (NULL, member_type, &v);
        g_value_init (value, dbus_g_type_get_collection
("GPtrArray",
        G_VALUE_TYPE (&v)));
    }
    break;

    case G_VARIANT_CLASS_TUPLE:
    {
        GValue v = { 0 };

        dbus_g_value_tuple_parse_variant (NULL, member_type, &v);
        g_value_init (value, dbus_g_type_get_collection
("GPtrArray",
        G_VALUE_TYPE (&v)));
    }
    break;

    case G_VARIANT_CLASS_DICT_ENTRY:
    case G_VARIANT_CLASS_MAYBE:
    default:
        g_critical ("unhandled GVariantClass array<%d>", type_char);
        g_return_if_reached ();
}

if (variant != NULL)
{
    for (i = 0; i < n; i++)
    {
        GVariant *child = g_variant_get_child_value (variant, i);
        GValue tmp = { 0 };

        dbus_g_value_parse_g_variant (child, &tmp);
        g_ptr_array_add (pa, g_value_dup_boxed (&tmp));
        g_variant_unref (child);
        g_value_unset (&tmp);
    }

    g_value_take_boxed (value, pa);
}
}

static void

```

```

dbus_g_value_parse_variant_by_type (GVariant *variant,
    const GVariantType *variant_type,
    GValue *value)
{
    gchar type_char = g_variant_type_peek_string (variant_type)[0];

    switch ((GVariantClass) type_char)
    {
        case G_VARIANT_CLASS_BOOLEAN:
            g_value_init (value, G_TYPE_BOOLEAN);

            if (variant != NULL)
                g_value_set_boolean (value, !!g_variant_get_boolean
(variant));

            break;

        case G_VARIANT_CLASS_BYTE:
            g_value_init (value, G_TYPE_UCHAR);

            if (variant != NULL)
                g_value_set_uchar (value, g_variant_get_byte (variant));

            break;

        case G_VARIANT_CLASS_UINT16:
            /* there is no G_TYPE_UINT16 */
            g_value_init (value, G_TYPE_UINT);

            if (variant != NULL)
                g_value_set_uint (value, g_variant_get_uint16 (variant));

            break;

        case G_VARIANT_CLASS_UINT32:
            g_value_init (value, G_TYPE_UINT);

            if (variant != NULL)
                g_value_set_uint (value, g_variant_get_uint32 (variant));

            break;

        case G_VARIANT_CLASS_UINT64:
            g_value_init (value, G_TYPE_UINT64);

            if (variant != NULL)
                g_value_set_uint64 (value, g_variant_get_uint64 (variant));

            break;

        case G_VARIANT_CLASS_INT16:
            /* there is no G_TYPE_INT16 */

```

```

    g_value_init (value, G_TYPE_INT);

    if (variant != NULL)
        g_value_set_int (value, g_variant_get_int16 (variant));

    break;

case G_VARIANT_CLASS_INT32:
    g_value_init (value, G_TYPE_INT);

    if (variant != NULL)
        g_value_set_int (value, g_variant_get_int32 (variant));

    break;

case G_VARIANT_CLASS_INT64:
    g_value_init (value, G_TYPE_INT64);

    if (variant != NULL)
        g_value_set_int64 (value, g_variant_get_int64 (variant));

    break;

case G_VARIANT_CLASS_DOUBLE:
    g_value_init (value, G_TYPE_DOUBLE);

    if (variant != NULL)
        g_value_set_double (value, g_variant_get_double (variant));

    break;

case G_VARIANT_CLASS_STRING:
    g_value_init (value, G_TYPE_STRING);

    if (variant != NULL)
        g_value_set_string (value, g_variant_get_string (variant,
NULL));

    break;

case G_VARIANT_CLASS_OBJECT_PATH:
    g_value_init (value, DBUS_TYPE_G_OBJECT_PATH);

    if (variant != NULL)
        g_value_set_boxed (value, g_variant_get_string (variant,
NULL));

    break;

case G_VARIANT_CLASS_SIGNATURE:
    g_value_init (value, DBUS_TYPE_G_SIGNATURE);

```

```

        if (variant != NULL)
            g_value_set_boxed (value, g_variant_get_string (variant,
NULL));

        break;

    case G_VARIANT_CLASS_VARIANT:
        g_value_init (value, G_TYPE_VALUE);

        if (variant != NULL)
            {
                GVariant *inner_variant = g_variant_get_variant (variant);
                GValue *inner_value = g_new0 (GValue, 1);

                dbus_g_value_parse_g_variant (inner_variant, inner_value);
                g_value_take_boxed (value, inner_value);
                g_variant_unref (inner_variant);
            }
        break;

    case G_VARIANT_CLASS_ARRAY:
        dbus_g_value_array_parse_variant (variant, variant_type,
value);
        break;

    case G_VARIANT_CLASS_TUPLE:
        dbus_g_value_tuple_parse_variant (variant, variant_type,
value);
        break;

    case G_VARIANT_CLASS_DICT_ENTRY:
        g_critical ("found a dict entry not in a dict");
        break;

    case G_VARIANT_CLASS_HANDLE:
    case G_VARIANT_CLASS_MAYBE:
        g_critical ("unhandled GVariantClass '%c' (%d)",
                    CLAMP (type_char, ' ', '~'),
                    type_char);
        break;
    }
}

/**
 * dbus_g_value_parse_g_variant:
 * @variant: a #GVariant
 * @value: a zero-filled #GValue
 *
 * Deserialize @variant and put an equivalent dbus-glib data structure
in
 * @value.
 *
 */

```

```
* It is an error if @variant contains any #GVariant extensions not
supported
* by dbus-glib, including handles (file descriptor passing) and
'maybe' types.
```

```
*/
void
dbus_g_value_parse_g_variant (GVariant *variant,
                              GValue *value)
{
    g_return_if_fail (variant != NULL);
    dbus_g_value_parse_variant_by_type (variant, g_variant_get_type
(variant),
    value);
}
```

File = dbus-gvalue-parse-variant.h

```
/* GVariant to dbus-glib escape hatch
*
* Copyright © 2010 Collabora Ltd. <http://www.collabora.co.uk/>
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* Alternatively, at your option, you can redistribute and/or modify
* this single file under the terms of the GNU Lesser General Public
License
* as published by the Free Software Foundation; either version 2.1 of
* that license, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#ifndef __DBUS_GVALUE_PARSE_VARIANT_H__
#define __DBUS_GVALUE_PARSE_VARIANT_H__
```

```
#include <glib-object.h>

G_BEGIN_DECLS

void dbus_g_value_parse_g_variant (GVariant *variant, GValue *value);

G_END_DECLS

#endif
```

File = dbus-gvalue-utils.c

```
/* -*- mode: C; c-file-style: "gnu" -*- */
/* dbus-gvalue-utils.c: Non-DBus-specific functions related to
GType/GValue
*
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus/dbus-glib.h"
#include "dbus-gtype-specialized-priv.h"
#include "dbus-gvalue-utils.h"
#include "dbus-gtest.h"
#include <glib.h>
#include <string.h>
#include <gobject/gvaluecollector.h>
```

```

static guint
fixed_type_get_size (GType type)
{
    switch (type)
    {
        case G_TYPE_CHAR:
        case G_TYPE_UCHAR:
            return sizeof (gchar);
        case G_TYPE_BOOLEAN:
            return sizeof (gboolean);
        case G_TYPE_LONG:
        case G_TYPE_ULONG:
            return sizeof (glong);
        case G_TYPE_INT:
        case G_TYPE_UINT:
            return sizeof (gint);
        case G_TYPE_INT64:
        case G_TYPE_UINT64:
            return sizeof (gint64);
        case G_TYPE_FLOAT:
            return sizeof (gfloat);
        case G_TYPE_DOUBLE:
            return sizeof (gdouble);
        default:
            return 0;
    }
}

```

```

gboolean
_dbus_g_type_is_fixed (GType type)
{
    return fixed_type_get_size (type) > 0;
}

```

```

guint
_dbus_g_type_fixed_get_size (GType type)
{
    g_assert (_dbus_g_type_is_fixed (type));
    return fixed_type_get_size (type);
}

```

```

gboolean
_dbus_gvalue_store (GValue          *value,
                   gpointer         storage)
{
    /* FIXME - can we use the GValue lcopy_value method
     * to do this in a cleaner way?
     */
    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_CHAR:
            *((gchar *) storage) = g_value_get_char (value);

```



```

    return TRUE;
case G_TYPE_UCHAR:
    *((guchar *) storage) = g_value_get_uchar (value);
    return TRUE;
case G_TYPE_BOOLEAN:
    *((gboolean *) storage) = g_value_get_boolean (value);
    return TRUE;
case G_TYPE_LONG:
    *((glong *) storage) = g_value_get_long (value);
    return TRUE;
case G_TYPE_ULONG:
    *((gulong *) storage) = g_value_get_ulong (value);
    return TRUE;
case G_TYPE_INT:
    *((gint *) storage) = g_value_get_int (value);
    return TRUE;
case G_TYPE_UINT:
    *((guint *) storage) = g_value_get_uint (value);
    return TRUE;
case G_TYPE_INT64:
    *((gint64 *) storage) = g_value_get_int64 (value);
    return TRUE;
case G_TYPE_UINT64:
    *((guint64 *) storage) = g_value_get_uint64 (value);
    return TRUE;
case G_TYPE_DOUBLE:
    *((gdouble *) storage) = g_value_get_double (value);
    return TRUE;
case G_TYPE_STRING:
    *((gchar **) storage) = (char*) g_value_get_string (value);
    return TRUE;
case G_TYPE_OBJECT:
    *((gpointer *) storage) = g_value_get_object (value);
    return TRUE;
case G_TYPE_BOXED:
    *((gpointer *) storage) = g_value_get_boxed (value);
    return TRUE;
default:
    return FALSE;
}
}

```

```

gboolean
_dbus_gvalue_set_from_pointer (GValue          *value,
                              gconstpointer    storage)
{
    /* FIXME - is there a better way to do this? */
    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_CHAR:
            g_value_set_char (value, *((gchar *) storage));
            return TRUE;
    }
}

```

```

case G_TYPE_UCHAR:
    g_value_set_uchar (value, *((guchar *) storage));
    return TRUE;
case G_TYPE_BOOLEAN:
    g_value_set_boolean (value, *((gboolean *) storage));
    return TRUE;
case G_TYPE_LONG:
    g_value_set_long (value, *((glong *) storage));
    return TRUE;
case G_TYPE_ULONG:
    g_value_set_ulong (value, *((gulong *) storage));
    return TRUE;
case G_TYPE_INT:
    g_value_set_int (value, *((gint *) storage));
    return TRUE;
case G_TYPE_UINT:
    g_value_set_uint (value, *((guint *) storage));
    return TRUE;
case G_TYPE_INT64:
    g_value_set_int64 (value, *((gint64 *) storage));
    return TRUE;
case G_TYPE_UINT64:
    g_value_set_uint64 (value, *((guint64 *) storage));
    return TRUE;
case G_TYPE_DOUBLE:
    g_value_set_double (value, *((gdouble *) storage));
    return TRUE;
case G_TYPE_STRING:
    g_value_set_string (value, *((gchar **) storage));
    return TRUE;
case G_TYPE_OBJECT:
    g_value_set_object (value, *((gpointer *) storage));
    return TRUE;
case G_TYPE_BOXED:
    g_value_set_boxed (value, *((gpointer *) storage));
    return TRUE;
default:
    return FALSE;
}
}

gboolean
_dbus_gvalue_take (GValue          *value,
                  GTypeCValue     *cvalue)
{
    GType g_type;
    GTypeValueTable *value_table;
    char *error_msg;

    g_type = G_VALUE_TYPE (value);
    value_table = g_type_value_table_peek (g_type);

```

```

    error_msg = value_table->collect_value (value, 1, cvalue,
G_VALUE_NOCOPY_CONTENTS);
    if (error_msg)
    {
        g_warning ("%s: %s", G_STRLOC, error_msg);
        g_free (error_msg);
        return FALSE;
    }
    /* Clear the NOCOPY_CONTENTS flag; we want to take ownership
    * of the value.
    */
    value->data[1].v_uint &= ~(G_VALUE_NOCOPY_CONTENTS);
    return TRUE;
}

```

```

gboolean
_dbus_gtype_can_signal_error (GType gtype)
{
    switch (gtype)
    {
        case G_TYPE_BOOLEAN:
        case G_TYPE_INT:
        case G_TYPE_UINT:
        case G_TYPE_STRING:
        case G_TYPE_BOXED:
        case G_TYPE_OBJECT:
            return TRUE;
        default:
            return FALSE;
    }
}

```

```

gboolean
_dbus_gvalue_signals_error (const GValue *value)
{
    /* Hardcoded rules for return value semantics for certain
    * types. Perhaps in the future we'd want an annotation
    * specifying which return values are errors, but in
    * reality people will probably just use boolean and
    * boxed, and there the semantics are pretty standard.
    */
    switch (G_TYPE_FUNDAMENTAL (G_VALUE_TYPE (value)))
    {
        case G_TYPE_BOOLEAN:
            return (g_value_get_boolean (value) == FALSE);
            break;
        case G_TYPE_INT:
            return (g_value_get_int (value) < 0);
            break;
        case G_TYPE_UINT:
            return (g_value_get_uint (value) == 0);
            break;
    }
}

```

```

    case G_TYPE_STRING:
        return (g_value_get_string (value) == NULL);
        break;
    case G_TYPE_BOXED:
        return (g_value_get_boxed (value) == NULL);
        break;
    case G_TYPE_OBJECT:
        return (g_value_get_object (value) == NULL);
        break;
    default:
        g_assert_not_reached ();
        return FALSE;
    }
}

```

```

static gboolean
hash_func_from_gtype (GType gtype, GHashFunc *func)
{
    switch (gtype)
    {
        case G_TYPE_CHAR:
        case G_TYPE_UCHAR:
        case G_TYPE_BOOLEAN:
        case G_TYPE_INT:
        case G_TYPE_UINT:
            *func = NULL;
            return TRUE;
        case G_TYPE_STRING:
            *func = g_str_hash;
            return TRUE;
        default:
            if (gtype == DBUS_TYPE_G_OBJECT_PATH)
            {
                *func = g_str_hash;
                return TRUE;
            }
            else if (gtype == DBUS_TYPE_G_SIGNATURE)
            {
                *func = g_str_hash;
                return TRUE;
            }
            return FALSE;
    }
}

```

```

static void
unset_and_free_g_value (gpointer val)
{
    GValue *value = val;

    g_value_unset (value);
}

```

```

    g_free (value);
}

static gboolean
gtype_can_simple_free (GType type);

static gboolean
hash_simple_free_from_gtype (GType gtype, GDestroyNotify *func)
{
    switch (gtype)
    {
        case G_TYPE_CHAR:
        case G_TYPE_UCHAR:
        case G_TYPE_BOOLEAN:
        case G_TYPE_INT:
        case G_TYPE_UINT:
            *func = NULL;
            return TRUE;
        case G_TYPE_DOUBLE:
        case G_TYPE_STRING:
            *func = g_free;
            return TRUE;
        default:
            if (gtype == G_TYPE_VALUE)
            {
                *func = unset_and_free_g_value;
                return TRUE;
            }
            else if (gtype == G_TYPE_VALUE_ARRAY)
            {
                *func = (GDestroyNotify) g_value_array_free;
                return TRUE;
            }
            else if (gtype == G_TYPE_STRV)
            {
                *func = (GDestroyNotify) g_strfreev;
                return TRUE;
            }
            else if (gtype == DBUS_TYPE_G_OBJECT_PATH)
            {
                *func = g_free;
                return TRUE;
            }
            else if (gtype == DBUS_TYPE_G_SIGNATURE)
            {
                *func = g_free;
                return TRUE;
            }
            else if (dbus_g_type_is_collection (gtype))
            {
                const DBusGTypeSpecializedCollectionVtable* vtable;
                vtable = dbus_g_type_collection_peek_vtable (gtype);
            }
    }
}

```

```

        if (vtable->base_vtable.simple_free_func)
        {
            *func = vtable->base_vtable.simple_free_func;
            return TRUE;
        }
    }
    else if (dbus_g_type_is_map (gtype))
    {
        const DBusGTypeSpecializedMapVtable* vtable;
        GType key_gtype, value_gtype;

        key_gtype = dbus_g_type_get_map_key_specialization (gtype);
        value_gtype = dbus_g_type_get_map_value_specialization
(gtype);

        /* if either the key or the value don't have "simple"
(without a
        * GType) free functions, then the hashtable's contents must
be freed
        * with hashtable_free, so the hashtable itself can't have a
simple
        * free function. */
        if (!gtype_can_simple_free (key_gtype) ||
            !gtype_can_simple_free (value_gtype))
            return FALSE;

        vtable = dbus_g_type_map_peek_vtable (gtype);
        if (vtable->base_vtable.simple_free_func)
        {
            *func = vtable->base_vtable.simple_free_func;
            return TRUE;
        }
    }
    else if (dbus_g_type_is_struct (gtype))
    {
        const DBusGTypeSpecializedStructVtable *vtable;
        vtable = dbus_g_type_struct_peek_vtable (gtype);
        if (vtable->base_vtable.simple_free_func)
        {
            *func = vtable->base_vtable.simple_free_func;
            return TRUE;
        }
    }
    return FALSE;
}

static gboolean
gtype_can_simple_free (GType type)
{
    GDestroyNotify func;
    return hash_simple_free_from_gtype (type, &func);
}

```

```

}

gboolean
_dbus_gtype_is_valid_hash_key (GType type)
{
    GHashFunc func;
    return hash_func_from_gtype (type, &func);
}

gboolean
_dbus_gtype_is_valid_hash_value (GType gtype)
{
    /* anything we can take into a GValue using gvalue_take_hash_value
    is fine */
    switch (g_type_fundamental (gtype))
    {
        case G_TYPE_CHAR:
        case G_TYPE_UCHAR:
        case G_TYPE_BOOLEAN:
        case G_TYPE_INT:
        case G_TYPE_UINT:
        case G_TYPE_DOUBLE:
        case G_TYPE_STRING:
        case G_TYPE_BOXED:
        case G_TYPE_OBJECT:
            return TRUE;
    }

    return FALSE;
}

GHashFunc
_dbus_g_hash_func_from_gtype (GType gtype)
{
    GHashFunc func;
    gboolean ret;
    ret = hash_func_from_gtype (gtype, &func);
    g_assert (ret != FALSE);
    return func;
}

GEqualFunc
_dbus_g_hash_equal_from_gtype (GType gtype)
{
    g_assert (_dbus_gtype_is_valid_hash_key (gtype));

    switch (gtype)
    {
        case G_TYPE_CHAR:
        case G_TYPE_UCHAR:
        case G_TYPE_BOOLEAN:
        case G_TYPE_INT:

```

```

case G_TYPE_UINT:
    return NULL;
case G_TYPE_STRING:
    return g_str_equal;
default:
    if (gtype == DBUS_TYPE_G_OBJECT_PATH)
        return g_str_equal;
    else if (gtype == DBUS_TYPE_G_SIGNATURE)
        return g_str_equal;
    g_assert_not_reached ();
    return NULL;
}
}

static void
hash_fake_simple_free_func (gpointer val)
{
    /* Havoc would be proud... :P */
    g_critical ("If you see this message then the author of this
application or "
"one of its libraries has tried to remove or replace the value
%p in a "
"hash table which was constructed by the D-Bus Glib
bindings.\n\n"

"However, it was not possible for the bindings to provide a
destroy "
"function to g_hash_table_new_full which is able to free this
value, as "
"its GType must be known in order to free it. This means the
memory "
"allocated to store the value has most likely just been
leaked.\n\n"

"To avoid this error, the GHashTable (or keys/values \"stolen\"
from "
"it) should be freed by using g_boxed_free as follows:\n"
"  g_boxed_free (dbus_g_type_get_map (\"GHashTable\", key_gtype,
"
"value_gtype), hash_table);\n", val);
}

GDestroyNotify
_dbus_g_hash_free_from_gtype (GType gtype)
{
    GDestroyNotify func;
    gboolean ret;

    ret = hash_simple_free_from_gtype (gtype, &func);

    /* if the value doesn't have a simple free function, we cannot
define a

```



```

    * meaningful free function here. instead, this hash table must be
    freed
    * using g_boxed_free so that the hash_free function gets invoked.
    if the
    * user does not do so, we provide a fake free function to provide a
    warning
    * in this case. */
    if (ret == FALSE)
    {
        g_assert (_dbus_gtype_is_valid_hash_value (gtype));

        func = hash_fake_simple_free_func;
    }

    return func;
}

```

```

static void gvalue_take_ptrarray_value (GValue *value, gpointer
instance);

```

```

static void
gvalue_take_hash_value (GValue *value, gpointer instance)
{
    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_CHAR:
            g_value_set_char (value, (gchar) GPOINTER_TO_INT (instance));
            break;
        case G_TYPE_UCHAR:
            g_value_set_uchar (value, (guchar) GPOINTER_TO_UINT (instance));
            break;
        case G_TYPE_BOOLEAN:
            g_value_set_boolean (value, (gboolean) GPOINTER_TO_UINT
(instance));
            break;
        case G_TYPE_INT:
            g_value_set_int (value, GPOINTER_TO_INT (instance));
            break;
        case G_TYPE_UINT:
            g_value_set_uint (value, GPOINTER_TO_UINT (instance));
            break;
        case G_TYPE_DOUBLE:
            g_value_set_double (value, *(gdouble *) instance);
            break;
        default:
            gvalue_take_ptrarray_value (value, instance);
            break;
    }
}

```

```

static gpointer ptrarray_value_from_gvalue (const GValue *value);

```

```

static gpointer
hash_value_from_gvalue (GValue *value)
{
    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_CHAR:
            return GINT_TO_POINTER ((int) g_value_get_char (value));
            break;
        case G_TYPE_UCHAR:
            return GUINT_TO_POINTER ((guint) g_value_get_uchar (value));
            break;
        case G_TYPE_BOOLEAN:
            return GUINT_TO_POINTER ((guint) g_value_get_boolean (value));
            break;
        case G_TYPE_INT:
            return GINT_TO_POINTER (g_value_get_int (value));
            break;
        case G_TYPE_UINT:
            return GUINT_TO_POINTER (g_value_get_uint (value));
            break;
        case G_TYPE_DOUBLE:
            {
                gdouble *p = (gdouble *) g_malloc0 (sizeof (gdouble));
                *p = g_value_get_double (value);
                return (gpointer) p;
            }
            break;
        default:
            return ptrarray_value_from_gvalue (value);
            break;
    }
}

```

```

struct DBusGHashTableValueForeachData
{
    DBusGTypeSpecializedMapIterator func;
    GType key_type;
    GType value_type;
    gpointer data;
};

```

```

static void
hashtable_foreach_with_values (gpointer key, gpointer value, gpointer
user_data)
{
    GValue key_val = {0, };
    GValue value_val = {0, };
    struct DBusGHashTableValueForeachData *data = user_data;

    g_value_init (&key_val, data->key_type);
    g_value_init (&value_val, data->value_type);
    gvalue_take_hash_value (&key_val, key);
}

```

```

    gvalue_take_hash_value (&value_val, value);

    data->func (&key_val, &value_val, data->data);
}

static void
hashtable_iterator (GType                                hash_type,
                   gpointer                               instance,
                   DBusGTypeSpecializedMapIterator      iterator,
                   gpointer                               user_data)
{
    struct DBusGHashTableValueForeachData data;
    GType key_gtype;
    GType value_gtype;

    key_gtype = dbus_g_type_get_map_key_specialization (hash_type);
    value_gtype = dbus_g_type_get_map_value_specialization (hash_type);

    data.func = iterator;
    data.key_type = key_gtype;
    data.value_type = value_gtype;
    data.data = user_data;

    g_hash_table_foreach (instance, hashtable_foreach_with_values,
&data);
}

void
_dbus_g_hash_table_insert_steal_values (GHashTable *table,
                                       GValue      *key_val,
                                       GValue      *value_val)
{
    gpointer key, val;

    key = hash_value_from_gvalue (key_val);
    val = hash_value_from_gvalue (value_val);

    g_hash_table_insert (table, key, val);
}

static void
hashtable_append (DBusGTypeSpecializedAppendContext *ctx,
                 GValue                               *key,
                 GValue                               *val)
{
    GHashTable *table;

    table = g_value_get_boxed (ctx->val);
    _dbus_g_hash_table_insert_steal_values (table, key, val);
}

```

```

static gpointer
hashtable_constructor (GType type)
{
    GHashTable *ret;
    GType key_gtype;
    GType value_gtype;

    key_gtype = dbus_g_type_get_map_key_specialization (type);
    value_gtype = dbus_g_type_get_map_value_specialization (type);

    ret = g_hash_table_new_full (_dbus_g_hash_func_from_gtype
(key_gtype),
                                _dbus_g_hash_equal_from_gtype (key_gtype),
                                _dbus_g_hash_free_from_gtype (key_gtype),
                                _dbus_g_hash_free_from_gtype (value_gtype));

    return ret;
}

static void
hashtable_insert_values (GHashTable *table,
                        const GValue *key_val,
                        const GValue *value_val)
{
    GValue key_copy = {0, };
    GValue value_copy = {0, };

    g_value_init (&key_copy, G_VALUE_TYPE (key_val));
    g_value_copy (key_val, &key_copy);
    g_value_init (&value_copy, G_VALUE_TYPE (value_val));
    g_value_copy (value_val, &value_copy);

    _dbus_g_hash_table_insert_steal_values (table, &key_copy,
&value_copy);
}

static void
hashtable_foreach_copy (const GValue *key, const GValue *val, gpointer
data)
{
    hashtable_insert_values ((GHashTable *) data, key, val);
}

static gpointer
hashtable_copy (GType type, gpointer src)
{
    GHashTable *ghash;
    GHashTable *ret;
    GValue hashval = {0,};

    ghash = src;

    ret = hashtable_constructor (type);

```

```

    g_value_init (&hashval, type);
    g_value_set_static_boxed (&hashval, ghash);
    dbus_g_type_map_value_iterate (&hashval, hashtable_foreach_copy,
ret);
    return ret;
}

/* we leave this here for backwards compatibility - any hash tables
nested
* inside hash tables will use this as their free function if users
were
* already relying on it, but dbus-glib itself will never use it
directly as
* hashtable_free is also defined. */
static void
hashtable_simple_free (gpointer val)
{
    g_hash_table_unref (val);
}

struct DBusGHashTableFreeData
{
    GType key_gtype;
    GType value_gtype;
};

static gboolean
hashtable_free_foreach_steal (gpointer key,
                              gpointer value,
                              gpointer user_data)
{
    struct DBusGHashTableFreeData *data = user_data;
    GValue val = { 0, };

    g_value_init (&val, data->key_gtype);
    gvalue_take_hash_value (&val, key);
    g_value_unset (&val);

    g_value_init (&val, data->value_gtype);
    gvalue_take_hash_value (&val, value);
    g_value_unset (&val);

    return TRUE;
}

static void
hashtable_free (GType type,
                gpointer val)
{
    struct DBusGHashTableFreeData data = { 0, };
    GHashTable *hash = val;

```

```

data.key_gtype = dbus_g_type_get_map_key_specialization (type);
data.value_gtype = dbus_g_type_get_map_value_specialization (type);

/* wheee, fun. two cases here. either:
 *
 * a) the keys and value types both have simple (ie, no * GType
parameter is
 * needed to know how to free them) free functions, in which case
they were
 * set as the hash free functions when the hash table was
constructed. in
 * this case, it's sufficient for us to unref the hash table as
before. we
 * have to keep doing this in order to maintain compatibility with
the ABI
 * which was around before hash tables could contain types which
don't have
 * simple free functions (such as GPtrArrays of other stuff). for
these
 * tables, users were able to ref the hash tables and add/remove
values, and
 * rely on meaningful free functions.
 *
 * b) for any other key or value types where /do/ need to know the
GType in
 * order to free it, this function is the only "right" way to free
the hash
 * table. both the key and value free functions were set to print a
big nasty
 * warning, and we free the contents of the hashtable with
foreach_steal.
 */
if (gtype_can_simple_free (data.key_gtype) &&
    gtype_can_simple_free (data.value_gtype))
    {
        g_hash_table_unref (hash);
    }
else
    {
        g_hash_table_foreach_steal (hash, hashtable_free_foreach_steal,
&data);
        g_hash_table_unref (hash);
    }
}

static gpointer
valuearray_constructor (GType type)
{
    GValueArray *ret;
    guint size = dbus_g_type_get_struct_size (type);
    guint i;

```

```

ret = g_value_array_new (size);
for (i=0; i < size; i++)
{
    GValue val = {0,};
    g_value_init (&val, dbus_g_type_get_struct_member_type (type,
i));
    g_value_array_append(ret, &val);
}
return (gpointer)ret;
}

```

```

static gpointer
valuearray_copy (GType type, gpointer src)
{
    return g_value_array_copy ((GValueArray*) src);
}

```

```

static void
valuearray_simple_free (gpointer val)
{
    g_value_array_free (val);
}

```

```

static gboolean
valuearray_get_member (GType type, gpointer instance,
                      guint member, GValue *ret)
{
    GValueArray *va = (GValueArray*) instance;
    const GValue *val;
    if (member < dbus_g_type_get_struct_size (type))
    {
        val = g_value_array_get_nth (va, member);
        g_value_copy (val, ret);
        return TRUE;
    }
    else
        return FALSE;
}

```

```

static gboolean
valuearray_set_member (GType type, gpointer instance,
                      guint member, const GValue *member_type)
{
    GValueArray *va = (GValueArray*) instance;
    GValue *vp;
    if (member < dbus_g_type_get_struct_size (type))
    {
        vp = g_value_array_get_nth (va, member);
        g_value_copy (member_type, vp);
        return TRUE;
    }
    else

```

```

        return FALSE;
    }

static gpointer
array_constructor (GType type)
{
    GArray *array;
    guint elt_size;
    GType elt_type;
    gboolean zero_terminated;
    gboolean clear;

    elt_type = dbus_g_type_get_collection_specialization (type);
    g_assert (elt_type != G_TYPE_INVALID);

    elt_size = _dbus_g_type_fixed_get_size (elt_type);

    /* These are "safe" defaults */
    zero_terminated = TRUE; /* ((struct _DBusGRealArray*) garray)-
>zero_terminated; */
    clear = TRUE; /* ((struct _DBusGRealArray*) garray)->clear; */

    array = g_array_new (zero_terminated, clear, elt_size);
    return array;
}

static void
array_iterator (GType garray_type,
                gpointer instance,
                DBusGTypeSpecializedCollectionIterator iterator,
                gpointer user_data)
{
    GArray *array;
    GType elt_gtype;
    guint i;

    array = instance;

    elt_gtype = dbus_g_type_get_collection_specialization (garray_type);

    for (i = 0; i < array->len; i++)
    {
        GValue val = {0, };
        g_value_init (&val, elt_gtype);

        switch (elt_gtype)
        {
            case G_TYPE_BOOLEAN:
                g_value_set_boolean (&val, !!g_array_index (array,
gboolean, i));
                break;

```



```

        case G_TYPE_FLOAT:
            g_value_set_float (&val, g_array_index (array, gfloat,
i));
            break;

        case G_TYPE_DOUBLE:
            g_value_set_double (&val, g_array_index (array, gdouble,
i));
            break;

        case G_TYPE_CHAR:
            g_value_set_char (&val, g_array_index (array, gchar, i));
            break;

        case G_TYPE_UCHAR:
            g_value_set_uchar (&val, g_array_index (array, guchar,
i));
            break;

        case G_TYPE_INT:
            g_value_set_int (&val, g_array_index (array, gint, i));
            break;

        case G_TYPE_UINT:
            g_value_set_uint (&val, g_array_index (array, guint, i));
            break;

        case G_TYPE_LONG:
            g_value_set_long (&val, g_array_index (array, glong, i));
            break;

        case G_TYPE_ULONG:
            g_value_set_ulong (&val, g_array_index (array, gulong,
i));
            break;

        case G_TYPE_INT64:
            g_value_set_int64 (&val, g_array_index (array, gint64,
i));
            break;

        case G_TYPE_UINT64:
            g_value_set_uint64 (&val, g_array_index (array, guint64,
i));
            break;

        default:
            g_assert_not_reached ();
    }

    iterator (&val, user_data);

```

```

    }
}

static void
array_append (DBusGTypeSpecializedAppendContext *ctx,
             GValue *value)
{
    GArray *array = g_value_get_boxed (ctx->val);
    GType elt_gtype;
    union {
        guint64 u64;
        gint64 i64;
        gulong ul;
        glong l;
        guint u;
        gint i;
        guchar uc;
        gchar c;
        gboolean b;
        gfloat f;
        gdouble d;
    } tmp;

    elt_gtype = dbus_g_type_get_collection_specialization (
        G_VALUE_TYPE (ctx->val));

    switch (elt_gtype)
    {
        case G_TYPE_BOOLEAN:
            tmp.b = g_value_get_boolean (value);
            g_array_append_val (array, tmp.b);
            break;

        case G_TYPE_FLOAT:
            tmp.f = g_value_get_float (value);
            g_array_append_val (array, tmp.f);
            break;

        case G_TYPE_DOUBLE:
            tmp.d = g_value_get_double (value);
            g_array_append_val (array, tmp.d);
            break;

        case G_TYPE_CHAR:
            tmp.c = g_value_get_char (value);
            g_array_append_val (array, tmp.c);
            break;

        case G_TYPE_UCHAR:
            tmp.uc = g_value_get_uchar (value);
            g_array_append_val (array, tmp.uc);
            break;
    }
}

```

```

    case G_TYPE_INT:
        tmp.i = g_value_get_int (value);
        g_array_append_val (array, tmp.i);
        break;

    case G_TYPE_UINT:
        tmp.u = g_value_get_uint (value);
        g_array_append_val (array, tmp.u);
        break;

    case G_TYPE_LONG:
        tmp.l = g_value_get_long (value);
        g_array_append_val (array, tmp.l);
        break;

    case G_TYPE_ULONG:
        tmp.ul = g_value_get_ulong (value);
        g_array_append_val (array, tmp.ul);
        break;

    case G_TYPE_INT64:
        tmp.i64 = g_value_get_int64 (value);
        g_array_append_val (array, tmp.i64);
        break;

    case G_TYPE_UINT64:
        tmp.u64 = g_value_get_uint64 (value);
        g_array_append_val (array, tmp.u64);
        break;

    default:
        g_assert_not_reached ();
}
}

static gpointer
array_copy (GType type, gpointer src)
{
    GArray *garray;
    GArray *new;

    garray = src;

    new = array_constructor (type);
    g_array_append_vals (new, garray->data, garray->len);

    return new;
}

static void
array_simple_free (gpointer val)

```

```

{
    GArray *array;
    array = val;
    g_array_free (array, TRUE);
}

static gboolean
array_fixed_accessor (GType type, gpointer instance, gpointer *values,
guint *len)
{
    GType elt_type;
    GArray *array = instance;

    elt_type = dbus_g_type_get_collection_specialization (type);
    if (!_dbus_g_type_is_fixed (elt_type))
        return FALSE;

    *values = array->data;
    *len = array->len;
    return TRUE;
}

static gpointer
ptrarray_constructor (GType type)
{
    /* Later we should determine a destructor, need g_ptr_array_destroy
    */
    return g_ptr_array_new ();
}

static void
gvalue_take_ptrarray_value (GValue *value, gpointer instance)
{
    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_STRING:
            g_value_take_string (value, instance);
            break;
        case G_TYPE_BOXED:
            g_value_take_boxed (value, instance);
            break;
        case G_TYPE_OBJECT:
            g_value_take_object (value, instance);
            break;
        default:
            g_assert_not_reached ();
            break;
    }
}

static gpointer
ptrarray_value_from_gvalue (const GValue *value)

```

```

{
    GValue tmp = {0, };

    /* if the NOCOPY flag is set, then value was created via set_static
and hence
    * is not owned by us. in order to preserve the "take" semantics
that the API
    * has in general (which avoids copying in the common case), we must
copy any
    * static values so that we can indiscriminately free the entire
collection
    * later. */
    if (value->data[1].v_uint & G_VALUE_NOCOPY_CONTENTS)
    {
        g_value_init (&tmp, G_VALUE_TYPE (value));
        g_value_copy (value, &tmp);
        value = &tmp;
    }

    switch (g_type_fundamental (G_VALUE_TYPE (value)))
    {
        case G_TYPE_STRING:
            return (gpointer) g_value_get_string (value);
            break;
        case G_TYPE_BOXED:
            return g_value_get_boxed (value);
            break;
        case G_TYPE_OBJECT:
            return g_value_get_object (value);
            break;
        default:
            g_assert_not_reached ();
            return NULL;
    }
}

static void
ptrarray_iterator (GType
ptrarray_type,
                  gpointer          instance,
                  DBusGTypeSpecializedCollectionIterator iterator,
                  gpointer          user_data)
{
    GPtrArray *ptrarray;
    GType elt_gtype;
    guint i;

    ptrarray = instance;

    elt_gtype = dbus_g_type_get_collection_specialization
(ptrarray_type);

```

```

    for (i = 0; i < ptrarray->len; i++)
    {
        GValue val = {0, };
        g_value_init (&val, elt_gtype);
        gvalue_take_ptrarray_value (&val, g_ptr_array_index (ptrarray,
i));
        iterator (&val, user_data);
    }
}

static void
ptrarray_copy_elt (const GValue *val, gpointer user_data)
{
    GPtrArray *dest = user_data;
    GValue val_copy = {0, };

    g_value_init (&val_copy, G_VALUE_TYPE (val));
    g_value_copy (val, &val_copy);

    g_ptr_array_add (dest, ptrarray_value_from_gvalue (&val_copy));
}

static gpointer
ptrarray_copy (GType type, gpointer src)
{
    GPtrArray *new;
    GValue array_val = {0, };

    g_value_init (&array_val, type);
    g_value_set_static_boxed (&array_val, src);

    new = ptrarray_constructor (type);
    dbus_g_type_collection_value_iterate (&array_val, ptrarray_copy_elt,
new);

    return new;
}

static void
ptrarray_append (DBusGTypeSpecializedAppendContext *ctx, GValue
*value)
{
    GPtrArray *array;

    array = g_value_get_boxed (ctx->val);

    g_ptr_array_add (array, ptrarray_value_from_gvalue (value));
}

static void
ptrarray_free (GType type, gpointer val)
{

```

```

GPtrArray *array;
GValue elt_val = {0, };
GType elt_gtype;
unsigned int i;

array = val;

elt_gtype = dbus_g_type_get_collection_specialization (type);

for (i = 0; i < array->len; i++)
{
    g_value_init (&elt_val, elt_gtype);
    gvalue_take_ptrarray_value (&elt_val, g_ptr_array_index (array,
i));
    g_value_unset (&elt_val);
}

g_ptr_array_free (array, TRUE);
}

static gpointer
slist_constructor (GType type)
{
    return NULL;
}

static void
slist_iterator (GType                list_type,
                gpointer             instance,
                DBusGTypeSpecializedCollectionIterator iterator,
                gpointer             user_data)
{
    GSList *slist;
    GType elt_gtype;

    slist = instance;

    elt_gtype = dbus_g_type_get_collection_specialization (list_type);

    for (slist = instance; slist != NULL; slist = slist->next)
    {
        GValue val = {0, };
        g_value_init (&val, elt_gtype);
        gvalue_take_ptrarray_value (&val, slist->data);
        iterator (&val, user_data);
    }
}

static void
slist_copy_elt (const GValue *val, gpointer user_data)
{
    GSList **dest = user_data;

```

```

GValue val_copy = {0, };

g_value_init (&val_copy, G_VALUE_TYPE (val));
g_value_copy (val, &val_copy);

*dest = g_slist_append (*dest, ptrarray_value_from_gvalue
(&val_copy));
}

static gpointer
slist_copy (GType type, gpointer src)
{
    GSList *new;
    GValue slist_val = {0, };

    g_value_init (&slist_val, type);
    g_value_set_static_boxed (&slist_val, src);

    new = slist_constructor (type);
    dbus_g_type_collection_value_iterate (&slist_val, slist_copy_elt,
&new);

    return new;
}

static void
slist_append (DBusGTypeSpecializedAppendContext *ctx, GValue *value)
{
    GSList *list;

    list = g_value_get_boxed (ctx->val);
    list = g_slist_prepend (list, ptrarray_value_from_gvalue (value));
    g_value_set_static_boxed (ctx->val, list);
}

static void
slist_end_append (DBusGTypeSpecializedAppendContext *ctx)
{
    GSList *list;

    /* if you append multiple times to the slist, this will reverse the
existing
* elements... we need an init_append function */
    list = g_value_get_boxed (ctx->val);
    list = g_slist_reverse (list);

    g_value_take_boxed (ctx->val, list);
}

static void
slist_free (GType type, gpointer val)
{

```



```

GSList *list;
GType elt_gtype;
list = val;

elt_gtype = dbus_g_type_get_collection_specialization (type);

while (list != NULL)
{
    GValue elt_val = {0, };
    g_value_init (&elt_val, elt_gtype);
    gvalue_take_ptrarray_value (&elt_val, list->data);
    g_value_unset (&elt_val);
    list = g_slist_next(list);
}
list=val;
g_slist_free (list);
}

void
_dbus_g_type_specialized_builtins_init (void)
{
    /* types with a simple_free function can be freed at run-time
without
* the destroy function needing to know the type, so they can be
* stored in hash tables */

    static const DBusGTypeSpecializedCollectionVtable array_vtable = {
        {
            array_constructor,
            NULL,
            array_copy,
            array_simple_free,
            NULL,
            NULL,
        },
        array_fixed_accessor,
        array_iterator,
        array_append,
        NULL
    };

    static const DBusGTypeSpecializedCollectionVtable ptrarray_vtable =
{
    {
        ptrarray_constructor,
        ptrarray_free,
        ptrarray_copy,
        NULL,
        NULL,
        NULL,
    },
}

```

```

    NULL,
    ptrarray_iterator,
    ptrarray_append,
    NULL,
};

static const DBusGTypeSpecializedCollectionVtable slist_vtable = {
    {
        slist_constructor,
        slist_free,
        slist_copy,
        NULL,
        NULL,
        NULL,
    },
    NULL,
    slist_iterator,
    slist_append,
    slist_end_append,
};

static const DBusGTypeSpecializedMapVtable hashtable_vtable = {
    {
        hashtable_constructor,
        hashtable_free,
        hashtable_copy,
        hashtable_simple_free,
        NULL,
        NULL
    },
    hashtable_iterator,
    hashtable_append
};

static const DBusGTypeSpecializedStructVtable valuearray_vtable = {
    {
        valuearray_constructor,
        NULL,
        valuearray_copy,
        valuearray_simple_free,
        NULL,
        NULL
    },
    valuearray_get_member,
    valuearray_set_member
};

_dbus_g_type_register_collection ("GSList", &slist_vtable, 0);
_dbus_g_type_register_collection ("GArray", &array_vtable, 0);
_dbus_g_type_register_collection ("GPtrArray", &ptrarray_vtable, 0);
_dbus_g_type_register_map ("GHashTable", &hashtable_vtable, 0);

```

```

    _dbus_g_type_register_struct ("GValueArray", &valuearray_vtable, 0);
}

#ifdef DBUS_BUILD_TESTS

typedef struct
{
    gboolean seen_foo;
    gboolean seen_baz;
} TestSpecializedHashData;

static void
test_specialized_hash (const GValue *key, const GValue *val, gpointer
user_data)
{
    TestSpecializedHashData *data = user_data;

    g_assert (G_VALUE_HOLDS_STRING (key));
    g_assert (G_VALUE_HOLDS_STRING (val));

    if (!strcmp (g_value_get_string (key), "foo"))
    {
        data->seen_foo = TRUE;
        g_assert (!strcmp (g_value_get_string (val), "bar"));
    }
    else if (!strcmp (g_value_get_string (key), "baz"))
    {
        data->seen_baz = TRUE;
        g_assert (!strcmp (g_value_get_string (val), "moo"));
    }
    else
    {
        g_assert_not_reached ();
    }
}

static void
test_specialized_hash_2 (const GValue *key, const GValue *val,
gpointer user_data)
{
    TestSpecializedHashData *data = user_data;
    const GValue *realval;

    g_assert (G_VALUE_HOLDS_STRING (key));
    g_assert (G_VALUE_TYPE (val) == G_TYPE_VALUE);

    realval = g_value_get_boxed (val);

    if (!strcmp (g_value_get_string (key), "foo"))
    {
        data->seen_foo = TRUE;
        g_assert (G_VALUE_HOLDS_UINT (realval));
    }
}

```

```

        g_assert (g_value_get_uint (realval) == 20);
    }
else if (!strcmp (g_value_get_string (key), "baz"))
    {
        data->seen_baz = TRUE;
        g_assert (G_VALUE_HOLDS_STRING (realval));
        g_assert (!strcmp ("bar", g_value_get_string (realval)));
    }
else
    {
        g_assert_not_reached ();
    }
}

gboolean
_dbus_gvalue_utils_test (const char *datadir)
{
    GType type;

    dbus_g_type_specialized_init ();
    _dbus_g_type_specialized_builtins_init ();

    type = dbus_g_type_get_collection ("GArray", G_TYPE_UINT);
    g_assert (dbus_g_type_is_collection (type));
    g_assert (dbus_g_type_get_collection_specialization (type) ==
G_TYPE_UINT);
    {
        GArray *instance;

        instance = dbus_g_type_specialized_construct (type);

        g_assert (instance->len == 0);

        g_array_free (instance, TRUE);
    }

    type = dbus_g_type_get_map ("GHashTable", G_TYPE_STRING,
G_TYPE_STRING);
    g_assert (dbus_g_type_is_map (type));
    g_assert (dbus_g_type_get_map_key_specialization (type) ==
G_TYPE_STRING);
    g_assert (dbus_g_type_get_map_value_specialization (type) ==
G_TYPE_STRING);
    {
        GHashTable *instance;
        GValue val = { 0, };
        TestSpecializedHashData hashdata;

        instance = dbus_g_type_specialized_construct (type);

        g_assert (g_hash_table_size (instance) == 0);
    }
}

```

```

    g_hash_table_insert (instance, g_strdup ("foo"), g_strdup
("bar"));
    g_hash_table_insert (instance, g_strdup ("baz"), g_strdup
("moo"));
    g_assert (g_hash_table_size (instance) == 2);

    g_value_init (&val, type);
    g_value_take_boxed (&val, instance);
    hashdata.seen_foo = FALSE;
    hashdata.seen_baz = FALSE;
    dbus_g_type_map_value_iterate (&val,
        test_specialized_hash,
        &hashdata);

    g_assert (hashdata.seen_foo);
    g_assert (hashdata.seen_baz);

    g_value_unset (&val);
}

type = dbus_g_type_get_map ("GHashTable", G_TYPE_STRING,
G_TYPE_VALUE);
g_assert (dbus_g_type_is_map (type));
g_assert (dbus_g_type_get_map_key_specialization (type) ==
G_TYPE_STRING);
g_assert (dbus_g_type_get_map_value_specialization (type) ==
G_TYPE_VALUE);
{
    GHashTable *instance;
    GValue val = { 0, };
    TestSpecializedHashData hashdata;
    DBusGTypeSpecializedAppendContext ctx;
    GValue *eltval;

    instance = dbus_g_type_specialized_construct (type);
    g_value_init (&val, type);
    g_value_take_boxed (&val, instance);

    dbus_g_type_specialized_init_append (&val, &ctx);

    {
        GValue keyval = { 0, };
        GValue valval = { 0, };
        g_value_init (&keyval, G_TYPE_STRING);
        g_value_set_string (&keyval, "foo");

        g_value_init (&valval, G_TYPE_VALUE);
        eltval = g_new0 (GValue, 1);
        g_value_init (eltval, G_TYPE_UINT);
        g_value_set_uint (eltval, 20);
        g_value_take_boxed (&valval, eltval);
        dbus_g_type_specialized_map_append (&ctx, &keyval, &valval);
    }
}

```

```

}

{
    GValue keyval = { 0, };
    GValue valval = { 0, };
    g_value_init (&keyval, G_TYPE_STRING);
    g_value_set_string (&keyval, "baz");
    g_value_init (&valval, G_TYPE_VALUE);
    eltval = g_new0 (GValue, 1);
    g_value_init (eltval, G_TYPE_STRING);
    g_value_set_string (eltval, "bar");
    g_value_take_boxed (&valval, eltval);
    dbus_g_type_specialized_map_append (&ctx, &keyval, &valval);
}

hashdata.seen_foo = FALSE;
hashdata.seen_baz = FALSE;
dbus_g_type_map_value_iterate (&val,
                               test_specialized_hash_2,
                               &hashdata);

g_assert (hashdata.seen_foo);
g_assert (hashdata.seen_baz);

g_value_unset (&val);
}

type = dbus_g_type_get_collection ("GSList", G_TYPE_OBJECT);
g_assert (dbus_g_type_is_collection (type));
g_assert (dbus_g_type_get_collection_specialization (type) ==
G_TYPE_OBJECT);
{
    GSList *instance, *tmp, *copy;
    GValue val = {0, };
    GValue copyval = {0, };
    DBusGTypeSpecializedAppendContext ctx;
    GObject *objects[3];
    int i;

    instance = dbus_g_type_specialized_construct (type);
    g_assert (instance == NULL);

    g_value_init (&val, type);
    g_value_take_boxed (&val, instance);

    dbus_g_type_specialized_init_append (&val, &ctx);

    for (i = 0; i < 3; i++)
        {
            GValue eltval = { 0, };
            GObject *obj = g_object_new (G_TYPE_OBJECT, NULL);

```

```

    g_assert (obj != NULL);
    objects[i] = obj;
    g_object_add_weak_pointer (obj, (gpointer) (objects + i));

    g_value_init (&eltval, G_TYPE_OBJECT);
    g_value_take_object (&eltval, obj);
    dbus_g_type_specialized_collection_append (&ctx, &eltval);
}

dbus_g_type_specialized_collection_end_append (&ctx);

instance = g_value_get_boxed (&val);
g_assert (g_slist_length (instance) == 3);

for (tmp = instance; tmp; tmp = tmp->next)
{
    GObject *obj = tmp->data;
    g_assert (G_IS_OBJECT (obj));
    g_assert (obj->ref_count == 1);
}

g_value_init (&copyval, type);
g_value_copy (&val, &copyval);

copy = g_value_get_boxed (&copyval);
g_assert (g_slist_length (copy) == 3);

for (tmp = copy; tmp; tmp = tmp->next)
{
    GObject *obj = tmp->data;
    g_assert (G_IS_OBJECT (obj));
    g_assert (obj->ref_count == 2);
}

g_value_unset (&copyval);

for (i = 0; i < 3; i++)
{
    g_assert (objects[i] != NULL);
}

for (tmp = instance; tmp; tmp = tmp->next)
{
    GObject *obj = tmp->data;
    g_assert (G_IS_OBJECT (obj));
    g_assert (obj->ref_count == 1);
}

g_value_unset (&val);

for (i = 0; i < 3; i++)
{

```

```

        g_assert (objects[i] == NULL);
    }
}

type = dbus_g_type_get_collection ("GPtrArray", G_TYPE_STRING);
g_assert (dbus_g_type_is_collection (type));
g_assert (dbus_g_type_get_collection_specialization (type) ==
G_TYPE_STRING);
{
    GPtrArray *instance;
    DBusGTypeSpecializedAppendContext ctx;
    GValue val = {0, };
    GValue eltval = {0, };

    instance = dbus_g_type_specialized_construct (type);

    g_assert (instance->len == 0);

    g_value_init (&val, type);
    g_value_take_boxed (&val, instance);

    dbus_g_type_specialized_init_append (&val, &ctx);

    g_value_init (&eltval, G_TYPE_STRING);
    g_value_set_static_string (&eltval, "foo");
    dbus_g_type_specialized_collection_append (&ctx, &eltval);

    g_value_reset (&eltval);
    g_value_set_static_string (&eltval, "bar");
    dbus_g_type_specialized_collection_append (&ctx, &eltval);

    g_value_reset (&eltval);
    g_value_set_static_string (&eltval, "baz");
    dbus_g_type_specialized_collection_append (&ctx, &eltval);

    dbus_g_type_specialized_collection_end_append (&ctx);

    g_assert (instance->len == 3);

    g_assert (!strcmp ("foo", g_ptr_array_index (instance, 0)));
    g_assert (!strcmp ("bar", g_ptr_array_index (instance, 1)));
    g_assert (!strcmp ("baz", g_ptr_array_index (instance, 2)));

    g_value_unset (&val);
}

type = dbus_g_type_get_struct ("GValueArray", G_TYPE_STRING,
G_TYPE_UINT, DBUS_TYPE_G_OBJECT_PATH, G_TYPE_INVALID);
g_assert (dbus_g_type_is_struct (type));
g_assert (dbus_g_type_get_struct_size (type) == 3);
g_assert (dbus_g_type_get_struct_member_type (type, 0) ==
G_TYPE_STRING);

```



```

    g_assert (dbus_g_type_get_struct_member_type (type, 1) ==
G_TYPE_UINT);
    g_assert (dbus_g_type_get_struct_member_type (type, 2) ==
DBUS_TYPE_G_OBJECT_PATH);
    {
        GValueArray *instance;
        GValue val = {0, };
        GValue memval = {0, };

        instance = dbus_g_type_specialized_construct (type);

        g_assert (instance->n_values == 3);

        g_value_init (&val, type);
        g_value_take_boxed (&val, instance);

        g_value_init (&memval, G_TYPE_STRING);
        g_value_set_static_string (&memval, "foo");
        dbus_g_type_struct_set_member (&val, 0, &memval);
        g_value_unset (&memval);

        g_value_init (&memval, G_TYPE_UINT);
        g_value_set_uint (&memval, 42);
        dbus_g_type_struct_set_member (&val, 1, &memval);
        g_value_unset (&memval);

        g_value_init (&memval, DBUS_TYPE_G_OBJECT_PATH);
        g_value_set_static_boxed (&memval, "/bar/moo/foo/baz");
        dbus_g_type_struct_set_member (&val, 2, &memval);
        g_value_unset (&memval);

        g_assert (instance->n_values == 3);

        g_value_init (&memval, G_TYPE_STRING);
        dbus_g_type_struct_get_member (&val, 0, &memval);
        g_assert (0 == strcmp (g_value_get_string (&memval), "foo"));
        g_value_unset (&memval);

        g_value_init (&memval, G_TYPE_UINT);
        dbus_g_type_struct_get_member (&val, 1, &memval);
        g_assert (g_value_get_uint (&memval) == 42);
        g_value_unset (&memval);

        g_value_init (&memval, DBUS_TYPE_G_OBJECT_PATH);
        dbus_g_type_struct_get_member (&val, 2, &memval);
        g_assert (0 == strcmp ((gchar*) g_value_get_boxed (&memval),
"/bar/moo/foo/baz"));
        g_value_unset (&memval);

        g_value_unset (&val);
    }
}

```

```

    type = dbus_g_type_get_struct ("GValueArray", G_TYPE_STRING,
G_TYPE_UINT, DBUS_TYPE_G_OBJECT_PATH, G_TYPE_INVALID);
    g_assert (dbus_g_type_is_struct (type));
    g_assert (dbus_g_type_get_struct_size (type) == 3);
    g_assert (dbus_g_type_get_struct_member_type (type, 0) ==
G_TYPE_STRING);
    g_assert (dbus_g_type_get_struct_member_type (type, 1) ==
G_TYPE_UINT);
    g_assert (dbus_g_type_get_struct_member_type (type, 2) ==
DBUS_TYPE_G_OBJECT_PATH);
    {
        GValueArray *instance;
        GValue val = {0, };

        instance = dbus_g_type_specialized_construct (type);

        g_assert (instance->n_values == 3);

        g_value_init (&val, type);
        g_value_take_boxed (&val, instance);

        dbus_g_type_struct_set (&val,
                                0, "foo",
                                1, 42,
                                2, "/bar/moo/foo/baz",
                                G_MAXUINT);

        g_assert (instance->n_values == 3);

        {
            gchar *string;
            guint intval;
            gchar *path;

            dbus_g_type_struct_get (&val,
                                    0, &string,
                                    1, &intval,
                                    2, &path,
                                    G_MAXUINT);

            g_assert (0 == strcmp (string, "foo"));
            g_assert (intval == 42);
            g_assert (0 == strcmp (path, "/bar/moo/foo/baz"));
        }

        g_value_unset (&val);
    }

    return TRUE;
}

```



```

typedef void (*DBusGHashValueForeachFunc) (GValue * key, GValue *val,
gpointer data);

void          _dbus_g_hash_table_value_foreach      (GHashTable
*table,
                                                    GType
hash_type,
                                                    DBusGHashValueForeachFunc  func,
                                                    gpointer
data);

void          _dbus_g_hash_table_insert_values     (GHashTable
*table,
                                                    GValue
*key_val,
                                                    GValue
*value_val);
void          _dbus_g_hash_table_insert_steal_values (GHashTable
*table,
                                                    GValue  *key_val,
                                                    GValue  *value_val);

gboolean      _dbus_gtype_is_valid_hash_key      (GType type);
gboolean      _dbus_gtype_is_valid_hash_value    (GType type);

GHashFunc     _dbus_g_hash_func_from_gtype      (GType gtype);
GEqualFunc    _dbus_g_hash_equal_from_gtype     (GType gtype);
GDestroyNotify _dbus_g_hash_free_from_gtype     (GType gtype);

gboolean      _dbus_gvalue_store                 (GValue
*value,
                                                    gpointer  storage);

gboolean      _dbus_gvalue_take                 (GValue
*value,
                                                    GTypeCValue  *cvalue);

gboolean      _dbus_gtype_can_signal_error      (GType
gtype);
gboolean      _dbus_gvalue_signals_error       (const GValue
*value);

G_END_DECLS

#endif

```

File = dbus-gvalue.c

```
/* -*- mode: C; c-file-style: "gnu" -*- */
```



```

typedef gboolean (*DBusGValueMarshalFunc)      (DBusMessageIter
*iter,
                                         const GValue      *value);
typedef gboolean (*DBusGValueDemarshalFunc)    (DBusGValueMarshalCtx
*context,
                                         DBusMessageIter  *iter,
                                         GValue           *value,
                                         GError          **error);

typedef struct {
    DBusGValueMarshalFunc    marshaller;
    DBusGValueDemarshalFunc demarshaller;
} DBusGTypeMarshalVtable;

typedef struct {
    const char                *sig;
    const DBusGTypeMarshalVtable *vtable;
} DBusGTypeMarshalData;

static GQuark
dbus_g_type_metadata_data_quark ()
{
    static GQuark quark;
    if (!quark)
        quark = g_quark_from_static_string ("DBusGTypeMetaData");

    return quark;
}

static void
set_type_metadata (GType type, const DBusGTypeMarshalData *data)
{
    g_type_set_qdata (type, dbus_g_type_metadata_data_quark (),
(gpoiner) data);
}

static void
register_basic (int typecode, const DBusGTypeMarshalData *typedata)
{
    set_type_metadata (_dbus_gtype_from_basic_typecode (typecode),
typedata);
}

void
_dbus_g_value_types_init (void)
{
    static gboolean types_initialized;

    static const DBusGTypeMarshalVtable basic_vtable = {
        marshal_basic,
        demarshal_basic
    };
};

```



```

if (types_initialized)
    return;

dbus_g_type_specialized_init ();

/* Register basic types */
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_BOOLEAN_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_BOOLEAN, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_BYTE_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_BYTE, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_INT16_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_INT16, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_UINT16_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_UINT16, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_UINT32_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_UINT32, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_INT32_AS_STRING,
        &basic_vtable,
    };
    register_basic (DBUS_TYPE_INT32, &typedata);
}
{
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_UINT64_AS_STRING,

```

```

    &basic_vtable,
};
register_basic (DBUS_TYPE_UINT64, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_INT64_AS_STRING,
    &basic_vtable,
};
register_basic (DBUS_TYPE_INT64, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_DOUBLE_AS_STRING,
    &basic_vtable,
};
register_basic (DBUS_TYPE_DOUBLE, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_STRING_AS_STRING,
    &basic_vtable,
};
register_basic (DBUS_TYPE_STRING, &typedata);
}
/* fundamental GTypes that don't map 1:1 with D-BUS types */
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_BYTE_AS_STRING,
    &basic_vtable,
};
set_type_metadata (G_TYPE_CHAR, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_INT32_AS_STRING,
    &basic_vtable,
};
set_type_metadata (G_TYPE_LONG, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_UINT32_AS_STRING,
    &basic_vtable,
};
set_type_metadata (G_TYPE_ULONG, &typedata);
}
{
static const DBusGTypeMarshalData typedata = {
    DBUS_TYPE_DOUBLE_AS_STRING,
    &basic_vtable,
};
};

```

```

    set_type_metadata (G_TYPE_FLOAT, &typedata);
}

/* Register complex types with builtin GType mappings */
{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_variant,
        demarshal_variant
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_VARIANT_AS_STRING,
        &vtable
    };
    set_type_metadata (G_TYPE_VALUE, &typedata);
};
{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_strv,
        demarshal_strv
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_ARRAY_AS_STRING DBUS_TYPE_STRING_AS_STRING,
        &vtable
    };
    set_type_metadata (G_TYPE_STRV, &typedata);
};

/* Register some types specific to the D-BUS GLib bindings */
{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_proxy,
        demarshal_proxy
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_OBJECT_PATH_AS_STRING,
        &vtable
    };
    set_type_metadata (DBUS_TYPE_G_PROXY, &typedata);
}

{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_object_path,
        demarshal_object_path
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_OBJECT_PATH_AS_STRING,
        &vtable
    };
    set_type_metadata (DBUS_TYPE_G_OBJECT_PATH, &typedata);
}

```

```

{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_object,
        demarshal_object
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_OBJECT_PATH_AS_STRING,
        &vtable
    };
    set_type_metadata (G_TYPE_OBJECT, &typedata);
}

{
    static const DBusGTypeMarshalVtable vtable = {
        marshal_signature,
        demarshal_signature
    };
    static const DBusGTypeMarshalData typedata = {
        DBUS_TYPE_SIGNATURE_AS_STRING,
        &vtable
    };
    set_type_metadata (DBUS_TYPE_G_SIGNATURE, &typedata);
}

types_initialized = TRUE;
}

/**
 * DBusGObjectPath:
 *
 * A typedef for a string used to represent D-Bus object paths.
 * Its GType is %DBUS_TYPE_G_OBJECT_PATH, derived from %G_TYPE_BOXED.
 *
 * Prior to version 0.FIXME this was used as the type name of
 * %DBUS_TYPE_G_OBJECT_PATH, but did not actually exist as a typedef.
 *
 * Since: 0.FIXME
 */

/**
 * DBUS_TYPE_G_OBJECT_PATH:
 *
 * The #GType of a #DBusGObjectPath, which is a boxed type containing
 a
 * D-Bus object path as a zero-terminated string. Object paths can be
 * copied with g_strdup() and freed with g_free(), just like
 %G_TYPE_STRING,
 * but have a distinct boxed type to allow them to be distinguished
 when
 * stored in a #GValue.
 */

```

```

* Returns: a type derived from %G_TYPE_BOXED
*/
GType
dbus_g_object_path_get_g_type (void)
{
    static GType type_id = 0;

    if (!type_id)
        type_id = g_boxed_type_register_static ("DBusGObjectPath",
                                                (GBoxedCopyFunc) g_strdup,
                                                (GBoxedFreeFunc) g_free);

    return type_id;
}

/**
 * DBusGSignature:
 *
 * A typedef for a string used to represent D-Bus signatures.
 * Its GType is %DBUS_TYPE_G_SIGNATURE, derived from %G_TYPE_BOXED.
 *
 * Prior to version 0.FIXME this was used as the type name of
 * %DBUS_TYPE_G_SIGNATURE, but did not actually exist as a typedef.
 *
 * Since: 0.FIXME
 */

/**
 * DBUS_TYPE_G_SIGNATURE:
 *
 * The #GType of a #DBusGSignature, which is a boxed type containing a
 * D-Bus signature as a zero-terminated string. Signatures can be
 * copied with g_strdup() and freed with g_free(), just like
 * %G_TYPE_STRING,
 * but have a distinct boxed type to allow them to be distinguished
 * when
 * stored in a #GValue.
 *
 * Returns: a type derived from %G_TYPE_BOXED
 */
GType
dbus_g_signature_get_g_type (void)
{
    static GType type_id = 0;

    if (G_UNLIKELY (type_id == 0))
        type_id = g_boxed_type_register_static ("DBusGSignature",
                                                (GBoxedCopyFunc) g_strdup,
                                                (GBoxedFreeFunc) g_free);

    return type_id;
}

```

```

char *
_dbus_gtype_to_signature (GType gtype)
{
    char *ret;
    DBusGTypeMarshalData *typedata;

    if (dbus_g_type_is_collection (gtype))
    {
        GType elt_gtype;
        char *subsig;

        elt_gtype = dbus_g_type_get_collection_specialization (gtype);
        subsig = _dbus_gtype_to_signature (elt_gtype);
        ret = g_strconcat (DBUS_TYPE_ARRAY_AS_STRING, subsig, NULL);
        g_free (subsig);
    }
    else if (dbus_g_type_is_map (gtype))
    {
        GType key_gtype;
        GType val_gtype;
        char *key_subsig;
        char *val_subsig;

        key_gtype = dbus_g_type_get_map_key_specialization (gtype);
        val_gtype = dbus_g_type_get_map_value_specialization (gtype);
        key_subsig = _dbus_gtype_to_signature (key_gtype);
        val_subsig = _dbus_gtype_to_signature (val_gtype);
        ret = g_strconcat (DBUS_TYPE_ARRAY_AS_STRING
        DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING, key_subsig, val_subsig,
        DBUS_DICT_ENTRY_END_CHAR_AS_STRING, NULL);
        g_free (key_subsig);
        g_free (val_subsig);
    }
    else if (dbus_g_type_is_struct (gtype))
    {
        guint i, size;
        GString *sig;
        size = dbus_g_type_get_struct_size (gtype);
        sig = g_string_sized_new (size+2); /*some sensible starting
size*/
        g_string_assign (sig, DBUS_STRUCT_BEGIN_CHAR_AS_STRING);
        for (i = 0; i < size; i++)
        {
            gchar *subsig;
            subsig = _dbus_gtype_to_signature (
                dbus_g_type_get_struct_member_type (gtype, i));
            g_string_append (sig, subsig);
            g_free (subsig);
        }
        g_string_append (sig, DBUS_STRUCT_END_CHAR_AS_STRING);
        ret = g_string_free (sig, FALSE);
    }
}

```

```

    }
else
    {
        typedata = g_type_get_qdata (gtype,
dbus_g_type_metadata_data_quark ());
        if (typedata == NULL)
            return NULL;
        ret = g_strdup (typedata->sig);
    }
return ret;
}

char *
dbus_gvalue_to_signature (const GValue *val)
{
    GType gtype;

    gtype = G_VALUE_TYPE (val);
    if (g_type_is_a (gtype, G_TYPE_VALUE_ARRAY))
    {
        GString *str;
        guint i;
        GValueArray *array;

        array = g_value_get_boxed (val);

        str = g_string_new (DBUS_STRUCT_BEGIN_CHAR_AS_STRING);
        for (i = 0; i < array->n_values; i++)
        {
            char *sig;
            sig = _dbus_gvalue_to_signature (g_value_array_get_nth (array,
i));
            g_string_append (str, sig);
            g_free (sig);
        }
        g_string_append (str, DBUS_STRUCT_END_CHAR_AS_STRING);

        return g_string_free (str, FALSE);
    }
    else
        return _dbus_gtype_to_signature (gtype);
}

static gboolean
demarshal_basic (DBusGValueMarshalCtx      *context,
                DBusMessageIter            *iter,
                GValue                      *value,
                GError                      **error)
{
    int current_type;

    current_type = dbus_message_iter_get_arg_type (iter);

```

```

g_assert (dbus_type_is_basic (current_type));

switch (current_type)
{
case DBUS_TYPE_BOOLEAN:
{
dbus_bool_t bool;
if (!G_VALUE HOLDS (value, G_TYPE_BOOLEAN))
goto invalid_type;
dbus_message_iter_get_basic (iter, &bool);
g_value_set_boolean (value, bool);
return TRUE;
}
case DBUS_TYPE_BYTE:
{
unsigned char byte;
if (!G_VALUE HOLDS (value, G_TYPE_UCHAR))
goto invalid_type;
dbus_message_iter_get_basic (iter, &byte);
g_value_set_uchar (value, byte);
return TRUE;
}
case DBUS_TYPE_INT32:
{
dbus_int32_t intval;
if (!G_VALUE HOLDS (value, G_TYPE_INT))
goto invalid_type;
dbus_message_iter_get_basic (iter, &intval);
g_value_set_int (value, intval);
return TRUE;
}
case DBUS_TYPE_UINT32:
{
dbus_uint32_t intval;
if (!G_VALUE HOLDS (value, G_TYPE_UINT))
goto invalid_type;
dbus_message_iter_get_basic (iter, &intval);
g_value_set_uint (value, intval);
return TRUE;
}
case DBUS_TYPE_INT64:
{
dbus_int64_t intval;
if (!G_VALUE HOLDS (value, G_TYPE_INT64))
goto invalid_type;
dbus_message_iter_get_basic (iter, &intval);
g_value_set_int64 (value, intval);
return TRUE;
}
case DBUS_TYPE_UINT64:
{
dbus_uint64_t intval;

```



```

        if (!G_VALUE HOLDS (value, G_TYPE_UINT64))
            goto invalid_type;
        dbus_message_iter_get_basic (iter, &intval);
        g_value_set_uint64 (value, intval);
        return TRUE;
    }
case DBUS_TYPE_DOUBLE:
    {
        double dval;
        if (!G_VALUE HOLDS (value, G_TYPE_DOUBLE))
            goto invalid_type;
        dbus_message_iter_get_basic (iter, &dval);
        g_value_set_double (value, dval);
        return TRUE;
    }
case DBUS_TYPE_INT16:
    {
        dbus_int16_t v;
        if (!G_VALUE HOLDS (value, G_TYPE_INT))
            goto invalid_type;
        dbus_message_iter_get_basic (iter, &v);
        g_value_set_int (value, v);
        return TRUE;
    }
case DBUS_TYPE_UINT16:
    {
        dbus_uint16_t v;
        if (!G_VALUE HOLDS (value, G_TYPE_UINT))
            goto invalid_type;
        dbus_message_iter_get_basic (iter, &v);
        g_value_set_uint (value, v);
        return TRUE;
    }
case DBUS_TYPE_STRING:
    {
        const char *s;
        if (!G_VALUE HOLDS (value, G_TYPE_STRING))
            goto invalid_type;
        dbus_message_iter_get_basic (iter, &s);
        g_value_set_string (value, s);
        return TRUE;
    }
default:
    /* fall through to invalid_type */
    break;
}
invalid_type:
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Expected type %s, got type code '%c'",
                g_type_name (G_VALUE_TYPE (value)),

```

```

        (guchar) current_type);
    return FALSE;
}

static gboolean
demarshal_static_variant (DBusGValueMarshalCtx *context,
                          DBusMessageIter *iter,
                          GValue *value,
                          GError **error)
{
    char *sig;
    DBusMessageIter subiter;
    GType variant_type;

    dbus_message_iter_recurse (iter, &subiter);
    sig = dbus_message_iter_get_signature (&subiter);

    variant_type = _dbus_gtype_from_signature (sig, context->proxy !=
NULL);
    if (variant_type != G_TYPE_INVALID)
        {
            g_value_init (value, variant_type);

            if (!_dbus_gvalue_demarshal (context, &subiter, value, error))
                {
                    dbus_free (sig);
                    return FALSE;
                }
        }
    dbus_free (sig);
    return TRUE;
}

static gboolean
demarshal_variant (DBusGValueMarshalCtx *context,
                  DBusMessageIter *iter,
                  GValue *value,
                  GError **error)
{
    GValue *variant_val;
    variant_val = g_new0 (GValue, 1);

    if (!demarshal_static_variant (context, iter, variant_val, error))
        return FALSE;

    g_value_take_boxed (value, variant_val);
    return TRUE;
}

static gboolean
demarshal_proxy (DBusGValueMarshalCtx *context,

```

```

        DBusMessageIter      *iter,
        GValue                *value,
        GError                **error)
{
    DBusGProxy *new_proxy;
    const char *objpath;
    int current_type;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_OBJECT_PATH)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS object path, got type code \'%c\'",
                    (guchar) current_type);
        return FALSE;
    }

    g_assert (context->proxy != NULL);

    dbus_message_iter_get_basic (iter, &objpath);

    new_proxy = dbus_g_proxy_new_from_proxy (context->proxy, NULL,
    objpath);
    g_value_take_object (value, new_proxy);

    return TRUE;
}

static gboolean
demarshal_object_path (DBusGValueMarshalCtx *context,
                      DBusMessageIter      *iter,
                      GValue                *value,
                      GError                **error)
{
    const char *objpath;
    int current_type;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_OBJECT_PATH)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS object path, got type code \'%c\'",
                    (guchar) current_type);
        return FALSE;
    }

    dbus_message_iter_get_basic (iter, &objpath);

```

```

    g_value_set_boxed (value, objpath);

    return TRUE;
}

static gboolean
demarshal_object (DBusGValueMarshalCtx *context,
                  DBusMessageIter *iter,
                  GValue *value,
                  GError **error)
{
    const char *objpath;
    int current_type;
    GObject *obj;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_OBJECT_PATH)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS object path, got type code \'%c\'",
                    (guchar) current_type);
        return FALSE;
    }
    g_assert (context->proxy == NULL);

    dbus_message_iter_get_basic (iter, &objpath);

    obj = dbus_g_connection_lookup_g_object (context->gconnection,
                                             objpath);
    if (obj == NULL)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Unregistered object at path '%s'",
                    objpath);
        return FALSE;
    }
    g_value_set_object (value, obj);

    return TRUE;
}

static gboolean
demarshal_signature (DBusGValueMarshalCtx *context,
                    DBusMessageIter *iter,
                    GValue *value,
                    GError **error)
{
    const char *sig;

```

```

int current_type;

current_type = dbus_message_iter_get_arg_type (iter);
if (current_type != DBUS_TYPE_SIGNATURE)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Expected D-BUS signature, got type code \'%c\'",
(guchar) current_type);
    return FALSE;
}

dbus_message_iter_get_basic (iter, &sig);

g_value_set_boxed (value, sig);

return TRUE;
}

static gboolean
demarshal_strv (DBusGValueMarshalCtx *context,
                DBusMessageIter *iter,
                GValue *value,
                GError **error)
{
    DBusMessageIter subiter;
    int current_type;
    GArray *arr;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_ARRAY)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS array, got type code \'%c\'", (guchar)
current_type);
        return FALSE;
    }

    dbus_message_iter_recurse (iter, &subiter);

    current_type = dbus_message_iter_get_arg_type (&subiter);
    if (current_type != DBUS_TYPE_INVALID
        && current_type != DBUS_TYPE_STRING)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS string, got type code \'%c\'", (guchar)
current_type);

```

```

        return FALSE;
    }

    arr = g_array_new (TRUE, FALSE, sizeof (char *));

    while ((current_type = dbus_message_iter_get_arg_type (&subiter)) !=
DBUS_TYPE_INVALID)
    {
        g_assert (current_type == DBUS_TYPE_STRING);
        const char *str;
        char *copy;

        dbus_message_iter_get_basic (&subiter, &str);
        copy = g_strdup (str);
        g_array_append_val (arr, copy);

        dbus_message_iter_next (&subiter);
    }

    g_value_take_boxed (value, arr->data);
    g_array_free (arr, FALSE);

    return TRUE;
}

static gboolean
demarshal_valuearray (DBusGValueMarshalCtx *context,
                      DBusMessageIter *iter,
                      GValue *value,
                      GError **error)
{
    int current_type;
    GValueArray *ret;
    DBusMessageIter subiter;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_STRUCT)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS struct, got type code \'%c\'", (guchar)
current_type);
        return FALSE;
    }

    dbus_message_iter_recurse (iter, &subiter);

    ret = g_value_array_new (12);

    while ((current_type = dbus_message_iter_get_arg_type (&subiter)) !=
DBUS_TYPE_INVALID)

```

```

    {
        GValue *val;
        GType elt_type;
        char *current_sig;

        g_value_array_append (ret, NULL);
        val = g_value_array_get_nth (ret, ret->n_values - 1);

        current_sig = dbus_message_iter_get_signature (&subiter);
        elt_type = _dbus_gtype_from_signature (current_sig, TRUE);

        if (elt_type == G_TYPE_INVALID)
        {
            g_value_array_free (ret);
            g_set_error (error,
                        DBUS_GERROR,
                        DBUS_GERROR_INVALID_ARGS,
                        "Couldn't demarshal argument with signature \"%s\"",
current_sig);
            dbus_free (current_sig);
            return FALSE;
        }
        dbus_free (current_sig);

        g_value_init (val, elt_type);

        if (!_dbus_gvalue_demarshal (context, &subiter, val, error))
        {
            g_value_array_free (ret);
            return FALSE;
        }

        dbus_message_iter_next (&subiter);
    }

    g_value_take_boxed (value, ret);

    return TRUE;
}

static gboolean
demarshal_map (DBusGValueMarshalCtx *context,
               DBusMessageIter *iter,
               GValue *value,
               GError **error)
{
    GType gtype;
    DBusMessageIter subiter;
    int current_type;
    gpointer ret;
    GType key_gtype;
    GType value_gtype;

```

```

DBusGTypeSpecializedAppendContext appendctx;

current_type = dbus_message_iter_get_arg_type (iter);
if (current_type != DBUS_TYPE_ARRAY)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Expected D-BUS array, got type code \'%c\'", (guchar)
current_type);
    return FALSE;
}

gtype = G_VALUE_TYPE (value);

dbus_message_iter_recurse (iter, &subiter);

current_type = dbus_message_iter_get_arg_type (&subiter);
if (current_type != DBUS_TYPE_INVALID
    && current_type != DBUS_TYPE_DICT_ENTRY)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Expected D-BUS dict entry, got type code \'%c\'",
(guchar) current_type);
    return FALSE;
}

key_gtype = dbus_g_type_get_map_key_specialization (gtype);
value_gtype = dbus_g_type_get_map_value_specialization (gtype);

ret = dbus_g_type_specialized_construct (gtype);
g_value_take_boxed (value, ret);

dbus_g_type_specialized_init_append (value, &appendctx);

while ((current_type = dbus_message_iter_get_arg_type (&subiter)) !=
DBUS_TYPE_INVALID)
{
    DBusMessageIter entry_iter;
    GValue key_value = {0,};
    GValue value_value = {0,};

    current_type = dbus_message_iter_get_arg_type (&subiter);
    g_assert (current_type == DBUS_TYPE_DICT_ENTRY);

    dbus_message_iter_recurse (&subiter, &entry_iter);

    g_value_init (&key_value, key_gtype);
    if (!_dbus_gvalue_demarshal (context,
                                &entry_iter,

```



```

        &key_value,
        error))

return FALSE;

dbus_message_iter_next (&entry_iter);

g_value_init (&value_value, value_gtype);
if (!_dbus_gvalue_demarshal (context,
                             &entry_iter,
                             &value_value,
                             error))

return FALSE;

dbus_g_type_specialized_map_append (&appendctx, &key_value,
&value_value);
/* Ownership of values passes to map, don't unset */

dbus_message_iter_next (&subiter);
}

return TRUE;
}

static gboolean
demarshal_struct (DBusGValueMarshalCtx *context,
                 DBusMessageIter *iter,
                 GValue *value,
                 GError **error)
{
    int current_type;
    DBusMessageIter subiter;
    guint i, size;
    GValue val = {0,};
    GType elt_type;

    current_type = dbus_message_iter_get_arg_type (iter);
    if (current_type != DBUS_TYPE_STRUCT)
    {
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Expected D-BUS struct, got type code \'%c\'",
                    (guchar) current_type);
        return FALSE;
    }

    dbus_message_iter_recurse (iter, &subiter);

    g_value_take_boxed (value,
                        dbus_g_type_specialized_construct (G_VALUE_TYPE (value)));

    size = dbus_g_type_get_struct_size (G_VALUE_TYPE (value));

```

```

for (i=0; i < size; i++)
{
    elt_type = dbus_g_type_get_struct_member_type
(G_VALUE_TYPE(value), i);
    if (elt_type == G_TYPE_INVALID)
    {
        g_value_unset (value);
        g_set_error (error,
                    DBUS_GERROR,
                    DBUS_GERROR_INVALID_ARGS,
                    "Couldn't demarshal argument, "
                    "struct type %s has no member %d",
                    g_type_name (G_VALUE_TYPE(value)), i);
        return FALSE;
    }

    g_value_init (&val, elt_type);

    if (!_dbus_gvalue_demarshal (context, &subiter, &val, error))
    {
        g_value_unset (&val);
        g_value_unset (value);
        return FALSE;
    }
    if (!_dbus_g_type_struct_set_member (value, i, &val))
    {
        g_value_unset (&val);
        g_value_unset (value);
        return FALSE;
    }

    dbus_message_iter_next (&subiter);
    g_value_unset (&val);
}

g_assert (dbus_message_iter_get_arg_type (&subiter) ==
DBUS_TYPE_INVALID);

return TRUE;
}

static DBusGValueDemarshalFunc
get_type_demarshaller (GType type)
{
    DBusGTypeMarshalData *typedata;

    typedata = g_type_get_qdata (type, dbus_g_type_metadata_data_quark
());
    if (typedata == NULL)

```

```

    {
        if (g_type_is_a (type, G_TYPE_VALUE_ARRAY))
            return demarshal_valuearray;
        if (dbus_g_type_is_collection (type))
            return demarshal_collection;
        if (dbus_g_type_is_map (type))
            return demarshal_map;
        if (dbus_g_type_is_struct (type))
            return demarshal_struct;

        g_warning ("No demarshaller registered for type \"%s\"",
g_type_name (type));
        return NULL;
    }
    g_assert (typedata->vtable);
    return typedata->vtable->demarshaller;
}

static gboolean
demarshal_collection (DBusGValueMarshalCtx *context,
                    DBusMessageIter *iter,
                    GValue *value,
                    GError **error)
{
    GType coltype;
    GType subtype;

    coltype = G_VALUE_TYPE (value);
    subtype = dbus_g_type_get_collection_specialization (coltype);

    if (_dbus_g_type_is_fixed (subtype))
        return demarshal_collection_array (context, iter, value, error);
    else
        return demarshal_collection_ptrarray (context, iter, value,
error);
}

static gboolean
demarshal_collection_ptrarray (DBusGValueMarshalCtx *context,
                    DBusMessageIter *iter,
                    GValue *value,
                    GError **error)
{
    GType coltype;
    GType subtype;
    gpointer instance;
    DBusGTypeSpecializedAppendContext ctx;
    DBusGValueDemarshalFunc demarshaller;
    DBusMessageIter subiter;
    int current_type;

    current_type = dbus_message_iter_get_arg_type (iter);

```

```

if (current_type != DBUS_TYPE_ARRAY)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Expected D-BUS array, got type code \'%c\'", (guchar)
current_type);
    return FALSE;
}

dbus_message_iter_recurse (iter, &subiter);

coltype = G_VALUE_TYPE (value);
subtype = dbus_g_type_get_collection_specialization (coltype);

demarshaller = get_type_demarshaller (subtype);

if (!demarshaller)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "No demarshaller registered for type \"%s\" of
collection \"%s\"",
                g_type_name (coltype),
                g_type_name (subtype));
    return FALSE;
}

instance = dbus_g_type_specialized_construct (coltype);
g_value_take_boxed (value, instance);

dbus_g_type_specialized_init_append (value, &ctx);

while ((current_type = dbus_message_iter_get_arg_type (&subiter)) !=
DBUS_TYPE_INVALID)
{
    GValue eltval = {0, };

    g_value_init (&eltval, subtype);

    if (!demarshaller (context, &subiter, &eltval, error))
    {
        dbus_g_type_specialized_collection_end_append (&ctx);
        g_value_unset (value);
        return FALSE;
    }
    dbus_g_type_specialized_collection_append (&ctx, &eltval);

    dbus_message_iter_next (&subiter);
}

```

```

    dbus_g_type_specialized_collection_end_append (&ctx);

    return TRUE;
}

static gboolean
demarshal_collection_array (DBusGValueMarshalCtx    *context,
                            DBusMessageIter         *iter,
                            GValue                  *value,
                            GError                  **error)
{
    DBusMessageIter subiter;
    GArray *ret;
    GType elt_gtype;
    int elt_size;
    void *msgarray;
    int msgarray_len;

    dbus_message_iter_recurse (iter, &subiter);

    elt_gtype = dbus_g_type_get_collection_specialization (G_VALUE_TYPE
(value));
    g_assert (elt_gtype != G_TYPE_INVALID);
    g_assert (_dbus_g_type_is_fixed (elt_gtype));

    elt_size = _dbus_g_type_fixed_get_size (elt_gtype);

    ret = g_array_new (FALSE, TRUE, elt_size);

    msgarray = NULL;
    dbus_message_iter_get_fixed_array (&subiter,
                                       &msgarray,
                                       &msgarray_len);
    g_assert (msgarray != NULL || msgarray_len == 0);

    if (msgarray_len)
        g_array_append_vals (ret, msgarray, (guint) msgarray_len);

    g_value_take_boxed (value, ret);

    return TRUE;
}

gboolean
_dbus_gvalue_demarshal (DBusGValueMarshalCtx    *context,
                        DBusMessageIter         *iter,
                        GValue                  *value,
                        GError                  **error)
{
    GType gtype;
    DBusGValueDemarshalFunc demarshaller;
    gboolean retcode = FALSE;

```

```

if (context->recursion_depth > DBUS_GLIB_MAX_VARIANT_RECURSION)
{
    g_set_error (error, DBUS_GERROR,
                DBUS_GERROR_NO_MEMORY,
                "Variant recursion limit exceeded");
    return FALSE;
}
context->recursion_depth++;

gtype = G_VALUE_TYPE (value);

demarshaller = get_type_demarshaller (gtype);

if (demarshaller == NULL)
{
    g_set_error (error,
                DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "No demarshaller registered for type \"%s\"",
                g_type_name (gtype));

    goto out;
}

retcode = demarshaller (context, iter, value, error);
out:
context->recursion_depth--;
return retcode;
}

gboolean
_dbus_gvalue_demarshal_variant (DBusGValueMarshalCtx    *context,
                                DBusMessageIter          *iter,
                                GValue                   *value,
                                GError                    **error)
{
    return demarshal_static_variant (context, iter, value, error);
}

GValueArray *
_dbus_gvalue_demarshal_message (DBusGValueMarshalCtx    *context,
                                DBusMessage              *message,
                                guint                     n_types,
                                const GType              *types,
                                GError                    **error)
{
    GValueArray *ret;
    DBusMessageIter iter;
    int current_type;
    guint index_;

```

```

    ret = g_value_array_new (6); /* 6 is a typical maximum for
arguments */

    dbus_message_iter_init (message, &iter);
    index_ = 0;
    while ((current_type = dbus_message_iter_get_arg_type (&iter)) !=
DBUS_TYPE_INVALID)
    {
        GValue *value;
        GType gtype;

        if (index_ >= n_types)
        {
            g_set_error (error, DBUS_GERROR,
                DBUS_GERROR_INVALID_ARGS,
                "Too many arguments in message");
            goto lose;
        }

        g_value_array_append (ret, NULL);
        value = g_value_array_get_nth (ret, index_);

        gtype = types[index_];
        g_value_init (value, gtype);

        if (!_dbus_gvalue_demarshal (context, &iter, value, error))
            goto lose;
        dbus_message_iter_next (&iter);
        index_++;
    }
    if (index_ < n_types)
    {
        g_set_error (error, DBUS_GERROR,
            DBUS_GERROR_INVALID_ARGS,
            "Too few arguments in message");
        goto lose;
    }

    return ret;
lose:
    g_value_array_free (ret);
    return NULL;
}

static void oom (void) G_GNUC_NORETURN;
static void
oom (void)
{
    g_error ("no memory");
}

static gboolean

```



```

                                &v))
        oom ();
    }
    return TRUE;
case G_TYPE_LONG:
    {
        dbus_int32_t v = g_value_get_long (value);
        if (!dbus_message_iter_append_basic (iter,
                                             DBUS_TYPE_INT32,
                                             &v))

            oom ();
    }
    return TRUE;
case G_TYPE_ULONG:
    {
        dbus_uint32_t v = g_value_get_ulong (value);
        if (!dbus_message_iter_append_basic (iter,
                                             DBUS_TYPE_UINT32,
                                             &v))

            oom ();
    }
    return TRUE;
case G_TYPE_INT64:
    {
        gint64 v = g_value_get_int64 (value);
        if (!dbus_message_iter_append_basic (iter,
                                             DBUS_TYPE_INT64,
                                             &v))

            oom ();
    }
    return TRUE;
case G_TYPE_UINT64:
    {
        guint64 v = g_value_get_uint64 (value);
        if (!dbus_message_iter_append_basic (iter,
                                             DBUS_TYPE_UINT64,
                                             &v))

            oom ();
    }
    return TRUE;
case G_TYPE_FLOAT:
    {
        double v = g_value_get_float (value);

        if (!dbus_message_iter_append_basic (iter,
                                             DBUS_TYPE_DOUBLE,
                                             &v))

            oom ();
    }
    return TRUE;
case G_TYPE_DOUBLE:
    {

```

```

    double v = g_value_get_double (value);

    if (!dbus_message_iter_append_basic (iter,
                                         DBUS_TYPE_DOUBLE,
                                         &v))
        oom ();
}
return TRUE;
case G_TYPE_STRING:
{
    const char *v = g_value_get_string (value);
    if (!v)
        v = "";

    /* FIXME: fd.o #16320: consider using g_return_if_fail to
check UTF-8
    * validity */

    if (!dbus_message_iter_append_basic (iter,
                                         DBUS_TYPE_STRING,
                                         &v))
    {
        gchar *s = g_strdup_value_contents (value);

        g_critical ("Unable to marshal string (not UTF-8 or OOM?):
%s",
                    s);
        g_free (s);
        return FALSE;
    }
}
return TRUE;

default:
{
    g_assert_not_reached ();
    return FALSE;
}
}

static gboolean
marshal_strv (DBusMessageIter *iter,
              const GValue *value)
{
    DBusMessageIter subiter;
    char **array;
    char **elt;
    gboolean ret = FALSE;

    g_assert (G_VALUE_TYPE (value) == g_strv_get_type ());

```

```

array = g_value_get_boxed (value);

if (!dbus_message_iter_open_container (iter,
                                       DBUS_TYPE_ARRAY,
                                       "s",
                                       &subiter))
    goto out;

if (array)
{
    for (elt = array; *elt; elt++)
    {
        if (!dbus_message_iter_append_basic (&subiter,
                                             DBUS_TYPE_STRING,
                                             elt))
            goto out;
    }
}

if (!dbus_message_iter_close_container (iter, &subiter))
    goto out;
ret = TRUE;
out:
return ret;
}

static gboolean
marshal_valuearray (DBusMessageIter *iter,
                   const GValue *value)
{
    GValueArray *array;
    guint i;
    DBusMessageIter subiter;

    g_assert (G_VALUE_TYPE (value) == G_TYPE_VALUE_ARRAY);

    array = g_value_get_boxed (value);

    if (!dbus_message_iter_open_container (iter,
                                           DBUS_TYPE_STRUCT,
                                           NULL,
                                           &subiter))
        oom ();

    if (array)
    {
        for (i = 0; i < array->n_values; i++)
        {
            if (!_dbus_gvalue_marshal (&subiter, g_value_array_get_nth
(array, i)))
            {
                dbus_message_iter_abandon_container (iter, &subiter);
            }
        }
    }
}

```

```

        return FALSE;
    }
}

return dbus_message_iter_close_container (iter, &subiter);
}

static gboolean
marshal_proxy (DBusMessageIter      *iter,
               const GValue          *value)
{
    const char *path;
    DBusGProxy *proxy;

    g_assert (G_VALUE_TYPE (value) == dbus_g_proxy_get_type ());

    proxy = g_value_get_object (value);
    g_return_val_if_fail (DBUS_IS_G_PROXY (proxy), FALSE);
    path = dbus_g_proxy_get_path (proxy);
    g_return_val_if_fail (g_variant_is_object_path (path), FALSE);

    if (!dbus_message_iter_append_basic (iter,
                                         DBUS_TYPE_OBJECT_PATH,
                                         &path))
        oom ();

    return TRUE;
}

static gboolean
marshal_object_path (DBusMessageIter      *iter,
                     const GValue          *value)
{
    const char *path;

    g_assert (G_VALUE_TYPE (value) == DBUS_TYPE_G_OBJECT_PATH);

    path = g_value_get_boxed (value);
    g_return_val_if_fail (g_variant_is_object_path (path), FALSE);

    if (!dbus_message_iter_append_basic (iter,
                                         DBUS_TYPE_OBJECT_PATH,
                                         &path))
        oom ();

    return TRUE;
}

static gboolean
marshal_object (DBusMessageIter      *iter,
                const GValue          *value)

```

```

{
    const char *path;
    GObject *obj;

    obj = g_value_get_object (value);
    g_return_val_if_fail (G_IS_OBJECT (obj), FALSE);

    path = _dbus_gobject_get_path (obj);
    g_return_val_if_fail (g_variant_is_object_path (path), FALSE);

    if (!dbus_message_iter_append_basic (iter,
                                        DBUS_TYPE_OBJECT_PATH,
                                        &path))

        oom ();

    return TRUE;
}

static gboolean
marshal_signature (DBusMessageIter *iter,
                  const GValue *value)
{
    const char *sig;

    g_assert (G_VALUE_TYPE (value) == DBUS_TYPE_G_SIGNATURE);

    sig = g_value_get_boxed (value);
    g_return_val_if_fail (g_variant_is_signature (sig), FALSE);

    /* failure here isn't strictly *guaranteed* to be OOM, since GDBus
might
* understand more type-codes than our libdbus */
    if (!dbus_message_iter_append_basic (iter,
                                        DBUS_TYPE_SIGNATURE,
                                        &sig))

        return FALSE;

    return TRUE;
}

struct DBusGLibHashMarshalData
{
    const char *entry_sig;
    DBusMessageIter *iter;
    gboolean err;
};

static void
marshal_map_entry (const GValue *key,
                  const GValue *value,
                  gpointer data)
{

```

```

struct DBusGLibHashMarshalData *hashdata = data;
DBusMessageIter subiter;

if (hashdata->err)
    return;

if (!dbus_message_iter_open_container (hashdata->iter,
                                       DBUS_TYPE_DICT_ENTRY,
                                       NULL,
                                       &subiter))

    goto lose;

if (!_dbus_gvalue_marshal (&subiter, key))
    goto lose;

if (!_dbus_gvalue_marshal (&subiter, value))
    goto lose;

if (!dbus_message_iter_close_container (hashdata->iter, &subiter))
    goto lose;

return;
lose:
hashdata->err = TRUE;
}

static gboolean
marshal_map (DBusMessageIter *iter,
            const GValue *value)
{
    GType gtype;
    DBusMessageIter arr_iter;
    struct DBusGLibHashMarshalData hashdata;
    char *key_sig;
    char *value_sig;
    GType key_type;
    GType value_type;
    char *entry_sig;
    char *array_sig;

    gtype = G_VALUE_TYPE (value);

    key_type = dbus_g_type_get_map_key_specialization (gtype);
    g_assert (_dbus_gtype_is_valid_hash_key (key_type));
    value_type = dbus_g_type_get_map_value_specialization (gtype);
    g_assert (_dbus_gtype_is_valid_hash_value (value_type));

    key_sig = _dbus_gtype_to_signature (key_type);
    if (!key_sig)
    {
        g_warning ("Cannot marshal type \"%s\" in map\n", g_type_name
(key_type));
    }
}

```

```

        return FALSE;
    }
    value_sig = _dbus_gtype_to_signature (value_type);
    if (!value_sig)
    {
        g_free (key_sig);
        g_warning ("Cannot marshal type \"%s\" in map\n", g_type_name
(value_type));
        return FALSE;
    }
    entry_sig = g_strdup_printf ("%s%s", key_sig, value_sig);
    g_free (key_sig);
    g_free (value_sig);
    array_sig = g_strdup_printf ("%c%s%c",
                                DBUS_DICT_ENTRY_BEGIN_CHAR,
                                entry_sig,
                                DBUS_DICT_ENTRY_END_CHAR);
    if (!dbus_message_iter_open_container (iter,
                                DBUS_TYPE_ARRAY,
                                array_sig,
                                &arr_iter))

        goto lose;

    hashdata.iter = &arr_iter;
    hashdata.err = FALSE;
    hashdata.entry_sig = entry_sig;

    dbus_g_type_map_value_iterate (value,
                                marshal_map_entry,
                                &hashdata);

    if (hashdata.err)
    {
        dbus_message_iter_abandon_container (iter, &arr_iter);
        goto lose;
    }
    else if (!dbus_message_iter_close_container (iter, &arr_iter))
    {
        goto lose;
    }
}

out:
    g_free (entry_sig);
    g_free (array_sig);
    return !hashdata.err;
lose:
    hashdata.err = TRUE;
    goto out;
}

static gboolean
marshal_struct (DBusMessageIter *iter,

```

```

                const GValue      *value)
{
    GType gtype;
    DBusMessageIter subiter;
    guint size, i;
    GValue val = {0,};

    gtype = G_VALUE_TYPE (value);

    size = dbus_g_type_get_struct_size (gtype);

    if (!dbus_message_iter_open_container (iter,
                                          DBUS_TYPE_STRUCT,
                                          NULL,
                                          &subiter))
        oom ();

    for (i = 0; i < size; i++)
    {
        g_value_init (&val, dbus_g_type_get_struct_member_type
                     (G_VALUE_TYPE(value), i));

        if (!dbus_g_type_struct_get_member (value, i, &val))
            goto abandon;

        if (!_dbus_gvalue_marshal (&subiter, &val))
            goto abandon;

        g_value_unset(&val);
    }

    return dbus_message_iter_close_container (iter, &subiter);

abandon:
    dbus_message_iter_abandon_container (iter, &subiter);
    return FALSE;
}

static gboolean
marshal_variant (DBusMessageIter      *iter,
                 const GValue          *value)
{
    GType value_gtype;
    DBusMessageIter subiter;
    char *variant_sig;
    GValue *real_value;
    gboolean ret = FALSE;

    real_value = g_value_get_boxed (value);
    value_gtype = G_VALUE_TYPE (real_value);

    variant_sig = _dbus_gvalue_to_signature (real_value);

```



```

    if (variant_sig == NULL)
    {
        g_warning ("Cannot marshal type \"%s\" in variant", g_type_name
(value_gtype));
        return FALSE;
    }

    if (!dbus_message_iter_open_container (iter,
        DBUS_TYPE_VARIANT,
        variant_sig,
        &subiter))

        goto out;

    if (!_dbus_gvalue_marshal (&subiter, real_value))
    {
        dbus_message_iter_abandon_container (iter, &subiter);
        goto out;
    }

    if (!dbus_message_iter_close_container (iter, &subiter))
        goto out;

    ret = TRUE;
out:
    g_free (variant_sig);
    return ret;
}

static DBusGValueMarshalFunc
get_type_marshallers (GType type)
{
    DBusGTypeMarshalData *typedata;

    typedata = g_type_get_qdata (type, dbus_g_type_metadata_data_quark
());
    if (typedata == NULL)
    {
        if (g_type_is_a (type, G_TYPE_VALUE_ARRAY))
            return marshal_valuearray;
        if (dbus_g_type_is_collection (type))
            return marshal_collection;
        if (dbus_g_type_is_map (type))
            return marshal_map;
        if (dbus_g_type_is_struct (type))
            return marshal_struct;

        g_warning ("No marshaller registered for type \"%s\"",
g_type_name (type));
        return NULL;
    }
    g_assert (typedata->vtable);
    return typedata->vtable->marshaller;
}

```

```

}

typedef struct
{
    DBusMessageIter *iter;
    DBusGValueMarshalFunc marshaller;
    gboolean err;
} DBusGValueCollectionMarshalData;

static void
collection_marshall_iterator (const GValue *eltval,
                             gpointer      user_data)
{
    DBusGValueCollectionMarshalData *data = user_data;

    if (data->err)
        return;

    if (!data->marshaller (data->iter, eltval))
        data->err = TRUE;
}

static gboolean
marshal_collection (DBusMessageIter *iter,
                   const GValue     *value)
{
    GType coltype;
    GType subtype;

    coltype = G_VALUE_TYPE (value);
    subtype = dbus_g_type_get_collection_specialization (coltype);

    if (_dbus_g_type_is_fixed (subtype))
        return marshal_collection_array (iter, value);
    else
        return marshal_collection_ptrarray (iter, value);
}

static gboolean
marshal_collection_ptrarray (DBusMessageIter *iter,
                             const GValue     *value)
{
    GType coltype;
    GType elt_gtype;
    DBusGValueCollectionMarshalData data;
    DBusMessageIter subiter;
    char *elt_sig;

    coltype = G_VALUE_TYPE (value);
    elt_gtype = dbus_g_type_get_collection_specialization (coltype);
    data.marshaller = get_type_marshaller (elt_gtype);
    if (!data.marshaller)

```

```

    return FALSE;

    elt_sig = _dbus_gtype_to_signature (elt_gtype);
    if (!elt_sig)
    {
        g_warning ("Cannot marshal type \"%s\" in collection\n",
g_type_name (elt_gtype));
        return FALSE;
    }
    g_assert (g_variant_is_signature (elt_sig));

    if (!dbus_message_iter_open_container (iter,
                                        DBUS_TYPE_ARRAY,
                                        elt_sig,
                                        &subiter))

        oom ();

    g_free (elt_sig);

    data.iter = &subiter;
    data.err = FALSE;

    dbus_g_type_collection_value_iterate (value,
                                        collection_marshal_iterator,
                                        &data);

    if (data.err)
    {
        dbus_message_iter_abandon_container (iter, &subiter);
        return FALSE;
    }

    return dbus_message_iter_close_container (iter, &subiter);
}

/* If any of these assertions are violated, then
marshal_collection_array
* is buggy for that type. dbus_g_value_basic_array_parse_variant()
has
* general size-conversion code, if needed. */
G_STATIC_ASSERT (sizeof (dbus_bool_t) == sizeof (gboolean));
G_STATIC_ASSERT (sizeof (dbus_int32_t) == sizeof (gint));
G_STATIC_ASSERT (sizeof (dbus_uint32_t) == sizeof (guint));
/* These should be true regardless, but just for completeness... */
G_STATIC_ASSERT (sizeof (dbus_int64_t) == sizeof (gint64));
G_STATIC_ASSERT (sizeof (dbus_uint64_t) == sizeof (guint64));
G_STATIC_ASSERT (sizeof (double) == sizeof (gdouble));

static gboolean
marshal_collection_array (DBusMessageIter *iter,
                        const GValue *value)
{

```

```

GType elt_gtype;
DBusMessageIter subiter;
GArray *array;
char *subsignature_str;

array = g_value_get_boxed (value);
g_return_val_if_fail (array != NULL, FALSE);

elt_gtype = dbus_g_type_get_collection_specialization (G_VALUE_TYPE
(value));
g_assert (_dbus_g_type_is_fixed (elt_gtype));
subsignature_str = _dbus_gtype_to_signature (elt_gtype);
if (!subsignature_str)
{
    g_warning ("Cannot marshal type \"%s\" in collection\n",
g_type_name (elt_gtype));
    return FALSE;
}
g_assert (g_variant_is_signature (subsignature_str));

if (!dbus_message_iter_open_container (iter,
DBUS_TYPE_ARRAY,
subsignature_str,
&subiter))

    oom ();

/* TODO - This assumes that basic values are the same size
 * is this always true?  If it is we can probably avoid
 * a lot of the overhead in _marshal_basic_instance...
 */
if (!dbus_message_iter_append_fixed_array (&subiter,
subsignature_str[0],
&(array->data),
array->len))
{
    g_critical ("Unable to serialize %u GArray members as signature
%s "
"(OOM or invalid boolean value?)", array->len,
subsignature_str);

    g_free (subsignature_str);
    dbus_message_iter_abandon_container (iter, &subiter);
    return FALSE;
}

g_free (subsignature_str);
return dbus_message_iter_close_container (iter, &subiter);
}

gboolean
_dbus_gvalue_marshal (DBusMessageIter *iter,
const GValue *value)

```

```

{
    GType gtype;
    DBusGValueMarshalFunc marshaller;

    gtype = G_VALUE_TYPE (value);

    marshaller = get_type_marshaller (gtype);
    if (marshaller == NULL)
        return FALSE;
    return marshaller (iter, value);
}

#ifdef DBUS_BUILD_TESTS

static void
assert_type_maps_to (GType gtype, const char *expected_sig)
{
    char *sig;
    sig = _dbus_gtype_to_signature (gtype);
    g_assert (sig != NULL);
    g_assert (!strcmp (expected_sig, sig));
    g_free (sig);
}

static void
assert_signature_maps_to (const char *sig, GType expected_gtype)
{
    g_assert (_dbus_gtype_from_signature (sig, TRUE) == expected_gtype);
}

static void
assert_bidirectional_mapping (GType gtype, const char *expected_sig)
{
    assert_type_maps_to (gtype, expected_sig);
    assert_signature_maps_to (expected_sig, gtype);
}

/*
 * Unit test for general glib stuff
 * Returns: %TRUE on success.
 */
gboolean
_dbus_gvalue_test (const char *test_data_dir)
{
    _dbus_g_value_types_init ();

    assert_bidirectional_mapping (G_TYPE_STRING,
DBUS_TYPE_STRING_AS_STRING);
    assert_bidirectional_mapping (G_TYPE_UCHAR,
DBUS_TYPE_BYTE_AS_STRING);
    assert_bidirectional_mapping (G_TYPE_UINT,
DBUS_TYPE_UINT32_AS_STRING);
}

```

```

    assert_bidirectional_mapping (dbus_g_type_get_map ("GHashTable",
G_TYPE_STRING, G_TYPE_VALUE),
                                DBUS_TYPE_ARRAY_AS_STRING
DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING DBUS_TYPE_STRING_AS_STRING
DBUS_TYPE_VARIANT_AS_STRING DBUS_DICT_ENTRY_END_CHAR_AS_STRING);
    assert_bidirectional_mapping (dbus_g_type_get_collection
("GPtrArray", DBUS_TYPE_G_OBJECT_PATH),
                                DBUS_TYPE_ARRAY_AS_STRING
DBUS_TYPE_OBJECT_PATH_AS_STRING);
    assert_bidirectional_mapping (dbus_g_type_get_collection ("GArray",
G_TYPE_INT),
                                DBUS_TYPE_ARRAY_AS_STRING
DBUS_TYPE_INT32_AS_STRING);

    assert_bidirectional_mapping (dbus_g_type_get_struct ("GValueArray",
G_TYPE_INT, G_TYPE_STRING, DBUS_TYPE_G_OBJECT_PATH, G_TYPE_INVALID),
                                DBUS_STRUCT_BEGIN_CHAR_AS_STRING
DBUS_TYPE_INT32_AS_STRING DBUS_TYPE_STRING_AS_STRING
DBUS_TYPE_OBJECT_PATH_AS_STRING DBUS_STRUCT_END_CHAR_AS_STRING );
    return TRUE;
}

```

```
#endif /* DBUS_BUILD_TESTS */
```

```
File = dbus-gvalue.h
```

```

#ifndef DBUS_GOBJECT_VALUE_H
#define DBUS_GOBJECT_VALUE_H

#include <dbus/dbus.h>
#include <dbus/dbus-signature.h>
#include <glib.h>
#include <glib-object.h>
#include "dbus/dbus-glib.h"

```

```
G_BEGIN_DECLS
```

```

typedef struct {
    DBusGConnection    *gconnection;
    DBusGProxy         *proxy;
    guint               recursion_depth;
} DBusGValueMarshalCtx;

```

```

void                _dbus_g_value_types_init        (void);

char *              _dbus_gtype_to_signature        (GType
type);
char *              _dbus_gvalue_to_signature      (const GValue
*val);

```

```

gboolean      _dbus_gvalue_demarshal      (DBusGValueMarshalCtx
*context,
                                         DbusMessageIter      *iter,
                                         GValue                *value,
                                         GError                **error);

gboolean      _dbus_gvalue_demarshal_variant (DBusGValueMarshalCtx
*context,
                                         DbusMessageIter      *iter,
                                         GValue                *value,
                                         GError                **error);

GValueArray * _dbus_gvalue_demarshal_message (DBusGValueMarshalCtx
*context,
                                         DbusMessage          *message,
                                         guint                n_params,
                                         const GType          *types,
                                         GError                **error);

gboolean      _dbus_gvalue_marshal        (DBusMessageIter
*iter,
                                         const GValue          *value);

G_END_DECLS

#endif /* DBUS_GOBJECT_VALUE_H */

```

File = dbus-hash.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-hash.c Generic hash table utility (internal to D-Bus
implementation)
*
* Copyright (C) 2002 Red Hat, Inc.
* Copyright (c) 1991-1993 The Regents of the University of
California.
* Copyright (c) 1994 Sun Microsystems, Inc.
*
* Hash table implementation based on generic/tclHash.c from the Tcl
* source code. The original Tcl license applies to portions of the
* code from tclHash.c; the Tcl license follows this standad D-Bus
* license information.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by

```

* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
/*
* The following copyright applies to code from the Tcl distribution.
*
* Copyright (c) 1991-1993 The Regents of the University of
California.
* Copyright (c) 1994 Sun Microsystems, Inc.
*
* This software is copyrighted by the Regents of the University of
* California, Sun Microsystems, Inc., Scriptics Corporation, and
* other parties. The following terms apply to all files associated
* with the software unless explicitly disclaimed in individual files.
*
* The authors hereby grant permission to use, copy, modify,
* distribute, and license this software and its documentation for any
* purpose, provided that existing copyright notices are retained in
* all copies and that this notice is included verbatim in any
* distributions. No written agreement, license, or royalty fee is
* required for any of the authorized uses. Modifications to this
* software may be copyrighted by their authors and need not follow
* the licensing terms described here, provided that the new terms are
* clearly indicated on the first page of each file where they apply.
*
* IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY
* PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
* DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION,
* OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*
* THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND
* NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS,
* AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE
* MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
*
* GOVERNMENT USE: If you are acquiring this software on behalf of the
* U.S. government, the Government shall have only "Restricted Rights"
* in the software and related documentation as defined in the Federal


```

* Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2).  If you
* are acquiring the software on behalf of the Department of Defense,
* the software shall be classified as "Commercial Computer Software"
* and the Government shall have only "Restricted Rights" as defined
* in Clause 252.227-7013 (c) (1) of DFARs.  Notwithstanding the
* foregoing, the authors grant the U.S. Government and others acting
* in its behalf permission to use and distribute the software in
* accordance with the terms specified in this license.
*/

#include <config.h>
#include "dbus-hash.h"
#include "dbus-internals.h"
#include "dbus-mempool.h"

/**
 * @defgroup DBusHashTable Hash table
 * @ingroup DBusInternals
 * @brief DBusHashTable data structure
 *
 * Types and functions related to DBusHashTable.
 */

/**
 * @defgroup DBusHashTableInternals Hash table implementation details
 * @ingroup DBusInternals
 * @brief DBusHashTable implementation details
 *
 * The guts of DBusHashTable.
 *
 * @{
 */

/**
 * When there are this many entries per bucket, on average, rebuild
 * the hash table to make it larger.
 */
#define REBUILD_MULTIPLIER 3

/**
 * Takes a preliminary integer hash value and produces an index into a
 * hash tables bucket list.  The idea is to make it so that
 * preliminary values that are arbitrarily similar will end up in
 * different buckets.  The hash function was taken from a
 * random-number generator.  (This is used to hash integers.)
 *
 * The down_shift drops off the high bits of the hash index, and
 * decreases as we increase the number of hash buckets (to keep more
 * range in the hash index).  The mask also strips high bits and strips
 * fewer high bits as the number of hash buckets increases.
 * I don't understand two things: why is the initial downshift 28
 * to keep 4 bits when the initial mask is 011 to keep 2 bits,

```

```

* and why do we have both a mask and a downshift?
*
*/
#define RANDOM_INDEX(table, i) \
    (((((intptr_t) (i))*1103515245) >> (table)->down_shift) & (table)->mask)

/**
 * Initial number of buckets in hash table (hash table statically
 * allocates its buckets for this size and below).
 * The initial mask has to be synced to this.
 */
#define DBUS_SMALL_HASH_TABLE 4

/**
 * Typedef for DBusHashEntry
 */
typedef struct DBusHashEntry DBusHashEntry;

/**
 * @brief Internal representation of a hash entry.
 *
 * A single entry (key-value pair) in the hash table.
 * Internal to hash table implementation.
 */
struct DBusHashEntry
{
    DBusHashEntry *next;    /**< Pointer to next entry in this
                            * hash bucket, or #NULL for end of
                            * chain.
                            */
    void *key;              /**< Hash key */
    void *value;            /**< Hash value */
};

/**
 * Function used to find and optionally create a hash entry.
 */
typedef DBusHashEntry* (* DBusFindEntryFunction) (DBusHashTable
*table,
                                                void
*key,
                                                dbus_bool_t
create_if_not_found,
                                                DBusHashEntry
***bucket,
                                                DBusPreallocatedHash
*preallocated);

/**
 * @brief Internals of DBusHashTable.
 *

```

```

* Hash table internals. Hash tables are opaque objects, they must be
* used via accessor functions.
*/
struct DBusHashTable {
    int refcount;                                /**< Reference count */

    DBusHashEntry **buckets;                    /**< Pointer to bucket array.
Each
                                                * element points to first entry
in
                                                * bucket's hash chain, or
#NULL.
                                                */
    DBusHashEntry *static_buckets[DBUS_SMALL_HASH_TABLE];
small tables
                                                /**< Bucket array used for
frees).
                                                * (to avoid mallocs and
                                                */
    int n_buckets;                              /**< Total number of buckets
allocated
                                                * at **buckets.
                                                */
    int n_entries;                              /**< Total number of entries
present
                                                * in table.
                                                */
    int hi_rebuild_size;                        /**< Enlarge table when
n_entries gets
                                                * to be this large.
                                                */
    int lo_rebuild_size;                        /**< Shrink table when
n_entries gets
                                                * below this.
                                                */
    int down_shift;                            /**< Shift count used in
hashing
                                                * function.  Designed to use
high-
                                                * order bits of randomized
keys.
                                                */
    int mask;                                  /**< Mask value used in hashing
                                                * function.
                                                */
    DBusHashType key_type;                     /**< Type of keys used in this
table */

    DBusFindEntryFunction find_function; /**< Function for finding
entries */

```

```

    DBusFreeFunction free_key_function;    /**< Function to free keys */
    DBusFreeFunction free_value_function; /**< Function to free values
*/

    DBusMemPool *entry_pool;              /**< Memory pool for hash
entries */
};

/**
 * @brief Internals of DBusHashIter.
 */
typedef struct
{
    DBusHashTable *table;                 /**< Pointer to table containing entry. */
    DBusHashEntry **bucket;              /**< Pointer to bucket that points to
 * first entry in this entry's chain:
 * used for deleting the entry.
 */
    DBusHashEntry *entry;                 /**< Current hash entry */
    DBusHashEntry *next_entry;           /**< Next entry to be iterated onto in
current bucket */
    int next_bucket;                      /**< index of next bucket */
    int n_entries_on_init;                /**< used to detect table resize since
initialization */
} DBusRealHashIter;

_DBUS_STATIC_ASSERT (sizeof (DBusRealHashIter) == sizeof
(DBusHashIter));

static DBusHashEntry* find_direct_function    (DBusHashTable
*table,
                                              void
*key,
                                              dbus_bool_t
create_if_not_found,
                                              DBusHashEntry
***bucket,
                                              DBusPreallocatedHash
*preallocated);
static DBusHashEntry* find_string_function   (DBusHashTable
*table,
                                              void
*key,
                                              dbus_bool_t
create_if_not_found,
                                              DBusHashEntry
***bucket,
                                              DBusPreallocatedHash
*preallocated);
static unsigned int   string_hash           (const char
*str);

```

```

static void          rebuild_table          (DBusHashTable
*table);
static DBusHashEntry* alloc_entry          (DBusHashTable
*table);
static void          remove_entry          (DBusHashTable
*table,
                                          DBusHashEntry
**bucket,
                                          DBusHashEntry
*entry);
static void          free_entry             (DBusHashTable
*table,
                                          DBusHashEntry
*entry);
static void          free_entry_data       (DBusHashTable
*table,
                                          DBusHashEntry
*entry);

/** @} */

/**
 * @addtogroup DBusHashTable
 * @{
 */

/**
 * @typedef DBusHashIter
 *
 * Public opaque hash table iterator object.
 */

/**
 * @typedef DBusHashTable
 *
 * Public opaque hash table object.
 */

/**
 * @typedef DBusHashType
 *
 * Indicates the type of a key in the hash table.
 */

/**
 * Constructs a new hash table. Should be freed with
 * _dbus_hash_table_unref(). If memory cannot be
 * allocated for the hash table, returns #NULL.
 *
 * @param type the type of hash key to use.
 * @param key_free_function function to free hash keys.

```

```

* @param value_free_function function to free hash values.
* @returns a new DBusHashTable or #NULL if no memory.
*/
DBusHashTable*
_dbus_hash_table_new (DBusHashType      type,
                     DBusFreeFunction key_free_function,
                     DBusFreeFunction value_free_function)
{
    DBusHashTable *table;
    DBusMemPool *entry_pool;

    table = dbus_new0 (DBusHashTable, 1);
    if (table == NULL)
        return NULL;

    entry_pool = _dbus_mem_pool_new (sizeof (DBusHashEntry), TRUE);
    if (entry_pool == NULL)
    {
        dbus_free (table);
        return NULL;
    }

    table->refcount = 1;
    table->entry_pool = entry_pool;

    _dbus_assert (DBUS_SMALL_HASH_TABLE == _DBUS_N_ELEMENTS (table->static_buckets));

    table->buckets = table->static_buckets;
    table->n_buckets = DBUS_SMALL_HASH_TABLE;
    table->n_entries = 0;
    table->hi_rebuild_size = DBUS_SMALL_HASH_TABLE * REBUILD_MULTIPLIER;
    table->lo_rebuild_size = 0;
    table->down_shift = 28;
    table->mask = 3;
    table->key_type = type;

    _dbus_assert (table->mask < table->n_buckets);

    switch (table->key_type)
    {
        case DBUS_HASH_INT:
        case DBUS_HASH_UINTPTR:
            table->find_function = find_direct_function;
            break;
        case DBUS_HASH_STRING:
            table->find_function = find_string_function;
            break;
        default:
            _dbus_assert_not_reached ("Unknown hash table type");
            break;
    }
}

```

```

    table->free_key_function = key_free_function;
    table->free_value_function = value_free_function;

    return table;
}

/**
 * Increments the reference count for a hash table.
 *
 * @param table the hash table to add a reference to.
 * @returns the hash table.
 */
DBusHashTable *
_dbus_hash_table_ref (DBusHashTable *table)
{
    table->refcount += 1;

    return table;
}

/**
 * Decrements the reference count for a hash table,
 * freeing the hash table if the count reaches zero.
 *
 * @param table the hash table to remove a reference from.
 */
void
_dbus_hash_table_unref (DBusHashTable *table)
{
    table->refcount -= 1;

    if (table->refcount == 0)
    {
#ifdef 0
        DBusHashEntry *entry;
        DBusHashEntry *next;
        int i;

        /* Free the entries in the table. */
        for (i = 0; i < table->n_buckets; i++)
        {
            entry = table->buckets[i];
            while (entry != NULL)
            {
                next = entry->next;

                free_entry (table, entry);

                entry = next;
            }

```

```

    }
#else
    DBusHashEntry *entry;
    int i;

    /* Free the entries in the table. */
    for (i = 0; i < table->n_buckets; i++)
    {
        entry = table->buckets[i];
        while (entry != NULL)
        {
            free_entry_data (table, entry);

            entry = entry->next;
        }
    }
    /* We can do this very quickly with memory pools ;- ) */
    _dbus_mem_pool_free (table->entry_pool);
#endif

    /* Free the bucket array, if it was dynamically allocated. */
    if (table->buckets != table->static_buckets)
        dbus_free (table->buckets);

    dbus_free (table);
}

/**
 * Removed all entries from a hash table.
 *
 * @param table the hash table to remove all entries from.
 */
void
_dbus_hash_table_remove_all (DBusHashTable *table)
{
    DBusHashIter iter;
    _dbus_hash_iter_init (table, &iter);
    while (_dbus_hash_iter_next (&iter))
    {
        _dbus_hash_iter_remove_entry(&iter);
    }
}

static DBusHashEntry*
alloc_entry (DBusHashTable *table)
{
    DBusHashEntry *entry;

    entry = _dbus_mem_pool_alloc (table->entry_pool);

    return entry;
}

```



```

}

static void
free_entry_data (DBusHashTable *table,
                DBusHashEntry *entry)
{
    if (table->free_key_function)
        (* table->free_key_function) (entry->key);
    if (table->free_value_function)
        (* table->free_value_function) (entry->value);
}

static void
free_entry (DBusHashTable *table,
            DBusHashEntry *entry)
{
    free_entry_data (table, entry);
    _dbus_mem_pool_dealloc (table->entry_pool, entry);
}

static void
remove_entry (DBusHashTable *table,
              DBusHashEntry **bucket,
              DBusHashEntry *entry)
{
    _dbus_assert (table != NULL);
    _dbus_assert (bucket != NULL);
    _dbus_assert (*bucket != NULL);
    _dbus_assert (entry != NULL);

    if (*bucket == entry)
        *bucket = entry->next;
    else
        {
            DBusHashEntry *prev;
            prev = *bucket;

            while (prev->next != entry)
                prev = prev->next;

            _dbus_assert (prev != NULL);

            prev->next = entry->next;
        }

    table->n_entries -= 1;
    free_entry (table, entry);
}

/**
 * Initializes a hash table iterator. To iterate over all entries in a
 * hash table, use the following code (the printf assumes a hash

```

```

* from strings to strings obviously):
*
* @code
* DBusHashIter iter;
*
* _dbus_hash_iter_init (table, &iter);
* while (_dbus_hash_iter_next (&iter))
* {
*     printf ("The first key is %s and value is %s\n",
*             _dbus_hash_iter_get_string_key (&iter),
*             _dbus_hash_iter_get_value (&iter));
* }
*
* @endcode
*
* The iterator is initialized pointing "one before" the first hash
* entry. The first call to _dbus_hash_iter_next() moves it onto
* the first valid entry or returns #FALSE if the hash table is
* empty. Subsequent calls move to the next valid entry or return
* #FALSE if there are no more entries.
*
* Note that it is guaranteed to be safe to remove a hash entry during
* iteration, but it is not safe to add a hash entry.
*
* @param table the hash table to iterate over.
* @param iter the iterator to initialize.
*/
void
_dbus_hash_iter_init (DBusHashTable *table,
                    DBusHashIter *iter)
{
    DBusRealHashIter *real;

    _dbus_assert (sizeof (DBusHashIter) == sizeof (DBusRealHashIter));

    real = (DBusRealHashIter*) iter;

    real->table = table;
    real->bucket = NULL;
    real->entry = NULL;
    real->next_entry = NULL;
    real->next_bucket = 0;
    real->n_entries_on_init = table->n_entries;
}

/**
* Move the hash iterator forward one step, to the next hash entry.
* The documentation for _dbus_hash_iter_init() explains in more
* detail.
*
* @param iter the iterator to move forward.

```

```

* @returns #FALSE if there are no more entries to move to.
*/
dbus_bool_t
_dbus_hash_iter_next (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    _dbus_assert (sizeof (DBusHashIter) == sizeof (DBusRealHashIter));

    real = (DBusRealHashIter*) iter;

    /* if this assertion failed someone probably added hash entries
     * during iteration, which is bad.
     */
    _dbus_assert (real->n_entries_on_init >= real->table->n_entries);

    /* Remember that real->entry may have been deleted */

    while (real->next_entry == NULL)
    {
        if (real->next_bucket >= real->table->n_buckets)
        {
            /* invalidate iter and return false */
            real->entry = NULL;
            real->table = NULL;
            real->bucket = NULL;
            return FALSE;
        }

        real->bucket = &(real->table->buckets[real->next_bucket]);
        real->next_entry = *(real->bucket);
        real->next_bucket += 1;
    }

    _dbus_assert (real->next_entry != NULL);
    _dbus_assert (real->bucket != NULL);

    real->entry = real->next_entry;
    real->next_entry = real->entry->next;

    return TRUE;
}

/**
 * Removes the current entry from the hash table.
 * If a key_free_function or value_free_function
 * was provided to _dbus_hash_table_new(),
 * frees the key and/or value for this entry.
 *
 * @param iter the hash table iterator.
 */
void

```

```

_dbus_hash_iter_remove_entry (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);
    _dbus_assert (real->entry != NULL);
    _dbus_assert (real->bucket != NULL);

    remove_entry (real->table, real->bucket, real->entry);

    real->entry = NULL; /* make it crash if you try to use this entry */
}

/**
 * Gets the value of the current entry.
 *
 * @param iter the hash table iterator.
 */
void*
_dbus_hash_iter_get_value (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);
    _dbus_assert (real->entry != NULL);

    return real->entry->value;
}

/**
 * Sets the value of the current entry.
 * If the hash table has a value_free_function
 * it will be used to free the previous value.
 * The hash table will own the passed-in value
 * (it will not be copied).
 *
 * @param iter the hash table iterator.
 * @param value the new value.
 */
void
_dbus_hash_iter_set_value (DBusHashIter *iter,
                          void *value)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);

```

```

    _dbus_assert (real->entry != NULL);

    if (real->table->free_value_function && value != real->entry->value)
        (* real->table->free_value_function) (real->entry->value);

    real->entry->value = value;
}

/**
 * Gets the key for the current entry.
 * Only works for hash tables of type #DBUS_HASH_INT.
 *
 * @param iter the hash table iterator.
 */
int
_dbus_hash_iter_get_int_key (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);
    _dbus_assert (real->entry != NULL);

    return _DBUS_POINTER_TO_INT (real->entry->key);
}

/**
 * Gets the key for the current entry.
 * Only works for hash tables of type #DBUS_HASH_UINTPTR.
 *
 * @param iter the hash table iterator.
 */
uintptr_t
_dbus_hash_iter_get_uintptr_key (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);
    _dbus_assert (real->entry != NULL);

    return (uintptr_t) real->entry->key;
}

/**
 * Gets the key for the current entry.
 * Only works for hash tables of type #DBUS_HASH_STRING
 * @param iter the hash table iterator.
 */
const char*

```

```

_dbus_hash_iter_get_string_key (DBusHashIter *iter)
{
    DBusRealHashIter *real;

    real = (DBusRealHashIter*) iter;

    _dbus_assert (real->table != NULL);
    _dbus_assert (real->entry != NULL);

    return real->entry->key;
}

/**
 * A low-level but efficient interface for manipulating the hash
 * table. It's efficient because you can get, set, and optionally
 * create the hash entry while only running the hash function one
 * time.
 *
 * Note that while calling _dbus_hash_iter_next() on the iterator
 * filled in by this function may work, it's completely
 * undefined which entries are after this iter and which
 * are before it. So it would be silly to iterate using this
 * iterator.
 *
 * If the hash entry is created, its value will be initialized
 * to all bits zero.
 *
 * #FALSE may be returned due to memory allocation failure, or
 * because create_if_not_found was #FALSE and the entry
 * did not exist.
 *
 * If create_if_not_found is #TRUE and the entry is created, the hash
 * table takes ownership of the key that's passed in.
 *
 * For a hash table of type #DBUS_HASH_INT, cast the int
 * key to the key parameter using #_DBUS_INT_TO_POINTER().
 *
 * @param table the hash table.
 * @param key the hash key.
 * @param create_if_not_found if #TRUE, create the entry if it didn't
 * exist.
 * @param iter the iterator to initialize.
 * @returns #TRUE if the hash entry now exists (and the iterator is
 * thus valid).
 */
dbus_bool_t
_dbus_hash_iter_lookup (DBusHashTable *table,
                       void *key,
                       dbus_bool_t create_if_not_found,
                       DBusHashIter *iter)
{
    DBusRealHashIter *real;

```

```

DBusHashEntry *entry;
DBusHashEntry **bucket;

_dbus_assert (sizeof (DBusHashIter) == sizeof (DBusRealHashIter));

real = (DBusRealHashIter*) iter;

entry = (* table->find_function) (table, key, create_if_not_found,
&bucket, NULL);

if (entry == NULL)
    return FALSE;

real->table = table;
real->bucket = bucket;
real->entry = entry;
real->next_entry = entry->next;
real->next_bucket = (bucket - table->buckets) + 1;
real->n_entries_on_init = table->n_entries;

_dbus_assert (&(table->buckets[real->next_bucket-1]) == real->
bucket);

return TRUE;
}

static void
add_allocated_entry (DBusHashTable *table,
                    DBusHashEntry *entry,
                    unsigned int idx,
                    void *key,
                    DBusHashEntry ***bucket)
{
    DBusHashEntry **b;

    entry->key = key;

    b = &(table->buckets[idx]);
    entry->next = *b;
    *b = entry;

    if (bucket)
        *bucket = b;

    table->n_entries += 1;

    /* note we ONLY rebuild when ADDING - because you can iterate over a
     * table and remove entries safely.
     */
    if (table->n_entries >= table->hi_rebuild_size ||
        table->n_entries < table->lo_rebuild_size)
        rebuild_table (table);
}

```

```

}

static DBusHashEntry*
add_entry (DBusHashTable      *table,
           unsigned int      idx,
           void               *key,
           DBusHashEntry     ***bucket,
           DBusPreallocatedHash *preallocated)
{
    DBusHashEntry *entry;

    if (preallocated == NULL)
    {
        entry = alloc_entry (table);
        if (entry == NULL)
        {
            if (bucket)
                *bucket = NULL;
            return NULL;
        }
    }
    else
    {
        entry = (DBusHashEntry*) preallocated;
    }

    add_allocated_entry (table, entry, idx, key, bucket);

    return entry;
}

/* This is g_str_hash from GLib which was
 * extensively discussed/tested/profiled
 */
static unsigned int
string_hash (const char *str)
{
    const char *p = str;
    unsigned int h = *p;

    if (h)
        for (p += 1; *p != '\0'; p++)
            h = (h << 5) - h + *p;

    return h;
}

/** Key comparison function */
typedef int (* KeyCompareFunc) (const void *key_a, const void *key_b);

static DBusHashEntry*
find_generic_function (DBusHashTable      *table,

```



```

        void                *key,
        unsigned int        idx,
        KeyCompareFunc      compare_func,
        dbus_bool_t        create_if_not_found,
        DBusHashEntry      ***bucket,
        DBusPreallocatedHash *preallocated)
{
    DBusHashEntry *entry;

    if (bucket)
        *bucket = NULL;

    /* Search all of the entries in this bucket. */
    entry = table->buckets[idx];
    while (entry != NULL)
    {
        if ((compare_func == NULL && key == entry->key) ||
            (compare_func != NULL && (* compare_func) (key, entry->key)
== 0))
        {
            if (bucket)
                *bucket = &(table->buckets[idx]);

            if (preallocated)
                _dbus_hash_table_free_preallocated_entry (table,
preallocated);

            return entry;
        }

        entry = entry->next;
    }

    if (create_if_not_found)
        entry = add_entry (table, idx, key, bucket, preallocated);
    else if (preallocated)
        _dbus_hash_table_free_preallocated_entry (table, preallocated);

    return entry;
}

static DBusHashEntry*
find_string_function (DBusHashTable      *table,
                    void                *key,
                    dbus_bool_t        create_if_not_found,
                    DBusHashEntry      ***bucket,
                    DBusPreallocatedHash *preallocated)
{
    unsigned int idx;

    idx = string_hash (key) & table->mask;

```

```

    return find_generic_function (table, key, idx,
                                  (KeyCompareFunc) strcmp,
create_if_not_found, bucket,
                                  preallocated);
}

static DBusHashEntry*
find_direct_function (DBusHashTable      *table,
                     void                *key,
                     dbus_bool_t        create_if_not_found,
                     DBusHashEntry     ***bucket,
                     DBusPreallocatedHash *preallocated)
{
    unsigned int idx;

    idx = RANDOM_INDEX (table, key) & table->mask;

    return find_generic_function (table, key, idx,
                                  NULL, create_if_not_found, bucket,
                                  preallocated);
}

static void
rebuild_table (DBusHashTable *table)
{
    int old_size;
    int new_buckets;
    DBusHashEntry **old_buckets;
    DBusHashEntry **old_chain;
    DBusHashEntry *entry;
    dbus_bool_t growing;

    /*
     * Allocate and initialize the new bucket array, and set up
     * hashing constants for new array size.
     */

    growing = table->n_entries >= table->hi_rebuild_size;

    old_size = table->n_buckets;
    old_buckets = table->buckets;

    if (growing)
    {
        /* overflow paranoia */
        if (table->n_buckets < _DBUS_INT_MAX / 4 &&
            table->down_shift >= 0)
            new_buckets = table->n_buckets * 4;
        else
            return; /* can't grow anymore */
    }
}

```

```

else
{
    new_buckets = table->n_buckets / 4;
    if (new_buckets < DBUS_SMALL_HASH_TABLE)
        return; /* don't bother shrinking this far */
}

table->buckets = dbus_new0 (DBusHashEntry*, new_buckets);
if (table->buckets == NULL)
{
    /* out of memory, yay - just don't reallocate, the table will
    * still work, albeit more slowly.
    */
    table->buckets = old_buckets;
    return;
}

table->n_buckets = new_buckets;

if (growing)
{
    table->lo_rebuild_size = table->hi_rebuild_size;
    table->hi_rebuild_size *= 4;

    table->down_shift -= 2; /* keep 2 more high bits
*/
    table->mask = (table->mask << 2) + 3; /* keep 2 more high bits
*/
}
else
{
    table->hi_rebuild_size = table->lo_rebuild_size;
    table->lo_rebuild_size /= 4;

    table->down_shift += 2; /* keep 2 fewer high bits */
    table->mask = table->mask >> 2; /* keep 2 fewer high bits */
}

#if 0
printf ("%s table to lo = %d hi = %d downshift = %d mask = 0x%x\n",
        growing ? "GROW" : "SHRINK",
        table->lo_rebuild_size,
        table->hi_rebuild_size,
        table->down_shift,
        table->mask);
#endif

_dbus_assert (table->lo_rebuild_size >= 0);
_dbus_assert (table->hi_rebuild_size > table->lo_rebuild_size);
_dbus_assert (table->mask != 0);
/* the mask is essentially the max index */
_dbus_assert (table->mask < table->n_buckets);

```

```

/*
 * Rehash all of the existing entries into the new bucket array.
 */

for (old_chain = old_buckets; old_size > 0; old_size--, old_chain++)
{
    for (entry = *old_chain; entry != NULL; entry = *old_chain)
    {
        unsigned int idx;
        DBusHashEntry **bucket;

        *old_chain = entry->next;
        switch (table->key_type)
        {
            case DBUS_HASH_STRING:
                idx = string_hash (entry->key) & table->mask;
                break;
            case DBUS_HASH_INT:
            case DBUS_HASH_UINTPTR:
                idx = RANDOM_INDEX (table, entry->key);
                break;
            default:
                idx = 0;
                _dbus_assert_not_reached ("Unknown hash table type");
                break;
        }

        bucket = &(table->buckets[idx]);
        entry->next = *bucket;
        *bucket = entry;
    }
}

/* Free the old bucket array, if it was dynamically allocated. */

if (old_buckets != table->static_buckets)
    dbus_free (old_buckets);
}

/**
 * Looks up the value for a given string in a hash table
 * of type #DBUS_HASH_STRING. Returns %NULL if the value
 * is not present. (A not-present entry is indistinguishable
 * from an entry with a value of %NULL.)
 * @param table the hash table.
 * @param key the string to look up.
 * @returns the value of the hash entry.
 */
void*
_dbus_hash_table_lookup_string (DBusHashTable *table,
                               const char *key)

```

```

{
    DBusHashEntry *entry;

    _dbus_assert (table->key_type == DBUS_HASH_STRING);

    entry = (* table->find_function) (table, (char*) key, FALSE, NULL,
    NULL);

    if (entry)
        return entry->value;
    else
        return NULL;
}

/**
 * Looks up the value for a given integer in a hash table
 * of type #DBUS_HASH_INT. Returns %NULL if the value
 * is not present. (A not-present entry is indistinguishable
 * from an entry with a value of %NULL.)
 * @param table the hash table.
 * @param key the integer to look up.
 * @returns the value of the hash entry.
 */
void*
_dbus_hash_table_lookup_int (DBusHashTable *table,
                             int          key)
{
    DBusHashEntry *entry;

    _dbus_assert (table->key_type == DBUS_HASH_INT);

    entry = (* table->find_function) (table, _DBUS_INT_TO_POINTER (key),
    FALSE, NULL, NULL);

    if (entry)
        return entry->value;
    else
        return NULL;
}

/**
 * Looks up the value for a given integer in a hash table
 * of type #DBUS_HASH_UINTPTR. Returns %NULL if the value
 * is not present. (A not-present entry is indistinguishable
 * from an entry with a value of %NULL.)
 * @param table the hash table.
 * @param key the integer to look up.
 * @returns the value of the hash entry.
 */
void*
_dbus_hash_table_lookup_uintptr (DBusHashTable *table,
                                 uintptr_t     key)

```

```

{
    DBusHashEntry *entry;

    _dbus_assert (table->key_type == DBUS_HASH_UINTPTR);

    entry = (* table->find_function) (table, (void*) key, FALSE, NULL,
    NULL);

    if (entry)
        return entry->value;
    else
        return NULL;
}

/**
 * Removes the hash entry for the given key. If no hash entry
 * for the key exists, does nothing.
 *
 * @param table the hash table.
 * @param key the hash key.
 * @returns #TRUE if the entry existed
 */
dbus_bool_t
_dbus_hash_table_remove_string (DBusHashTable *table,
                                const char *key)
{
    DBusHashEntry *entry;
    DBusHashEntry **bucket;

    _dbus_assert (table->key_type == DBUS_HASH_STRING);

    entry = (* table->find_function) (table, (char*) key, FALSE,
    &bucket, NULL);

    if (entry)
    {
        remove_entry (table, bucket, entry);
        return TRUE;
    }
    else
        return FALSE;
}

/**
 * Removes the hash entry for the given key. If no hash entry
 * for the key exists, does nothing.
 *
 * @param table the hash table.
 * @param key the hash key.
 * @returns #TRUE if the entry existed
 */
dbus_bool_t

```

```

_dbus_hash_table_remove_int (DBusHashTable *table,
                             int            key)
{
    DBusHashEntry *entry;
    DBusHashEntry **bucket;

    _dbus_assert (table->key_type == DBUS_HASH_INT);

    entry = (* table->find_function) (table, _DBUS_INT_TO_POINTER (key),
    FALSE, &bucket, NULL);

    if (entry)
        {
            remove_entry (table, bucket, entry);
            return TRUE;
        }
    else
        return FALSE;
}

/**
 * Removes the hash entry for the given key. If no hash entry
 * for the key exists, does nothing.
 *
 * @param table the hash table.
 * @param key the hash key.
 * @returns #TRUE if the entry existed
 */
dbus_bool_t
_dbus_hash_table_remove_uintptr (DBusHashTable *table,
                                uintptr_t      key)
{
    DBusHashEntry *entry;
    DBusHashEntry **bucket;

    _dbus_assert (table->key_type == DBUS_HASH_UINTPTR);

    entry = (* table->find_function) (table, (void*) key, FALSE,
    &bucket, NULL);

    if (entry)
        {
            remove_entry (table, bucket, entry);
            return TRUE;
        }
    else
        return FALSE;
}

/**
 * Creates a hash entry with the given key and value.
 * The key and value are not copied; they are stored

```

```

* in the hash table by reference. If an entry with the
* given key already exists, the previous key and value
* are overwritten (and freed if the hash table has
* a key_free_function and/or value_free_function).
*
* Returns #FALSE if memory for the new hash entry
* can't be allocated.
*
* @param table the hash table.
* @param key the hash entry key.
* @param value the hash entry value.
*/
dbus_bool_t
_dbus_hash_table_insert_string (DBusHashTable *table,
                                char          *key,
                                void          *value)
{
    DBusPreallocatedHash *preallocated;

    _dbus_assert (table->key_type == DBUS_HASH_STRING);

    preallocated = _dbus_hash_table_preallocate_entry (table);
    if (preallocated == NULL)
        return FALSE;

    _dbus_hash_table_insert_string_preallocated (table, preallocated,
                                                key, value);

    return TRUE;
}

/**
* Creates a hash entry with the given key and value.
* The key and value are not copied; they are stored
* in the hash table by reference. If an entry with the
* given key already exists, the previous key and value
* are overwritten (and freed if the hash table has
* a key_free_function and/or value_free_function).
*
* Returns #FALSE if memory for the new hash entry
* can't be allocated.
*
* @param table the hash table.
* @param key the hash entry key.
* @param value the hash entry value.
*/
dbus_bool_t
_dbus_hash_table_insert_int (DBusHashTable *table,
                             int          key,
                             void          *value)
{
    DBusHashEntry *entry;

```



```

    _dbus_assert (table->key_type == DBUS_HASH_INT);

    entry = (* table->find_function) (table, _DBUS_INT_TO_POINTER (key),
TRUE, NULL, NULL);

    if (entry == NULL)
        return FALSE; /* no memory */

    if (table->free_key_function && entry->key != _DBUS_INT_TO_POINTER
(key))
        (* table->free_key_function) (entry->key);

    if (table->free_value_function && entry->value != value)
        (* table->free_value_function) (entry->value);

    entry->key = _DBUS_INT_TO_POINTER (key);
    entry->value = value;

    return TRUE;
}

/**
 * Creates a hash entry with the given key and value.
 * The key and value are not copied; they are stored
 * in the hash table by reference. If an entry with the
 * given key already exists, the previous key and value
 * are overwritten (and freed if the hash table has
 * a key_free_function and/or value_free_function).
 *
 * Returns #FALSE if memory for the new hash entry
 * can't be allocated.
 *
 * @param table the hash table.
 * @param key the hash entry key.
 * @param value the hash entry value.
 */
dbus_bool_t
_dbus_hash_table_insert_uintptr (DBusHashTable *table,
                                uintptr_t      key,
                                void           *value)
{
    DBusHashEntry *entry;

    _dbus_assert (table->key_type == DBUS_HASH_UINTPTR);

    entry = (* table->find_function) (table, (void*) key, TRUE, NULL,
NULL);

    if (entry == NULL)
        return FALSE; /* no memory */

```

```

    if (table->free_key_function && entry->key != (void*) key)
        (* table->free_key_function) (entry->key);

    if (table->free_value_function && entry->value != value)
        (* table->free_value_function) (entry->value);

    entry->key = (void*) key;
    entry->value = value;

    return TRUE;
}

/**
 * Preallocate an opaque data blob that allows us to insert into the
 * hash table at a later time without allocating any memory.
 *
 * @param table the hash table
 * @returns the preallocated data, or #NULL if no memory
 */
DBusPreallocatedHash*
_dbus_hash_table_preallocate_entry (DBusHashTable *table)
{
    DBusHashEntry *entry;

    entry = alloc_entry (table);

    return (DBusPreallocatedHash*) entry;
}

/**
 * Frees an opaque DBusPreallocatedHash that was *not* used
 * in order to insert into the hash table.
 *
 * @param table the hash table
 * @param preallocated the preallocated data
 */
void
_dbus_hash_table_free_preallocated_entry (DBusHashTable *table,
                                          DBusPreallocatedHash
                                          *preallocated)
{
    DBusHashEntry *entry;

    _dbus_assert (preallocated != NULL);

    entry = (DBusHashEntry*) preallocated;

    /* Don't use free_entry(), since this entry has no key/data */
    _dbus_mem_pool_dealloc (table->entry_pool, entry);
}

/**

```

```

* Inserts a string-keyed entry into the hash table, using a
* preallocated data block from
* _dbus_hash_table_preallocate_entry(). This function cannot fail due
* to lack of memory. The DBusPreallocatedHash object is consumed and
* should not be reused or freed. Otherwise this function works
* just like _dbus_hash_table_insert_string().
*
* @param table the hash table
* @param preallocated the preallocated data
* @param key the hash key
* @param value the value
*/
void
_dbus_hash_table_insert_string_preallocated (DBusHashTable
*table,
                                           DBusPreallocatedHash
*preallocated,
                                           char
*key,
                                           void
*value)
{
    DBusHashEntry *entry;

    _dbus_assert (table->key_type == DBUS_HASH_STRING);
    _dbus_assert (preallocated != NULL);

    entry = (* table->find_function) (table, key, TRUE, NULL,
preallocated);

    _dbus_assert (entry != NULL);

    if (table->free_key_function && entry->key != key)
        (* table->free_key_function) (entry->key);

    if (table->free_value_function && entry->value != value)
        (* table->free_value_function) (entry->value);

    entry->key = key;
    entry->value = value;
}

/**
* Gets the number of hash entries in a hash table.
*
* @param table the hash table.
* @returns the number of entries in the table.
*/
int
_dbus_hash_table_get_n_entries (DBusHashTable *table)
{
    return table->n_entries;
}

```

```

}

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

/* If you're wondering why the hash table test takes
 * forever to run, it's because we call this function
 * in inner loops thus making things quadratic.
 */
static int
count_entries (DBusHashTable *table)
{
    DBusHashIter iter;
    int count;

    count = 0;
    _dbus_hash_iter_init (table, &iter);
    while (_dbus_hash_iter_next (&iter))
        ++count;

    _dbus_assert (count == _dbus_hash_table_get_n_entries (table));

    return count;
}

/**
 * @ingroup DBusHashTableInternals
 * Unit test for DBusHashTable
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_hash_test (void)
{
    int i;
    DBusHashTable *table1;
    DBusHashTable *table2;
    DBusHashTable *table3;
    DBusHashIter iter;
#define N_HASH_KEYS 5000
    char **keys;
    dbus_bool_t ret = FALSE;

    keys = dbus_new (char *, N_HASH_KEYS);
    if (keys == NULL)
        _dbus_assert_not_reached ("no memory");

    for (i = 0; i < N_HASH_KEYS; i++)
    {
        keys[i] = dbus_malloc (128);

```

```

        if (keys[i] == NULL)
            _dbus_assert_not_reached ("no memory");
    }

printf ("Computing test hash keys...\n");
i = 0;
while (i < N_HASH_KEYS)
    {
        int len;

        len = sprintf (keys[i], "Hash key %d", i);
        _dbus_assert (*(keys[i] + len) == '\0');
        ++i;
    }
printf ("... done.\n");

table1 = _dbus_hash_table_new (DBUS_HASH_STRING,
                               dbus_free, dbus_free);
if (table1 == NULL)
    goto out;

table2 = _dbus_hash_table_new (DBUS_HASH_INT,
                               NULL, dbus_free);
if (table2 == NULL)
    goto out;

table3 = _dbus_hash_table_new (DBUS_HASH_UINTPTR,
                               NULL, dbus_free);
if (table3 == NULL)
    goto out;

/* Insert and remove a bunch of stuff, counting the table in between
 * to be sure it's not broken and that iteration works
 */
i = 0;
while (i < 3000)
    {
        void *value;
        char *key;

        key = _dbus_strdup (keys[i]);
        if (key == NULL)
            goto out;
        value = _dbus_strdup ("Value!");
        if (value == NULL)
            goto out;

        if (!_dbus_hash_table_insert_string (table1,
                                             key, value))
            goto out;
    }

```

```

value = _dbus_strdup (keys[i]);
if (value == NULL)
    goto out;

if (!_dbus_hash_table_insert_int (table2,
                                i, value))
    goto out;

value = _dbus_strdup (keys[i]);
if (value == NULL)
    goto out;

if (!_dbus_hash_table_insert_uintptr (table3,
                                     i, value))
    goto out;

_dbus_assert (count_entries (table1) == i + 1);
_dbus_assert (count_entries (table2) == i + 1);
_dbus_assert (count_entries (table3) == i + 1);

value = _dbus_hash_table_lookup_string (table1, keys[i]);
_dbus_assert (value != NULL);
_dbus_assert (strcmp (value, "Value!") == 0);

value = _dbus_hash_table_lookup_int (table2, i);
_dbus_assert (value != NULL);
_dbus_assert (strcmp (value, keys[i]) == 0);

value = _dbus_hash_table_lookup_uintptr (table3, i);
_dbus_assert (value != NULL);
_dbus_assert (strcmp (value, keys[i]) == 0);

++i;
}

--i;
while (i >= 0)
{
    _dbus_hash_table_remove_string (table1,
                                   keys[i]);

    _dbus_hash_table_remove_int (table2, i);

    _dbus_hash_table_remove_uintptr (table3, i);

    _dbus_assert (count_entries (table1) == i);
    _dbus_assert (count_entries (table2) == i);
    _dbus_assert (count_entries (table3) == i);

    --i;
}

```

```

_dbus_hash_table_ref (table1);
_dbus_hash_table_ref (table2);
_dbus_hash_table_ref (table3);
_dbus_hash_table_unref (table1);
_dbus_hash_table_unref (table2);
_dbus_hash_table_unref (table3);
_dbus_hash_table_unref (table1);
_dbus_hash_table_unref (table2);
_dbus_hash_table_unref (table3);
table3 = NULL;

/* Insert a bunch of stuff then check
 * that iteration works correctly (finds the right
 * values, iter_set_value works, etc.)
 */
table1 = _dbus_hash_table_new (DBUS_HASH_STRING,
                              dbus_free, dbus_free);
if (table1 == NULL)
    goto out;

table2 = _dbus_hash_table_new (DBUS_HASH_INT,
                              NULL, dbus_free);
if (table2 == NULL)
    goto out;

i = 0;
while (i < 5000)
{
    char *key;
    void *value;

    key = _dbus_strdup (keys[i]);
    if (key == NULL)
        goto out;
    value = _dbus_strdup ("Value!");
    if (value == NULL)
        goto out;

    if (!_dbus_hash_table_insert_string (table1,
                                       key, value))
        goto out;

    value = _dbus_strdup (keys[i]);
    if (value == NULL)
        goto out;

    if (!_dbus_hash_table_insert_int (table2,
                                    i, value))
        goto out;

    _dbus_assert (count_entries (table1) == i + 1);
    _dbus_assert (count_entries (table2) == i + 1);
}

```

```

    ++i;
}

_dbus_hash_iter_init (table1, &iter);
while (_dbus_hash_iter_next (&iter))
{
    const char *key;
    void *value;

    key = _dbus_hash_iter_get_string_key (&iter);
    value = _dbus_hash_iter_get_value (&iter);

    _dbus_assert (_dbus_hash_table_lookup_string (table1, key) ==
value);

    value = _dbus_strdup ("Different value!");
    if (value == NULL)
        goto out;

    _dbus_hash_iter_set_value (&iter, value);

    _dbus_assert (_dbus_hash_table_lookup_string (table1, key) ==
value);
}

_dbus_hash_iter_init (table1, &iter);
while (_dbus_hash_iter_next (&iter))
{
    _dbus_hash_iter_remove_entry (&iter);
    _dbus_assert (count_entries (table1) == i - 1);
    --i;
}

_dbus_hash_iter_init (table2, &iter);
while (_dbus_hash_iter_next (&iter))
{
    int key;
    void *value;

    key = _dbus_hash_iter_get_int_key (&iter);
    value = _dbus_hash_iter_get_value (&iter);

    _dbus_assert (_dbus_hash_table_lookup_int (table2, key) ==
value);

    value = _dbus_strdup ("Different value!");
    if (value == NULL)
        goto out;

    _dbus_hash_iter_set_value (&iter, value);
}

```



```

        _dbus_assert (_dbus_hash_table_lookup_int (table2, key) ==
value);
    }

    i = count_entries (table2);
    _dbus_hash_iter_init (table2, &iter);
    while (_dbus_hash_iter_next (&iter))
    {
        _dbus_hash_iter_remove_entry (&iter);
        _dbus_assert (count_entries (table2) + 1 == i);
        --i;
    }

    /* add/remove interleaved, to check that we grow/shrink the table
    * appropriately
    */
    i = 0;
    while (i < 1000)
    {
        char *key;
        void *value;

        key = _dbus_strdup (keys[i]);
        if (key == NULL)
            goto out;

        value = _dbus_strdup ("Value!");
        if (value == NULL)
            goto out;

        if (!_dbus_hash_table_insert_string (table1,
                                            key, value))
            goto out;

        ++i;
    }

    --i;
    while (i >= 0)
    {
        char *key;
        void *value;

        key = _dbus_strdup (keys[i]);
        if (key == NULL)
            goto out;
        value = _dbus_strdup ("Value!");
        if (value == NULL)
            goto out;

        if (!_dbus_hash_table_remove_string (table1, keys[i]))
            goto out;
    }

```

```

    if (!_dbus_hash_table_insert_string (table1,
                                        key, value))
        goto out;

    if (!_dbus_hash_table_remove_string (table1, keys[i]))
        goto out;

    _dbus_assert (_dbus_hash_table_get_n_entries (table1) == i);

    --i;
}

/* nuke these tables */
_dbus_hash_table_unref (table1);
_dbus_hash_table_unref (table2);

/* Now do a bunch of things again using _dbus_hash_iter_lookup() to
 * be sure that interface works.
 */
table1 = _dbus_hash_table_new (DBUS_HASH_STRING,
                              dbus_free, dbus_free);
if (table1 == NULL)
    goto out;

table2 = _dbus_hash_table_new (DBUS_HASH_INT,
                              NULL, dbus_free);
if (table2 == NULL)
    goto out;

i = 0;
while (i < 3000)
{
    void *value;
    char *key;

    key = _dbus_strdup (keys[i]);
    if (key == NULL)
        goto out;
    value = _dbus_strdup ("Value!");
    if (value == NULL)
        goto out;

    if (!_dbus_hash_iter_lookup (table1,
                                key, TRUE, &iter))
        goto out;
    _dbus_assert (_dbus_hash_iter_get_value (&iter) == NULL);
    _dbus_hash_iter_set_value (&iter, value);

    value = _dbus_strdup (keys[i]);
    if (value == NULL)

```

```

        goto out;

    if (!_dbus_hash_iter_lookup (table2,
                                _DBUS_INT_TO_POINTER (i), TRUE,
&iter))
        goto out;
    _dbus_assert (_dbus_hash_iter_get_value (&iter) == NULL);
    _dbus_hash_iter_set_value (&iter, value);

    _dbus_assert (count_entries (table1) == i + 1);
    _dbus_assert (count_entries (table2) == i + 1);

    if (!_dbus_hash_iter_lookup (table1, keys[i], FALSE, &iter))
        goto out;

    value = _dbus_hash_iter_get_value (&iter);
    _dbus_assert (value != NULL);
    _dbus_assert (strcmp (value, "Value!") == 0);

    /* Iterate just to be sure it works, though
     * it's a stupid thing to do
     */
    while (_dbus_hash_iter_next (&iter))
        ;

    if (!_dbus_hash_iter_lookup (table2, _DBUS_INT_TO_POINTER (i),
FALSE, &iter))
        goto out;

    value = _dbus_hash_iter_get_value (&iter);
    _dbus_assert (value != NULL);
    _dbus_assert (strcmp (value, keys[i]) == 0);

    /* Iterate just to be sure it works, though
     * it's a stupid thing to do
     */
    while (_dbus_hash_iter_next (&iter))
        ;

    ++i;
}

--i;
while (i >= 0)
{
    if (!_dbus_hash_iter_lookup (table1, keys[i], FALSE, &iter))
        _dbus_assert_not_reached ("hash entry should have existed");
    _dbus_hash_iter_remove_entry (&iter);

    if (!_dbus_hash_iter_lookup (table2, _DBUS_INT_TO_POINTER (i),
FALSE, &iter))
        _dbus_assert_not_reached ("hash entry should have existed");
}

```

```

    _dbus_hash_iter_remove_entry (&iter);

    _dbus_assert (count_entries (table1) == i);
    _dbus_assert (count_entries (table2) == i);

    --i;
}

_dbus_hash_table_unref (table1);
_dbus_hash_table_unref (table2);

ret = TRUE;

out:
for (i = 0; i < N_HASH_KEYS; i++)
    dbus_free (keys[i]);

dbus_free (keys);

return ret;
}

#endif /* DBUS_BUILD_TESTS */

File = dbus-hash.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-hash.h Generic hash table utility (internal to D-Bus
implementation)
*
* Copyright (C) 2002 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

```

```

*
*/

#ifndef DBUS_HASH_H
#define DBUS_HASH_H

#ifdef HAVE_STDINT_H
#include <stdint.h>
#endif

#ifdef HAVE_INTTYPES_H
#include <inttypes.h>
#endif

#include <dbus/dbus-memory.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-sysdeps.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusHashTable
 * @{
 */

/** Hash iterator object. The iterator is on the stack, but its real
 * fields are hidden privately.
 */
struct DBusHashIter
{
    void *dummy1; /**< Do not use. */
    void *dummy2; /**< Do not use. */
    void *dummy3; /**< Do not use. */
    void *dummy4; /**< Do not use. */
    int dummy5; /**< Do not use. */
    int dummy6; /**< Do not use. */
};

typedef struct DBusHashTable DBusHashTable;
typedef struct DBusHashIter DBusHashIter;

/* Allowing an arbitrary function as with GLib
 * would be nicer for a public API, but for
 * an internal API this saves typing, we can add
 * more whenever we feel like it.
 */
typedef enum
{
    DBUS_HASH_STRING,          /**< Hash keys are strings. */
    DBUS_HASH_INT,            /**< Hash keys are integers. */
    DBUS_HASH_UINTPTR         /**< Hash keys are integer capable to hold
a pointer. */

```

```

} DBusHashTable;

DBusHashTable* _dbus_hash_table_new          (DBusHashTable
type,                                         DBusFreeFunction
key_free_function,                           DBusFreeFunction
value_free_function);
DBusHashTable* _dbus_hash_table_ref          (DBusHashTable
*table);
void           _dbus_hash_table_unref        (DBusHashTable
*table);
void           _dbus_hash_table_remove_all   (DBusHashTable
*table);
void           _dbus_hash_iter_init          (DBusHashTable
*table,                                         DBusHashIter
*iter);
dbus_bool_t    _dbus_hash_iter_next          (DBusHashIter
*iter);
void           _dbus_hash_iter_remove_entry  (DBusHashIter
*iter);
void*          _dbus_hash_iter_get_value     (DBusHashIter
*iter);
void           _dbus_hash_iter_set_value     (DBusHashIter
*iter,                                         void
*value);
int            _dbus_hash_iter_get_int_key   (DBusHashIter
*iter);
const char*    _dbus_hash_iter_get_string_key (DBusHashIter
*iter);
uintptr_t     _dbus_hash_iter_get_uintptr_key (DBusHashIter
*iter);
dbus_bool_t    _dbus_hash_iter_lookup        (DBusHashTable
*table,                                         void
*key,                                         dbus_bool_t
create_if_not_found,                           DBusHashIter
*iter);
void*          _dbus_hash_table_lookup_string (DBusHashTable
*table,                                         const char
*key);
void*          _dbus_hash_table_lookup_int   (DBusHashTable
*table,                                         int
key);
void*          _dbus_hash_table_lookup_uintptr (DBusHashTable
*table,

```

```

                                uintptr_t
key);
dbus_bool_t    _dbus_hash_table_remove_string (DBusHashTable
*table,
                                const char
*key);
dbus_bool_t    _dbus_hash_table_remove_int (DBusHashTable
*table,
                                int
key);
dbus_bool_t    _dbus_hash_table_remove_uintptr (DBusHashTable
*table,
                                uintptr_t
key);
dbus_bool_t    _dbus_hash_table_insert_string (DBusHashTable
*table,
                                char
*key,
                                void
*value);
dbus_bool_t    _dbus_hash_table_insert_int (DBusHashTable
*table,
                                int
key,
                                void
*value);
dbus_bool_t    _dbus_hash_table_insert_uintptr (DBusHashTable
*table,
                                uintptr_t
key,
                                void
*value);
int            _dbus_hash_table_get_n_entries (DBusHashTable
*table);

/* Preallocation */

/** A preallocated hash entry */
typedef struct DBusPreallocatedHash DBusPreallocatedHash;

DBusPreallocatedHash *_dbus_hash_table_preallocate_entry
(DBusHashTable      *table);
void                 _dbus_hash_table_free_preallocated_entry
(DBusHashTable      *table,

DBusPreallocatedHash *preallocated);
void                 _dbus_hash_table_insert_string_preallocated
(DBusHashTable      *table,

DBusPreallocatedHash *preallocated,

char                 *key,

```

```

void                                *value);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_HASH_H */

File = dbus-internals.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-internals.c  random utility stuff (internal to D-Bus
implementation)
 *
 * Copyright (C) 2002, 2003  Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */

#include <config.h>
#include "dbus-internals.h"
#include "dbus-protocol.h"
#include "dbus-marshal-basic.h"
#include "dbus-test.h"
#include "dbus-valgrind-internal.h"
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <stdlib.h>
#ifdef DBUS_USE_OUTPUT_DEBUG_STRING
#include <windows.h>

```



```

#include <mbstring.h>
#endif

/**
 * @defgroup DBusInternals D-Bus secret internal implementation
 * details
 * @brief Documentation useful when developing or debugging D-Bus
 * itself.
 *
 */

/**
 * @defgroup DBusInternalsUtils Utilities and portability
 * @ingroup DBusInternals
 * @brief Utility functions (_dbus_assert(), _dbus_warn(), etc.)
 * @{
 */

/**
 * @def _dbus_assert
 *
 * Aborts with an error message if the condition is false.
 *
 * @param condition condition which must be true.
 */

/**
 * @def _dbus_assert_not_reached
 *
 * Aborts with an error message if called.
 * The given explanation will be printed.
 *
 * @param explanation explanation of what happened if the code was
 * reached.
 */

/**
 * @def _DBUS_N_ELEMENTS
 *
 * Computes the number of elements in a fixed-size array using
 * sizeof().
 *
 * @param array the array to count elements in.
 */

/**
 * @def _DBUS_POINTER_TO_INT
 *
 * Safely casts a void* to an integer; should only be used on void*
 * that actually contain integers, for example one created with
 * _DBUS_INT_TO_POINTER. Only guaranteed to preserve 32 bits.
 * (i.e. it's used to store 32-bit ints in pointers, but

```

```

* can't be used to store 64-bit pointers in ints.)
*
* @param pointer pointer to extract an integer from.
*/
/**
* @def _DBUS_INT_TO_POINTER
*
* Safely stuffs an integer into a pointer, to be extracted later with
* _DBUS_POINTER_TO_INT. Only guaranteed to preserve 32 bits.
*
* @param integer the integer to stuff into a pointer.
*/
/**
* @def _DBUS_ZERO
*
* Sets all bits in an object to zero.
*
* @param object the object to be zeroed.
*/
/**
* @def _DBUS_INT16_MIN
*
* Minimum value of type "int16"
*/
/**
* @def _DBUS_INT16_MAX
*
* Maximum value of type "int16"
*/
/**
* @def _DBUS_UINT16_MAX
*
* Maximum value of type "uint16"
*/

/**
* @def _DBUS_INT32_MIN
*
* Minimum value of type "int32"
*/
/**
* @def _DBUS_INT32_MAX
*
* Maximum value of type "int32"
*/
/**
* @def _DBUS_UINT32_MAX
*
* Maximum value of type "uint32"
*/
/**

```

```

* @def _DBUS_INT_MIN
*
* Minimum value of type "int"
*/
/**
* @def _DBUS_INT_MAX
*
* Maximum value of type "int"
*/
/**
* @def _DBUS_UINT_MAX
*
* Maximum value of type "uint"
*/

/**
* @typedef DBusForeachFunction
*
* Used to iterate over each item in a collection, such as
* a DBusList.
*/

/**
* @def _DBUS_LOCK_NAME
*
* Expands to name of a global lock variable.
*/

/**
* @def _DBUS_DEFINE_GLOBAL_LOCK
*
* Defines a global lock variable with the given name.
* The lock must be added to the list to initialize
* in dbus_threads_init().
*/

/**
* @def _DBUS_DECLARE_GLOBAL_LOCK
*
* Expands to declaration of a global lock defined
* with _DBUS_DEFINE_GLOBAL_LOCK.
* The lock must be added to the list to initialize
* in dbus_threads_init().
*/

/**
* @def _DBUS_LOCK
*
* Locks a global lock
*/

/**

```

```

* @def _DBUS_UNLOCK
*
* Unlocks a global lock
*/

/**
* Fixed "out of memory" error message, just to avoid
* making up a different string every time and wasting
* space.
*/
const char *_dbus_no_memory_message = "Not enough memory";

static dbus_bool_t warn_initted = FALSE;
static dbus_bool_t fatal_warnings = FALSE;
static dbus_bool_t fatal_warnings_on_check_failed = TRUE;

static void
init_warnings(void)
{
    if (!warn_initted)
    {
        const char *s;
        s = _dbus_getenv ("DBUS_FATAL_WARNINGS");
        if (s && *s)
        {
            if (*s == '0')
            {
                fatal_warnings = FALSE;
                fatal_warnings_on_check_failed = FALSE;
            }
            else if (*s == '1')
            {
                fatal_warnings = TRUE;
                fatal_warnings_on_check_failed = TRUE;
            }
            else
            {
                fprintf(stderr, "DBUS_FATAL_WARNINGS should be set to 0
or 1 if set, not '%s'",
                    s);
            }
        }

        warn_initted = TRUE;
    }
}

/**
* Prints a warning message to stderr. Can optionally be made to exit
* fatally by setting DBUS_FATAL_WARNINGS, but this is rarely
* used. This function should be considered pretty much equivalent to
* fprintf(stderr). _dbus_warn_check_failed() on the other hand is

```

```

* suitable for use when a programming mistake has been made.
*
* @param format printf-style format string.
*/
void
_dbus_warn (const char *format,
            ...)
{
    va_list args;

    if (!warn_initted)
        init_warnings ();

    va_start (args, format);
    vfprintf (stderr, format, args);
    va_end (args);

    if (fatal_warnings)
    {
        fflush (stderr);
        _dbus_abort ();
    }
}

/**
 * Prints a "critical" warning to stderr when an assertion fails;
 * differs from _dbus_warn primarily in that it prefixes the pid and
 * defaults to fatal. This should be used only when a programming
 * error has been detected. (NOT for unavoidable errors that an app
 * might handle - those should be returned as DBusError.) Calling this
 * means "there is a bug"
 */
void
_dbus_warn_check_failed(const char *format,
                       ...)
{
    va_list args;

    if (!warn_initted)
        init_warnings ();

    fprintf (stderr, "process %lu: ", _dbus_pid_for_log ());

    va_start (args, format);
    vfprintf (stderr, format, args);
    va_end (args);

    if (fatal_warnings_on_check_failed)
    {
        fflush (stderr);
        _dbus_abort ();
    }
}

```

```

}

#ifdef DBUS_ENABLE_VERBOSE_MODE

static dbus_bool_t verbose_inittd = FALSE;
static dbus_bool_t verbose = TRUE;

/** Whether to show the current thread in verbose messages */
#define PTHREAD_IN_VERBOSE 0
#ifdef PTHREAD_IN_VERBOSE
#include <pthread.h>
#endif

#ifdef DBUS_USE_OUTPUT_DEBUG_STRING
static char module_name[1024];
#endif

static inline void
_dbus_verbose_init (void)
{
    if (!verbose_inittd)
    {
        const char *p = _dbus_getenv ("DBUS_VERBOSE");
        verbose = p != NULL && *p == '1';
        verbose_inittd = TRUE;
#ifdef DBUS_USE_OUTPUT_DEBUG_STRING
    {
        char *last_period, *last_slash;
        GetModuleFileName(0, module_name, sizeof(module_name)-1);
        last_period = _mbsrchr(module_name, '.');
        if (last_period)
            *last_period = '\\0';
        last_slash = _mbsrchr(module_name, '\\');
        if (last_slash)
            strcpy(module_name, last_slash+1);
            strcat(module_name, ": ");
        }
#endif
    }
}

/** @def DBUS_IS_DIR_SEPARATOR(c)
 * macro for checking if character c is a patch separator
 *
 * @todo move to a header file so that others can use this too
 */
#ifdef DBUS_WIN
#define DBUS_IS_DIR_SEPARATOR(c) (c == '\\\\' || c == '/')
#else
#define DBUS_IS_DIR_SEPARATOR(c) (c == '/')
#endif

```

```

/**
 remove source root from file path
 the source root is determined by
 */
static char *_dbus_file_path_extract_elements_from_tail(const char
*file,int level)
{
    static int prefix = -1;

    if (prefix == -1)
        {
            char *p = (char *)file + strlen(file);
            int i = 0;
            prefix = 0;
            for (;p >= file;p--)
                {
                    if (DBUS_IS_DIR_SEPARATOR(*p))
                        {
                            if (++i >= level)
                                {
                                    prefix = p-file+1;
                                    break;
                                }
                        }
                }
        }
    return (char *)file+prefix;
}

/**
 * Implementation of dbus_is_verbose() macro if built with verbose
 logging
 * enabled.
 * @returns whether verbose logging is active.
 */
dbus_bool_t
_dbus_is_verbose_real (void)
{
    _dbus_verbose_init ();
    return verbose;
}

/**
 * Prints a warning message to stderr
 * if the user has enabled verbose mode.
 * This is the real function implementation,
 * use _dbus_verbose() macro in code.
 *
 * @param format printf-style format string.
 */
void
_dbus_verbose_real (

```

```

#ifdef DBUS_CPP_SUPPORTS_VARIABLE_MACRO_ARGUMENTS
        const char *file,
        const int line,
        const char *function,
#endif

        const char *format,
        ...)
{
    va_list args;
    static dbus_bool_t need_pid = TRUE;
    int len;

    /* things are written a bit oddly here so that
     * in the non-verbose case we just have the one
     * conditional and return immediately.
     */
    if (!dbus_is_verbose_real())
        return;

#ifdef DBUS_USE_OUTPUT_DEBUG_STRING
    /* Print out pid before the line */
    if (need_pid)
    {
#ifdef PTHREAD_IN_VERBOSE
        fprintf (stderr, "%lu: 0x%lx: ", _dbus_pid_for_log (),
pthread_self ());
#else
        fprintf (stderr, "%lu: ", _dbus_pid_for_log ());
#endif
    }
#endif

    /* Only print pid again if the next line is a new line */
    len = strlen (format);
    if (format[len-1] == '\n')
        need_pid = TRUE;
    else
        need_pid = FALSE;

    va_start (args, format);
#ifdef DBUS_USE_OUTPUT_DEBUG_STRING
    {
        char buf[1024];
        strcpy(buf,module_name);
#ifdef DBUS_CPP_SUPPORTS_VARIABLE_MACRO_ARGUMENTS
        sprintf (buf+strlen(buf), "[%s(%d):%s]
",_dbus_file_path_extract_elements_from_tail(file,2),line,function);
#endif
        vsprintf (buf+strlen(buf),format, args);
        va_end (args);
        OutputDebugStringA(buf);
    }
}

```



```

#else
#ifdef DBUS_CPP_SUPPORTS_VARIABLE_MACRO_ARGUMENTS
    fprintf (stderr, "[%s(%d):%s]
", _dbus_file_path_extract_elements_from_tail(file,2),line,function);
#endif

    vfprintf (stderr, format, args);
    va_end (args);

    fflush (stderr);
#endif
}

/**
 * Reinitializes the verbose logging code, used
 * as a hack in dbus-spawn.c so that a child
 * process re-reads its pid
 */
void
_dbus_verbose_reset_real (void)
{
    verbose_initted = FALSE;
}

void
_dbus_trace_ref (const char *obj_name,
                void *obj,
                int old_refcount,
                int new_refcount,
                const char *why,
                const char *env_var,
                int *enabled)
{
    _dbus_assert (obj_name != NULL);
    _dbus_assert (obj != NULL);
    _dbus_assert (old_refcount >= -1);
    _dbus_assert (new_refcount >= -1);

    if (old_refcount == -1)
    {
        _dbus_assert (new_refcount == -1);
    }
    else
    {
        _dbus_assert (new_refcount >= 0);
        _dbus_assert (old_refcount >= 0);
        _dbus_assert (old_refcount > 0 || new_refcount > 0);
    }

    _dbus_assert (why != NULL);
    _dbus_assert (env_var != NULL);
}

```

```

_dbus_assert (enabled != NULL);

if (*enabled < 0)
{
    const char *s = _dbus_getenv (env_var);

    *enabled = FALSE;

    if (s && *s)
    {
        if (*s == '0')
            *enabled = FALSE;
        else if (*s == '1')
            *enabled = TRUE;
        else
            _dbus_warn ("%s should be 0 or 1 if set, not '%s'",
env_var, s);
    }
}

if (*enabled)
{
    if (old_refcount == -1)
    {
        VALGRIND_PRINTF_BACKTRACE ("%s %p ref stolen (%s)",
                                obj_name, obj, why);
        _dbus_verbose ("%s %p ref stolen (%s)",
                        obj_name, obj, why);
    }
    else
    {
        VALGRIND_PRINTF_BACKTRACE ("%s %p %d -> %d refs (%s)",
                                obj_name, obj,
                                old_refcount, new_refcount, why);
        _dbus_verbose ("%s %p %d -> %d refs (%s)",
                        obj_name, obj, old_refcount, new_refcount,
why);
    }
}
}

#endif /* DBUS_ENABLE_VERBOSE_MODE */

/**
 * Duplicates a string. Result must be freed with
 * dbus_free(). Returns #NULL if memory allocation fails.
 * If the string to be duplicated is #NULL, returns #NULL.
 *
 * @param str string to duplicate.
 * @returns newly-allocated copy.
 */
char*

```

```

_dbus_strdup (const char *str)
{
    size_t len;
    char *copy;

    if (str == NULL)
        return NULL;

    len = strlen (str);

    copy = dbus_malloc (len + 1);
    if (copy == NULL)
        return NULL;

    memcpy (copy, str, len + 1);

    return copy;
}

/**
 * Duplicates a block of memory. Returns
 * #NULL on failure.
 *
 * @param mem memory to copy
 * @param n_bytes number of bytes to copy
 * @returns the copy
 */
void*
_dbus_memdup (const void *mem,
              size_t      n_bytes)
{
    void *copy;

    copy = dbus_malloc (n_bytes);
    if (copy == NULL)
        return NULL;

    memcpy (copy, mem, n_bytes);

    return copy;
}

/**
 * Duplicates a string array. Result may be freed with
 * dbus_free_string_array(). Returns #NULL if memory allocation fails.
 * If the array to be duplicated is #NULL, returns #NULL.
 *
 * @param array array to duplicate.
 * @returns newly-allocated copy.
 */
char**
_dbus_dup_string_array (const char **array)

```

```

{
    int len;
    int i;
    char **copy;

    if (array == NULL)
        return NULL;

    for (len = 0; array[len] != NULL; ++len)
        ;

    copy = dbus_new0 (char*, len + 1);
    if (copy == NULL)
        return NULL;

    i = 0;
    while (i < len)
    {
        copy[i] = _dbus_strdup (array[i]);
        if (copy[i] == NULL)
        {
            dbus_free_string_array (copy);
            return NULL;
        }

        ++i;
    }

    return copy;
}

/**
 * Checks whether a string array contains the given string.
 *
 * @param array array to search.
 * @param str string to look for
 * @returns #TRUE if array contains string
 */
dbus_bool_t
_dbus_string_array_contains (const char **array,
                             const char *str)
{
    int i;

    i = 0;
    while (array[i] != NULL)
    {
        if (strcmp (array[i], str) == 0)
            return TRUE;
        ++i;
    }
}

```

```

    return FALSE;
}

/**
 * Generates a new UUID. If you change how this is done,
 * there's some text about it in the spec that should also change.
 *
 * @param uuid the uuid to initialize
 */
void
_dbus_generate_uuid (DBusGUID *uuid)
{
    long now;

    /* don't use monotonic time because the UUID may be saved to disk,
    e.g.
    * it may persist across reboots
    */
    _dbus_get_real_time (&now, NULL);

    uuid->as_uint32s[DBUS_UUID_LENGTH_WORDS - 1] = DBUS_UINT32_TO_BE
(now);

    _dbus_generate_random_bytes_buffer (uuid->as_bytes,
DBUS_UUID_LENGTH_BYTES - 4);
}

/**
 * Hex-encode a UUID.
 *
 * @param uuid the uuid
 * @param encoded string to append hex uuid to
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_uuid_encode (const DBusGUID *uuid,
                  DBusString      *encoded)
{
    DBusString binary;
    _dbus_string_init_const_len (&binary, uuid->as_bytes,
DBUS_UUID_LENGTH_BYTES);
    return _dbus_string_hex_encode (&binary, 0, encoded,
_dbus_string_get_length (encoded));
}

static dbus_bool_t
_dbus_read_uuid_file_without_creating (const DBusString *filename,
                                      DBusGUID          *uuid,
                                      DBusError          *error)
{
    DBusString contents;
    DBusString decoded;

```

```

int end;

if (!_dbus_string_init (&contents))
{
    _DBUS_SET_OOM (error);
    return FALSE;
}

if (!_dbus_string_init (&decoded))
{
    _dbus_string_free (&contents);
    _DBUS_SET_OOM (error);
    return FALSE;
}

if (!_dbus_file_get_contents (&contents, filename, error))
    goto error;

_dbus_string_chop_white (&contents);

if (_dbus_string_get_length (&contents) != DBUS_UUID_LENGTH_HEX)
{
    dbus_set_error (error, DBUS_ERROR_INVALID_FILE_CONTENT,
        "UUID file '%s' should contain a hex string of
length %d, not length %d, with no other text",
        _dbus_string_get_const_data (filename),
        DBUS_UUID_LENGTH_HEX,
        _dbus_string_get_length (&contents));
    goto error;
}

if (!_dbus_string_hex_decode (&contents, 0, &end, &decoded, 0))
{
    _DBUS_SET_OOM (error);
    goto error;
}

if (end == 0)
{
    dbus_set_error (error, DBUS_ERROR_INVALID_FILE_CONTENT,
        "UUID file '%s' contains invalid hex data",
        _dbus_string_get_const_data (filename));
    goto error;
}

if (_dbus_string_get_length (&decoded) != DBUS_UUID_LENGTH_BYTES)
{
    dbus_set_error (error, DBUS_ERROR_INVALID_FILE_CONTENT,
        "UUID file '%s' contains %d bytes of hex-encoded
data instead of %d",
        _dbus_string_get_const_data (filename),
        _dbus_string_get_length (&decoded),

```

```

        DBUS_UUID_LENGTH_BYTES);
    goto error;
}

_dbus_string_copy_to_buffer (&decoded, uuid->as_bytes,
DBUS_UUID_LENGTH_BYTES);

_dbus_string_free (&decoded);
_dbus_string_free (&contents);

_DBUS_ASSERT_ERROR_IS_CLEAR (error);

return TRUE;

error:
    _DBUS_ASSERT_ERROR_IS_SET (error);
    _dbus_string_free (&contents);
    _dbus_string_free (&decoded);
    return FALSE;
}

static dbus_bool_t
_dbus_create_uid_file_exclusively (const DBusString *filename,
                                   DBusGUID          *uuid,
                                   DBusError         *error)
{
    DBusString encoded;

    if (!_dbus_string_init (&encoded))
    {
        _DBUS_SET_OOM (error);
        return FALSE;
    }

    _dbus_generate_uid (uuid);

    if (!_dbus_uuid_encode (uuid, &encoded))
    {
        _DBUS_SET_OOM (error);
        goto error;
    }

    if (!_dbus_string_append_byte (&encoded, '\n'))
    {
        _DBUS_SET_OOM (error);
        goto error;
    }

    if (!_dbus_string_save_to_file (&encoded, filename, TRUE, error))
        goto error;

    _dbus_string_free (&encoded);
}

```

```

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    return TRUE;

error:
    _DBUS_ASSERT_ERROR_IS_SET (error);
    _dbus_string_free (&encoded);
    return FALSE;
}

/**
 * Reads (and optionally writes) a uuid to a file. Initializes the
 * uuid
 * unless an error is returned.
 *
 * @param filename the name of the file
 * @param uuid uuid to be initialized with the loaded uuid
 * @param create_if_not_found #TRUE to create a new uuid and save it
 * if the file doesn't exist
 * @param error the error return
 * @returns #FALSE if the error is set
 */
dbus_bool_t
_dbus_read_uuid_file (const DBusString *filename,
                     DBusGUID         *uuid,
                     dbus_bool_t       create_if_not_found,
                     DBusError         *error)
{
    DBusError read_error = DBUS_ERROR_INIT;

    if (_dbus_read_uuid_file_without_creating (filename, uuid,
&read_error))
        return TRUE;

    if (!create_if_not_found)
    {
        dbus_move_error (&read_error, error);
        return FALSE;
    }

    /* If the file exists and contains junk, we want to keep that error
     * message instead of overwriting it with a "file exists" error
     * message when we try to write
     */
    if (dbus_error_has_name (&read_error,
DBUS_ERROR_INVALID_FILE_CONTENT))
    {
        dbus_move_error (&read_error, error);
        return FALSE;
    }
    else
    {

```



```

        dbus_error_free (&read_error);
        return _dbus_create_uuid_file_exclusively (filename, uuid,
error);
    }
}

_DBUS_DEFINE_GLOBAL_LOCK (machine_uuid);
static int machine_uuid_initialized_generation = 0;
static DBusGUID machine_uuid;

/**
 * Gets the hex-encoded UUID of the machine this function is
 * executed on. This UUID is guaranteed to be the same for a given
 * machine at least until it next reboots, though it also
 * makes some effort to be the same forever, it may change if the
 * machine is reconfigured or its hardware is modified.
 *
 * @param uuid_str string to append hex-encoded machine uuid to
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_get_local_machine_uuid_encoded (DBusString *uuid_str)
{
    dbus_bool_t ok;

    _DBUS_LOCK (machine_uuid);
    if (machine_uuid_initialized_generation != _dbus_current_generation)
    {
        DBusError error = DBUS_ERROR_INIT;

        if (!_dbus_read_local_machine_uuid (&machine_uuid, FALSE,
&error))
        {
#ifdef DBUS_BUILD_TESTS
            /* For the test suite, we may not be installed so just
continue silently
            * here. But in a production build, we want to be nice and
loud about
            * this.
            */
            _dbus_warn_check_failed ("D-Bus library appears to be
incorrectly set up; failed to read machine uuid: %s\n"
"See the manual page for dbus-
uuidgen to correct this issue.\n",
error.message);
#endif
            dbus_error_free (&error);

            _dbus_generate_uuid (&machine_uuid);
        }
    }
}

```

```

    ok = _dbus_uuid_encode (&machine_uuid, uuid_str);

    _DBUS_UNLOCK (machine_uuid);

    return ok;
}

#ifdef DBUS_DISABLE_CHECKS
/** String used in _dbus_return_if_fail macro */
const char *_dbus_return_if_fail_warning_format =
"arguments to %s() were incorrect, assertion \"%s\" failed in file %s
line %d.\n"
"This is normally a bug in some application using the D-Bus
library.\n";
#endif

#ifdef DBUS_DISABLE_ASSERT
/**
 * Internals of _dbus_assert(); it's a function
 * rather than a macro with the inline code so
 * that the assertion failure blocks don't show up
 * in test suite coverage, and to shrink code size.
 *
 * @param condition TRUE if assertion succeeded
 * @param condition_text condition as a string
 * @param file file the assertion is in
 * @param line line the assertion is in
 * @param func function the assertion is in
 */
void
_dbus_real_assert (dbus_bool_t condition,
                  const char *condition_text,
                  const char *file,
                  int line,
                  const char *func)
{
    if (_DBUS_UNLIKELY (!condition))
    {
        _dbus_warn ("%lu: assertion failed \"%s\" file \"%s\" line %d
function %s\n",
                  _dbus_pid_for_log (), condition_text, file, line,
func);
        _dbus_abort ();
    }
}

/**
 * Internals of _dbus_assert_not_reached(); it's a function
 * rather than a macro with the inline code so
 * that the assertion failure blocks don't show up
 * in test suite coverage, and to shrink code size.

```

```

*
* @param explanation what was reached that shouldn't have been
* @param file file the assertion is in
* @param line line the assertion is in
*/
void
_dbus_real_assert_not_reached (const char *explanation,
                               const char *file,
                               int         line)
{
    _dbus_warn ("File \"%s\" line %d process %lu should not have been
reached: %s\n",
               file, line, _dbus_pid_for_log (), explanation);
    _dbus_abort ();
}
#endif /* DBUS_DISABLE_ASSERT */

#ifdef DBUS_BUILD_TESTS
static dbus_bool_t
run_failing_each_malloc (int         n_mallocs,
                         const char  *description,
                         DBusTestMemoryFunction func,
                         void         *data)
{
    n_mallocs += 10; /* fudge factor to ensure reallocs etc. are covered
*/

    while (n_mallocs >= 0)
    {
        _dbus_set_fail_alloc_counter (n_mallocs);

        _dbus_verbose ("\n===\n%s: (will fail malloc %d with %d
failures)\n===\n",
                       description, n_mallocs,
                       _dbus_get_fail_alloc_failures ());

        if (!(* func) (data))
            return FALSE;

        n_mallocs -= 1;
    }

    _dbus_set_fail_alloc_counter (_DBUS_INT_MAX);

    return TRUE;
}

/**
 * Tests how well the given function responds to out-of-memory
 * situations. Calls the function repeatedly, failing a different
 * call to malloc() each time. If the function ever returns #FALSE,
 * the test fails. The function should return #TRUE whenever something

```

```

* valid (such as returning an error, or succeeding) occurs, and
#FALSE
* if it gets confused in some way.
*
* @param description description of the test used in verbose output
* @param func function to call
* @param data data to pass to function
* @returns #TRUE if the function never returns FALSE
*/
dbus_bool_t
_dbus_test_oom_handling (const char          *description,
                        DBusTestMemoryFunction func,
                        void                 *data)
{
    int approx_mallocs;
    const char *setting;
    int max_failures_to_try;
    int i;

    /* Run once to see about how many mallocs are involved */

    _dbus_set_fail_alloc_counter (_DBUS_INT_MAX);

    _dbus_verbose ("Running once to count mallocs\n");

    if (!(* func) (data))
        return FALSE;

    approx_mallocs = _DBUS_INT_MAX - _dbus_get_fail_alloc_counter ();

    _dbus_verbose ("\n=====
total\n=====
",
                  description, approx_mallocs);

    setting = _dbus_getenv ("DBUS_TEST_MALLOC_FAILURES");
    if (setting != NULL)
    {
        DBusString str;
        long v;
        _dbus_string_init_const (&str, setting);
        v = 4;
        if (!_dbus_string_parse_int (&str, 0, &v, NULL))
            _dbus_warn ("couldn't parse '%s' as integer\n", setting);
        max_failures_to_try = v;
    }
    else
    {
        max_failures_to_try = 4;
    }

    i = setting ? max_failures_to_try - 1 : 1;
    while (i < max_failures_to_try)

```

```

    {
        _dbus_set_fail_alloc_failures (i);
        if (!run_failing_each_malloc (approx_mallocs, description, func,
data))
            return FALSE;
        ++i;
    }

    _dbus_verbose ("\n=====\n%s: all iterations
passed\n=====\n",
description);

    return TRUE;
}
#endif /* DBUS_BUILD_TESTS */

/** @} */

```

File = dbus-internals.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-internals.h  random utility stuff (internal to D-Bus
implementation)
*
* Copyright (C) 2002, 2003  Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
*
*/
#ifdef DBUS_INSIDE_DBUS_H
#error "You can't include dbus-internals.h in the public header
dbus.h"
#endif

```

```

#ifndef DBUS_INTERNALS_H
#define DBUS_INTERNALS_H

#include <dbus/dbus-memory.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-sysdeps.h>
#include <dbus/dbus-threads-internal.h>

DBUS_BEGIN_DECLS

#ifndef DBUS_SESSION_BUS_DEFAULT_ADDRESS
#define DBUS_SESSION_BUS_DEFAULT_ADDRESS "autolaunch:"
#endif

void _dbus_warn          (const char *format,
                          ...) _DBUS_GNUC_PRINTF (1, 2);

void _dbus_warn_check_failed (const char *format,
                              ...) _DBUS_GNUC_PRINTF (1, 2);

#if defined (__STDC_VERSION__) && (__STDC_VERSION__ >= 199901L)
#define _DBUS_FUNCTION_NAME __func__
#elif defined (__GNUC__) || defined (__MSC_VER)
#define _DBUS_FUNCTION_NAME __FUNCTION__
#else
#define _DBUS_FUNCTION_NAME "unknown function"
#endif

/*
 * (code from GLib)
 *
 * The _DBUS_LIKELY and _DBUS_UNLIKELY macros let the programmer give
hints to
 * the compiler about the expected result of an expression. Some
compilers
 * can use this information for optimizations.
 *
 * The _DBUS_BOOLEAN_EXPR macro is intended to trigger a gcc warning
when
 * putting assignments in the macro arg
 */
#if defined (__GNUC__) && (__GNUC__ > 2) && defined (__OPTIMIZE__)
#define _DBUS_BOOLEAN_EXPR(expr) \
__extension__ ({ \
    int _dbus_boolean_var_; \
    if (expr) \
        _dbus_boolean_var_ = 1; \
    else \
        _dbus_boolean_var_ = 0; \
})

```

```

    _dbus_boolean_var_;
})
#define _DBUS_LIKELY(expr) (__builtin_expect
(_DBUS_BOOLEAN_EXPR(expr), 1))
#define _DBUS_UNLIKELY(expr) (__builtin_expect
(_DBUS_BOOLEAN_EXPR(expr), 0))
#else
#define _DBUS_LIKELY(expr) (expr)
#define _DBUS_UNLIKELY(expr) (expr)
#endif

#ifdef DBUS_ENABLE_VERBOSE_MODE

/*
at least gnu cc and msvc compiler are known to
have support for variable macro argument lists
add other compilers is required
*/
#if defined(__GNUC__) || defined(_MSC_VER)
#define DBUS_CPP_SUPPORTS_VARIABLE_MACRO_ARGUMENTS
#endif

#ifdef DBUS_CPP_SUPPORTS_VARIABLE_MACRO_ARGUMENTS
void _dbus_verbose_real      (const char *file, const int line, const
char *function,
                             const char *format,...)
    _DBUS_GNUC_PRINTF (4, 5);
# define _dbus_verbose(fmt,...) _dbus_verbose_real(
__FILE__, __LINE__, __FUNCTION__,fmt, ## __VA_ARGS__)
#else
void _dbus_verbose_real      (const char *format,
                             ...) _DBUS_GNUC_PRINTF (1, 2);
# define _dbus_verbose _dbus_verbose_real
#endif
void _dbus_verbose_reset_real (void);
dbus_bool_t _dbus_is_verbose_real (void);

# define _dbus_verbose_reset _dbus_verbose_reset_real
# define _dbus_is_verbose _dbus_is_verbose_real
#else
# ifdef HAVE_ISO_VARARGS
#   define _dbus_verbose(...) do { } while (0)
# elif defined (HAVE_GNUC_VARARGS)
#   define _dbus_verbose(format...) do { } while (0)
# else
static void _dbus_verbose(const char * x,...) {};
# endif
# define _dbus_verbose_reset() do { } while (0)
# define _dbus_is_verbose() FALSE
#endif /* !DBUS_ENABLE_VERBOSE_MODE */

void _dbus_trace_ref (const char *obj_name,

```

```

        void          *obj,
        int           old_refcount,
        int           new_refcount,
        const char    *why,
        const char    *env_var,
        int           *enabled);

const char* _dbus_strerror (int error_number);

#ifdef DBUS_DISABLE_ASSERT
#define _dbus_assert(condition) do { } while (0)
#else
void _dbus_real_assert (dbus_bool_t condition,
                       const char *condition_text,
                       const char *file,
                       int line,
                       const char *func);
#define _dbus_assert(condition)
\
    _dbus_real_assert ((condition) != 0, #condition, __FILE__, __LINE__,
    _DBUS_FUNCTION_NAME)
#endif /* !DBUS_DISABLE_ASSERT */

#ifdef DBUS_DISABLE_ASSERT
#define _dbus_assert_not_reached(explanation) do { } while (0)
#else
void _dbus_real_assert_not_reached (const char *explanation,
                                    const char *file,
                                    int line)

_DBUS_GNUC_NORETURN;
#define _dbus_assert_not_reached(explanation)
\
    _dbus_real_assert_not_reached (explanation, __FILE__, __LINE__)
#endif /* !DBUS_DISABLE_ASSERT */

#ifdef DBUS_DISABLE_CHECKS
#define _dbus_return_if_fail(condition)
#define _dbus_return_val_if_fail(condition, val)
#else

extern const char *_dbus_return_if_fail_warning_format;

#define _dbus_return_if_fail(condition) do {
\
    _dbus_assert ((*const char*)_DBUS_FUNCTION_NAME) != '_');
\
    if (!(condition)) {
\
        _dbus_warn_check_failed (_dbus_return_if_fail_warning_format,
\
                                _DBUS_FUNCTION_NAME, #condition,
__FILE__, __LINE__); \

```



```

    return;
\
} } while (0)

#define _dbus_return_val_if_fail(condition, val) do {
\
    _dbus_assert ((* (const char*)_DBUS_FUNCTION_NAME) != '_');
\
    if (!(condition)) {
\
        _dbus_warn_check_failed (_dbus_return_if_fail_warning_format,
\
                                _DBUS_FUNCTION_NAME, #condition,
__FILE__, __LINE__);
        return (val);
\
    } } while (0)

#endif /* !DBUS_DISABLE_ASSERT */

#define _DBUS_N_ELEMENTS(array) ((int) (sizeof ((array)) / sizeof
((array)[0])))

#define _DBUS_POINTER_TO_INT(pointer) ((intptr_t)(pointer))
#define _DBUS_INT_TO_POINTER(integer) ((void*)((intptr_t)(integer)))

#define _DBUS_ZERO(object) (memset (&(object), '\0', sizeof
((object))))

#define _DBUS_STRUCT_OFFSET(struct_type, member) \
    ((intptr_t) ((unsigned char*) &((struct_type*) 0)->member))

#ifdef DBUS_DISABLE_CHECKS
/* this is an assert and not an error, but in the typical --disable-
checks case (you're trying
* to really minimize code size), disabling these assertions makes
sense.
*/
#define _DBUS_ASSERT_ERROR_IS_SET(error) do { } while (0)
#define _DBUS_ASSERT_ERROR_IS_CLEAR(error) do { } while (0)
#else
#define _DBUS_ASSERT_ERROR_IS_SET(error) _dbus_assert ((error) ==
NULL || dbus_error_is_set ((error)))
#define _DBUS_ASSERT_ERROR_IS_CLEAR(error) _dbus_assert ((error) ==
NULL || !dbus_error_is_set ((error)))
#endif

#define _dbus_return_if_error_is_set(error) _dbus_return_if_fail
((error) == NULL || !dbus_error_is_set ((error)))
#define _dbus_return_val_if_error_is_set(error, val)
_dbus_return_val_if_fail ((error) == NULL || !dbus_error_is_set
((error)), (val))

```

```

/* This alignment thing is from ORBit2 */
/* Align a value upward to a boundary, expressed as a number of bytes.
 * E.g. align to an 8-byte boundary with argument of 8.
 */

/*
 * (this + boundary - 1)
 * &
 * ~(boundary - 1)
 */

#define _DBUS_ALIGN_VALUE(this, boundary) \
    (((uintptr_t)(this)) + ((uintptr_t)(boundary) - 1)) &
    (~(((uintptr_t)(boundary)) - 1))

#define _DBUS_ALIGN_ADDRESS(this, boundary) \
    ((void*)_DBUS_ALIGN_VALUE(this, boundary))

char*      _dbus_strdup      (const char *str);
void*      _dbus_memdup     (const void *mem,
                             size_t      n_bytes);
dbus_bool_t _dbus_string_array_contains (const char **array,
                                         const char *str);
char**     _dbus_dup_string_array (const char **array);

#define _DBUS_INT16_MIN      ((dbus_int16_t) 0x8000)
#define _DBUS_INT16_MAX      ((dbus_int16_t) 0x7fff)
#define _DBUS_UINT16_MAX     ((dbus_uint16_t) 0xffff)
#define _DBUS_INT32_MIN      ((dbus_int32_t) 0x80000000)
#define _DBUS_INT32_MAX      ((dbus_int32_t) 0x7fffffff)
#define _DBUS_UINT32_MAX     ((dbus_uint32_t) 0xffffffff)
/* using 32-bit here is sort of bogus */
#define _DBUS_INT_MIN        _DBUS_INT32_MIN
#define _DBUS_INT_MAX        _DBUS_INT32_MAX
#define _DBUS_UINT_MAX        _DBUS_UINT32_MAX
#ifdef DBUS_HAVE_INT64
#define _DBUS_INT64_MAX      DBUS_INT64_CONSTANT (0x7fffffffffffffff)
#define _DBUS_UINT64_MAX    DBUS_UINT64_CONSTANT (0xffffffffffffffff)
#endif
#define _DBUS_ONE_KILOBYTE 1024
#define _DBUS_ONE_MEGABYTE 1024 * _DBUS_ONE_KILOBYTE
#define _DBUS_ONE_HOUR_IN_MILLISECONDS (1000 * 60 * 60)
#define _DBUS_USEC_PER_SECOND (1000000)

#undef      MAX
#define MAX(a, b)  (((a) > (b)) ? (a) : (b))

#undef      MIN
#define MIN(a, b)  (((a) < (b)) ? (a) : (b))

```

```

#undef ABS
#define ABS(a) ((a) < 0) ? -(a) : (a)

#define _DBUS_ISASCII(c) ((c) != '\0' && (((c) & ~0x7f) == 0))

typedef void (* DBusForeachFunction) (void *element,
                                       void *data);

dbus_bool_t _dbus_set_fd_nonblocking (int fd,
                                     DBusError *error);

void _dbus_verbose_bytes (const unsigned char *data,
                          int len,
                          int offset);
void _dbus_verbose_bytes_of_string (const DBusString *str,
                                    int start,
                                    int len);

extern const char *_dbus_no_memory_message;
#define _DBUS_SET_OOM(error) dbus_set_error_const ((error),
DBUS_ERROR_NO_MEMORY, _dbus_no_memory_message)

#ifdef DBUS_BUILD_TESTS
/* Memory debugging */
void _dbus_set_fail_alloc_counter (int
until_next_fail);
int _dbus_get_fail_alloc_counter (void);
void _dbus_set_fail_alloc_failures (int
failures_per_failure);
int _dbus_get_fail_alloc_failures (void);
dbus_bool_t _dbus_decrement_fail_alloc_counter (void);
dbus_bool_t _dbus_disable_mem_pools (void);
int _dbus_get_malloc_blocks_outstanding (void);

typedef dbus_bool_t (* DBusTestMemoryFunction) (void *data);
dbus_bool_t _dbus_test_oom_handling (const char
*description,
                                     DBusTestMemoryFunction func,
                                     void *data);
#else
#define _dbus_set_fail_alloc_counter(n)
#define _dbus_get_fail_alloc_counter _DBUS_INT_MAX

/* These are constant expressions so that blocks
 * they protect should be optimized away
 */
#define _dbus_decrement_fail_alloc_counter() (FALSE)
#define _dbus_disable_mem_pools() (FALSE)
#define _dbus_get_malloc_blocks_outstanding (0)
#endif /* !DBUS_BUILD_TESTS */

typedef void (* DBusShutdownFunction) (void *data);

```

```

dbus_bool_t _dbus_register_shutdown_func (DBusShutdownFunction
function,
                                         void
                                         *data);

extern int _dbus_current_generation;

/* Thread initializers */
#define _DBUS_LOCK_NAME(name) _dbus_lock_##name
#define _DBUS_DECLARE_GLOBAL_LOCK(name) extern DBusRMutex
*_dbus_lock_##name
#define _DBUS_DEFINE_GLOBAL_LOCK(name) DBusRMutex
*_dbus_lock_##name
#define _DBUS_LOCK(name) _dbus_rmutex_lock
(_dbus_lock_##name)
#define _DBUS_UNLOCK(name) _dbus_rmutex_unlock
(_dbus_lock_##name)

/* 1-5 */
_DBUS_DECLARE_GLOBAL_LOCK (list);
_DBUS_DECLARE_GLOBAL_LOCK (connection_slots);
_DBUS_DECLARE_GLOBAL_LOCK (pending_call_slots);
_DBUS_DECLARE_GLOBAL_LOCK (server_slots);
_DBUS_DECLARE_GLOBAL_LOCK (message_slots);
/* 5-10 */
_DBUS_DECLARE_GLOBAL_LOCK (bus);
_DBUS_DECLARE_GLOBAL_LOCK (bus_datas);
_DBUS_DECLARE_GLOBAL_LOCK (shutdown_funcs);
_DBUS_DECLARE_GLOBAL_LOCK (system_users);
_DBUS_DECLARE_GLOBAL_LOCK (message_cache);
/* 10-14 */
_DBUS_DECLARE_GLOBAL_LOCK (shared_connections);
_DBUS_DECLARE_GLOBAL_LOCK (win_fds);
_DBUS_DECLARE_GLOBAL_LOCK (sid_atom_cache);
_DBUS_DECLARE_GLOBAL_LOCK (machine_uuid);

#ifdef !DBUS_USE_SYNC
_DBUS_DECLARE_GLOBAL_LOCK (atomic);
#define _DBUS_N_GLOBAL_LOCKS (15)
#else
#define _DBUS_N_GLOBAL_LOCKS (14)
#endif

dbus_bool_t _dbus_threads_init_debug (void);

dbus_bool_t _dbus_address_append_escaped (DBusString
                                         *escaped,
                                         const DBusString
                                         *unescaped);

void _dbus_set_bad_address (DBusError
                           *error,
                           const char
                           *address_problem_type,

```

```

const char
*address_problem_field,
const char
*address_problem_other);

#define DBUS_UUID_LENGTH_BYTES 16
#define DBUS_UUID_LENGTH_WORDS (DBUS_UUID_LENGTH_BYTES / 4)
#define DBUS_UUID_LENGTH_HEX (DBUS_UUID_LENGTH_BYTES * 2)

/**
 * A globally unique ID ; we have one for each DbusServer, and also
 one for each
 * machine with libdbus installed on it.
 */
union DbusGUID
{
    dbus_uint32_t as_uint32s[DBUS_UUID_LENGTH_WORDS];    /**< guid as
four uint32 values */
    char as_bytes[DBUS_UUID_LENGTH_BYTES];              /**< guid as
16 single-byte values */
};

void _dbus_generate_uuid (DBusGUID *uuid);
dbus_bool_t _dbus_uuid_encode (const DbusGUID *uuid,
DBusString *encoded);
dbus_bool_t _dbus_read_uuid_file (const DbusString *filename,
DBusGUID *uuid,
dbus_bool_t
create_if_not_found,
DBusError *error);

dbus_bool_t _dbus_get_local_machine_uuid_encoded (DBusString
*uuid_str);

#define _DBUS_PASTE2(a, b) a ## b
#define _DBUS_PASTE(a, b) _DBUS_PASTE2 (a, b)
#define _DBUS_STATIC_ASSERT(expr) \
    typedef struct { char _assertion[(expr) ? 1 : -1]; } \
    _DBUS_PASTE (_DBUS_STATIC_ASSERT_, __LINE__)

DBUS_END_DECLS

#endif /* DBUS_INTERNALS_H */

File = dbus-keyring.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-keyring.c Store secret cookies in your homedir
 *
 * Copyright (C) 2003, 2004 Red Hat Inc.

```

```
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-keyring.h"
#include "dbus-protocol.h"
#include <dbus/dbus-string.h>
#include <dbus/dbus-list.h>
#include <dbus/dbus-sysdeps.h>
```

```
/**
 * @defgroup DBusKeyring keyring class
 * @ingroup DBusInternals
 * @brief DBusKeyring data structure
 *
 * Types and functions related to DBusKeyring. DBusKeyring is intended
 * to manage cookies used to authenticate clients to servers. This is
 * essentially the "verify that client can read the user's homedir"
 * authentication mechanism. Both client and server must have access
 * to the homedir.
 *
 * The secret keys are not kept in locked memory, and are written to a
 * file in the user's homedir. However they are transient (only used
 * by a single server instance for a fixed period of time, then
 * discarded). Also, the keys are not sent over the wire.
 *
 * @todo there's a memory leak on some codepath in here, I saw it once
 * when running make check - probably some specific initial cookies
 * present in the cookie file, then depending on what we do with them.
 */
```

```
/**
 * @defgroup DBusKeyringInternals DBusKeyring implementation details
```

```

* @ingroup DBusInternals
* @brief DBusKeyring implementation details
*
* The guts of DBusKeyring.
*
* @{
*/

/** The maximum age of a key before we create a new key to use in
 * challenges. This isn't super-reliably enforced, since system
 * clocks can change or be wrong, but we make a best effort to only
 * use keys for a short time.
 */
#define NEW_KEY_TIMEOUT_SECONDS      (60*5)
/**
 * The time after which we drop a key from the secrets file.
 * The EXPIRE_KEYS_TIMEOUT_SECONDS - NEW_KEY_TIMEOUT_SECONDS is the
minimum
 * time window a client has to complete authentication.
 */
#define EXPIRE_KEYS_TIMEOUT_SECONDS (NEW_KEY_TIMEOUT_SECONDS + (60*2))
/**
 * The maximum amount of time a key can be in the future.
 */
#define MAX_TIME_TRAVEL_SECONDS (60*5)

/**
 * Maximum number of keys in the keyring before
 * we just ignore the rest
 */
#ifdef DBUS_BUILD_TESTS
#define MAX_KEYS_IN_FILE 10
#else
#define MAX_KEYS_IN_FILE 256
#endif

/**
 * A single key from the cookie file
 */
typedef struct
{
    dbus_int32_t id; /**< identifier used to refer to the key */

    long creation_time; /**< when the key was generated,
 * as unix timestamp. signed long
 * matches struct timeval.
 */

    DBusString secret; /**< the actual key */
} DBusKey;

```

```

/**
 * @brief Internals of DBusKeyring.
 *
 * DBusKeyring internals. DBusKeyring is an opaque object, it must be
 * used via accessor functions.
 */
struct DBusKeyring
{
    int refcount;           /**< Reference count */
    DBusString directory;  /**< Directory the below two items are
inside */
    DBusString filename;   /**< Keyring filename */
    DBusString filename_lock; /**< Name of lockfile */
    DBusKey *keys;         /**< Keys loaded from the file */
    int n_keys;           /**< Number of keys */
    DBusCredentials *credentials; /**< Credentials containing user the
keyring is for */
};

static DBusKeyring*
_dbus_keyring_new (void)
{
    DBusKeyring *keyring;

    keyring = dbus_new0 (DBusKeyring, 1);
    if (keyring == NULL)
        goto out_0;

    if (!_dbus_string_init (&keyring->directory))
        goto out_1;

    if (!_dbus_string_init (&keyring->filename))
        goto out_2;

    if (!_dbus_string_init (&keyring->filename_lock))
        goto out_3;

    keyring->refcount = 1;
    keyring->keys = NULL;
    keyring->n_keys = 0;

    return keyring;

out_3:
    _dbus_string_free (&keyring->filename);
out_2:
    _dbus_string_free (&keyring->directory);
out_1:
    dbus_free (keyring);
out_0:
    return NULL;
}

```



```

static void
free_keys (DBusKey *keys,
           int      n_keys)
{
    int i;

    /* should be safe for args NULL, 0 */

    i = 0;
    while (i < n_keys)
    {
        _dbus_string_free (&keys[i].secret);
        ++i;
    }

    dbus_free (keys);
}

/* Our locking scheme is highly unreliable.  However, there is
 * unfortunately no reliable locking scheme in user home directories;
 * between bugs in Linux NFS, people using Tru64 or other total crap
 * NFS, AFS, random-file-system-of-the-week, and so forth, fcntl() in
 * homedirs simply generates tons of bug reports.  This has been
 * learned through hard experience with GConf, unfortunately.
 *
 * This bad hack might work better for the kind of lock we have here,
 * which we don't expect to hold for any length of time.  Crashing
 * while we hold it should be unlikely, and timing out such that we
 * delete a stale lock should also be unlikely except when the
 * filesystem is running really slowly.  Stuff might break in corner
 * cases but as long as it's not a security-level breakage it should
 * be OK.
 */

/** Maximum number of timeouts waiting for lock before we decide it's
    stale */
#define MAX_LOCK_TIMEOUTS 32
/** Length of each timeout while waiting for a lock */
#define LOCK_TIMEOUT_MILLISECONDS 250

static dbus_bool_t
_dbus_keyring_lock (DBusKeyring *keyring)
{
    int n_timeouts;

    n_timeouts = 0;
    while (n_timeouts < MAX_LOCK_TIMEOUTS)
    {
        DBusError error = DBUS_ERROR_INIT;

        if (_dbus_create_file_exclusively (&keyring->filename_lock,

```

```

                                &error))
        break;

        _dbus_verbose ("Did not get lock file, sleeping %d milliseconds
(%s)\n",
                        LOCK_TIMEOUT_MILLISECONDS, error.message);
        dbus_error_free (&error);

        _dbus_sleep_milliseconds (LOCK_TIMEOUT_MILLISECONDS);

        ++n_timeouts;
    }

    if (n_timeouts == MAX_LOCK_TIMEOUTS)
    {
        DBusError error = DBUS_ERROR_INIT;

        _dbus_verbose ("Lock file timed out %d times, assuming stale\n",
                        n_timeouts);

        if (!_dbus_delete_file (&keyring->filename_lock, &error))
        {
            _dbus_verbose ("Couldn't delete old lock file: %s\n",
                            error.message);
            dbus_error_free (&error);
            return FALSE;
        }

        if (!_dbus_create_file_exclusively (&keyring->filename_lock,
                                            &error))
        {
            _dbus_verbose ("Couldn't create lock file after deleting
stale one: %s\n",
                            error.message);
            dbus_error_free (&error);
            return FALSE;
        }
    }

    return TRUE;
}

static void
_dbus_keyring_unlock (DBusKeyring *keyring)
{
    DBusError error = DBUS_ERROR_INIT;

    if (!_dbus_delete_file (&keyring->filename_lock, &error))
    {
        _dbus_warn ("Failed to delete lock file: %s\n",
                    error.message);
        dbus_error_free (&error);
    }
}

```

```

    }
}

static DBusKey*
find_key_by_id (DBusKey *keys,
               int      n_keys,
               int      id)
{
    int i;

    i = 0;
    while (i < n_keys)
    {
        if (keys[i].id == id)
            return &keys[i];

        ++i;
    }

    return NULL;
}

static dbus_bool_t
add_new_key (DBusKey **keys_p,
            int      *n_keys_p,
            DBusError *error)
{
    DBusKey *new;
    DBusString bytes;
    int id;
    long timestamp;
    const unsigned char *s;
    dbus_bool_t retval;
    DBusKey *keys;
    int n_keys;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (!_dbus_string_init (&bytes))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return FALSE;
    }

    keys = *keys_p;
    n_keys = *n_keys_p;
    retval = FALSE;

    /* Generate an integer ID and then the actual key. */
    retry:

    if (!_dbus_generate_random_bytes (&bytes, 4))

```

```

    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

s = (const unsigned char*) _dbus_string_get_const_data (&bytes);

id = s[0] | (s[1] << 8) | (s[2] << 16) | (s[3] << 24);
if (id < 0)
    id = - id;
_dbus_assert (id >= 0);

if (find_key_by_id (keys, n_keys, id) != NULL)
    {
        _dbus_string_set_length (&bytes, 0);
        _dbus_verbose ("Key ID %d already existed, trying another
one\n",
                        id);
        goto retry;
    }

_dbus_verbose ("Creating key with ID %d\n", id);

#define KEY_LENGTH_BYTES 24
_dbus_string_set_length (&bytes, 0);
if (!_dbus_generate_random_bytes (&bytes, KEY_LENGTH_BYTES))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

new = dbus_realloc (keys, sizeof (DBusKey) * (n_keys + 1));
if (new == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

keys = new;
*keys_p = keys; /* otherwise *keys_p ends up invalid */
n_keys += 1;

if (!_dbus_string_init (&keys[n_keys-1].secret))
    {
        n_keys -= 1; /* we don't want to free the one we didn't init */
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

_dbus_get_real_time (&timestamp, NULL);

keys[n_keys-1].id = id;

```

```

keys[n_keys-1].creation_time = timestamp;
if (!_dbus_string_move (&bytes, 0,
                       &keys[n_keys-1].secret,
                       0))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&keys[n_keys-1].secret);
    n_keys -= 1;
    goto out;
}

retval = TRUE;

out:
*n_keys_p = n_keys;

_dbus_string_free (&bytes);
return retval;
}

/**
 * Reloads the keyring file, optionally adds one new key to the file,
 * removes all expired keys from the file iff a key was added, then
 * resaves the file. Stores the keys from the file in keyring->keys.
 * Note that the file is only resaved (written to) if a key is added,
 * this means that only servers ever write to the file and need to
 * lock it, which avoids a lot of lock contention at login time and
 * such.
 *
 * @param keyring the keyring
 * @param add_new #TRUE to add a new key to the file, expire keys, and
resave
 * @param error return location for errors
 * @returns #FALSE on failure
 */
static dbus_bool_t
_dbus_keyring_reload (DBusKeyring *keyring,
                     dbus_bool_t add_new,
                     DBusError *error)
{
    DBusString contents;
    DBusString line;
    dbus_bool_t retval;
    dbus_bool_t have_lock;
    DBusKey *keys;
    int n_keys;
    int i;
    long now;
    DBusError tmp_error;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

```

```

    if (!_dbus_check_dir_is_private_to_user (&keyring->directory,
error))
        return FALSE;

    if (!_dbus_string_init (&contents))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return FALSE;
    }

    if (!_dbus_string_init (&line))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        _dbus_string_free (&contents);
        return FALSE;
    }

    keys = NULL;
    n_keys = 0;
    retval = FALSE;
    have_lock = FALSE;

    _dbus_get_real_time (&now, NULL);

    if (add_new)
    {
        if (!_dbus_keyring_lock (keyring))
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                "Could not lock keyring file to add to it");
            goto out;
        }

        have_lock = TRUE;
    }

    dbus_error_init (&tmp_error);
    if (!_dbus_file_get_contents (&contents,
                                &keyring->filename,
                                &tmp_error))
    {
        _dbus_verbose ("Failed to load keyring file: %s\n",
            tmp_error.message);
        /* continue with empty keyring file, so we recreate it */
        dbus_error_free (&tmp_error);
    }

    if (!_dbus_string_validate_ascii (&contents, 0,
                                    _dbus_string_get_length
(&contents)))
    {

```

```

    _dbus_warn ("Secret keyring file contains non-ASCII! Ignoring
existing contents\n");
    _dbus_string_set_length (&contents, 0);
}

/* FIXME this is badly inefficient for large keyring files
 * (not that large keyring files exist outside of test suites)
 */
while (_dbus_string_pop_line (&contents, &line))
{
    int next;
    long val;
    int id;
    long timestamp;
    int len;
    int end;
    DBusKey *new;

    /* Don't load more than the max. */
    if (n_keys >= (add_new ? MAX_KEYS_IN_FILE - 1 :
MAX_KEYS_IN_FILE))
        break;

    next = 0;
    if (!_dbus_string_parse_int (&line, 0, &val, &next))
    {
        _dbus_verbose ("could not parse secret key ID at start of
line\n");
        continue;
    }

    if (val > _DBUS_INT32_MAX || val < 0)
    {
        _dbus_verbose ("invalid secret key ID at start of line\n");
        continue;
    }

    id = val;

    _dbus_string_skip_blank (&line, next, &next);

    if (!_dbus_string_parse_int (&line, next, &timestamp, &next))
    {
        _dbus_verbose ("could not parse secret key timestamp\n");
        continue;
    }

    if (timestamp < 0 ||
        (now + MAX_TIME_TRAVEL_SECONDS) < timestamp ||
        (now - EXPIRE_KEYS_TIMEOUT_SECONDS) > timestamp)
    {

```

```

        _dbus_verbose ("dropping/ignoring %ld-seconds old key with
timestamp %ld as current time is %ld\n",
                    now - timestamp, timestamp, now);
        continue;
    }

    _dbus_string_skip_blank (&line, next, &next);

    len = _dbus_string_get_length (&line);

    if ((len - next) == 0)
    {
        _dbus_verbose ("no secret key after ID and timestamp\n");
        continue;
    }

    /* We have all three parts */
    new = dbus_realloc (keys, sizeof (DBusKey) * (n_keys + 1));
    if (new == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

    keys = new;
    n_keys += 1;

    if (!_dbus_string_init (&keys[n_keys-1].secret))
    {
        n_keys -= 1; /* we don't want to free the one we didn't init
*/
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

    keys[n_keys-1].id = id;
    keys[n_keys-1].creation_time = timestamp;
    if (!_dbus_string_hex_decode (&line, next, &end,
                                &keys[n_keys-1].secret, 0))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto out;
    }

    if (_dbus_string_get_length (&line) != end)
    {
        _dbus_verbose ("invalid hex encoding in keyring file\n");
        _dbus_string_free (&keys[n_keys - 1].secret);
        n_keys -= 1;
        continue;
    }
}

```



```

    }

    if (keyring->keys)
        free_keys (keyring->keys, keyring->n_keys);
    keyring->keys = keys;
    keyring->n_keys = n_keys;
    keys = NULL;
    n_keys = 0;

    retval = TRUE;

out:
    if (have_lock)
        _dbus_keyring_unlock (keyring);

    if (! ((retval == TRUE && (error == NULL || error->name == NULL)) ||
          (retval == FALSE && (error == NULL || error->name != NULL))))
    {
        if (error && error->name)
            _dbus_verbose ("error is %s: %s\n", error->name, error-
>message);
        _dbus_warn ("returning %d but error pointer %p name %s\n",
                    retval, error, error->name ? error->name :
"(none)");
        _dbus_assert_not_reached ("didn't handle errors properly");
    }

    if (keys != NULL)
    {
        i = 0;
        while (i < n_keys)
        {
            _dbus_string_zero (&keys[i].secret);
            _dbus_string_free (&keys[i].secret);
            ++i;
        }

        dbus_free (keys);
    }

    _dbus_string_free (&contents);
    _dbus_string_free (&line);

    return retval;
}

/** @} */ /* end of internals */

/**
 * @addtogroup DBusKeyring
 *
 * @{

```

```

 */

/**
 * Increments reference count of the keyring
 *
 * @param keyring the keyring
 * @returns the keyring
 */
DBusKeyring *
_dbus_keyring_ref (DBusKeyring *keyring)
{
    keyring->refcount += 1;

    return keyring;
}

/**
 * Decrements refcount and finalizes if it reaches
 * zero.
 *
 * @param keyring the keyring
 */
void
_dbus_keyring_unref (DBusKeyring *keyring)
{
    keyring->refcount -= 1;

    if (keyring->refcount == 0)
    {
        if (keyring->credentials)
            _dbus_credentials_unref (keyring->credentials);

        _dbus_string_free (&keyring->filename);
        _dbus_string_free (&keyring->filename_lock);
        _dbus_string_free (&keyring->directory);
        free_keys (keyring->keys, keyring->n_keys);
        dbus_free (keyring);
    }
}

/**
 * Creates a new keyring that lives in the ~/.dbus-keyrings directory
 * of the given user credentials. If the credentials are #NULL or
 * empty, uses those of the current process.
 *
 * @param username username to get keyring for, or #NULL
 * @param context which keyring to get
 * @param error return location for errors
 * @returns the keyring or #NULL on error
 */
DBusKeyring*
_dbus_keyring_new_for_credentials (DBusCredentials *credentials,

```

```

                                const DBusString *context,
                                DBusError          *error)
{
    DBusString ringdir;
    DBusKeyring *keyring;
    dbus_bool_t error_set;
    DBusError tmp_error;
    DBusCredentials *our_credentials;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (_dbus_check_setuid ())
    {
        dbus_set_error_const (error, DBUS_ERROR_NOT_SUPPORTED,
                              "Unable to create DBus keyring when
setuid");
        return NULL;
    }

    keyring = NULL;
    error_set = FALSE;
    our_credentials = NULL;

    if (!_dbus_string_init (&ringdir))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return NULL;
    }

    if (credentials != NULL)
    {
        our_credentials = _dbus_credentials_copy (credentials);
    }
    else
    {
        our_credentials = _dbus_credentials_new_from_current_process ();
    }

    if (our_credentials == NULL)
        goto failed;

    if (_dbus_credentials_are_anonymous (our_credentials))
    {
        if (!_dbus_credentials_add_from_current_process
(our_credentials))
            goto failed;
    }

    if (!_dbus_append_keyring_directory_for_credentials (&ringdir,
our_credentials))
        goto failed;

```

```

keyring = _dbus_keyring_new ();
if (keyring == NULL)
    goto failed;

_dbus_assert (keyring->credentials == NULL);
keyring->credentials = our_credentials;
our_credentials = NULL; /* so we don't unref it again later */

/* should have been validated already, but paranoia check here */
if (!_dbus_keyring_validate_context (context))
{
    error_set = TRUE;
    dbus_set_error_const (error,
                        DBUS_ERROR_FAILED,
                        "Invalid context in keyring creation");
    goto failed;
}

/* Save keyring dir in the keyring object */
if (!_dbus_string_copy (&ringdir, 0,
                    &keyring->directory, 0))
    goto failed;

/* Create keyring->filename based on keyring dir and context */
if (!_dbus_string_copy (&keyring->directory, 0,
                    &keyring->filename, 0))
    goto failed;

if (!_dbus_concat_dir_and_file (&keyring->filename,
                    context))
    goto failed;

/* Create lockfile name */
if (!_dbus_string_copy (&keyring->filename, 0,
                    &keyring->filename_lock, 0))
    goto failed;

if (!_dbus_string_append (&keyring->filename_lock, ".lock"))
    goto failed;

/* Reload keyring */
dbus_error_init (&tmp_error);
if (!_dbus_keyring_reload (keyring, FALSE, &tmp_error))
{
    _dbus_verbose ("didn't load an existing keyring: %s\n",
                tmp_error.message);
    dbus_error_free (&tmp_error);
}

/* We don't fail fatally if we can't create the directory,
 * but the keyring will probably always be empty

```

```

    * unless someone else manages to create it
    */
dbus_error_init (&tmp_error);
if (!_dbus_create_directory (&keyring->directory,
                            &tmp_error))
    {
        _dbus_verbose ("Creating keyring directory: %s\n",
                       tmp_error.message);
        dbus_error_free (&tmp_error);
    }

_dbus_string_free (&ringdir);

return keyring;

failed:
if (!error_set)
    dbus_set_error_const (error,
                          DBUS_ERROR_NO_MEMORY,
                          NULL);

if (our_credentials)
    _dbus_credentials_unref (our_credentials);
if (keyring)
    _dbus_keyring_unref (keyring);
_dbus_string_free (&ringdir);
return NULL;
}

/**
 * Checks whether the context is a valid context.
 * Contexts that might cause confusion when used
 * in filenames are not allowed (contexts can't
 * start with a dot or contain dir separators).
 *
 * @todo this is the most inefficient implementation
 * imaginable.
 *
 * @param context the context
 * @returns #TRUE if valid
 */
dbus_bool_t
_dbus_keyring_validate_context (const DBusString *context)
{
    if (_dbus_string_get_length (context) == 0)
        {
            _dbus_verbose ("context is zero-length\n");
            return FALSE;
        }

    if (!_dbus_string_validate_ascii (context, 0,

```

```

                                                                    _dbus_string_get_length
(context))
{
    _dbus_verbose ("context not valid ascii\n");
    return FALSE;
}

/* no directory separators */
if (_dbus_string_find (context, 0, "/", NULL))
{
    _dbus_verbose ("context contains a slash\n");
    return FALSE;
}

if (_dbus_string_find (context, 0, "\\ ", NULL))
{
    _dbus_verbose ("context contains a backslash\n");
    return FALSE;
}

/* prevent attempts to use dotfiles or "." or ".lock"
 * all of which might allow some kind of attack
 */
if (_dbus_string_find (context, 0, ".", NULL))
{
    _dbus_verbose ("context contains a dot\n");
    return FALSE;
}

/* no spaces/tabs, those are used for separators in the protocol */
if (_dbus_string_find_blank (context, 0, NULL))
{
    _dbus_verbose ("context contains a blank\n");
    return FALSE;
}

if (_dbus_string_find (context, 0, "\n", NULL))
{
    _dbus_verbose ("context contains a newline\n");
    return FALSE;
}

if (_dbus_string_find (context, 0, "\r", NULL))
{
    _dbus_verbose ("context contains a carriage return\n");
    return FALSE;
}

return TRUE;
}

static DBusKey*

```

```

find_recent_key (DBusKeyring *keyring)
{
    int i;
    long tv_sec, tv_usec;

    _dbus_get_real_time (&tv_sec, &tv_usec);

    i = 0;
    while (i < keyring->n_keys)
    {
        DBusKey *key = &keyring->keys[i];

        _dbus_verbose ("Key %d is %ld seconds old\n",
                       i, tv_sec - key->creation_time);

        if ((tv_sec - NEW_KEY_TIMEOUT_SECONDS) < key->creation_time)
            return key;

        ++i;
    }

    return NULL;
}

/**
 * Gets a recent key to use for authentication.
 * If no recent key exists, creates one. Returns
 * the key ID. If a key can't be written to the keyring
 * file so no recent key can be created, returns -1.
 * All valid keys are > 0.
 *
 * @param keyring the keyring
 * @param error error on failure
 * @returns key ID to use for auth, or -1 on failure
 */
int
_dbus_keyring_get_best_key (DBusKeyring *keyring,
                            DBusError *error)
{
    DBusKey *key;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    key = find_recent_key (keyring);
    if (key)
        return key->id;

    /* All our keys are too old, or we've never loaded the
     * keyring. Create a new one.
     */
    if (!_dbus_keyring_reload (keyring, TRUE,
                               error))

```



```

    return -1;

    key = find_recent_key (keyring);
    if (key)
        return key->id;
    else
        {
            dbus_set_error_const (error,
                                  DBUS_ERROR_FAILED,
                                  "No recent-enough key found in keyring,
and unable to create a new key");
            return -1;
        }
}

/**
 * Checks whether the keyring is for the same user as the given
credentials.
 *
 * @param keyring the keyring
 * @param credentials the credentials to check
 *
 * @returns #TRUE if the keyring belongs to the given user
 */
dbus_bool_t
_dbus_keyring_is_for_credentials (DBusKeyring          *keyring,
                                  DBusCredentials      *credentials)
{
    return _dbus_credentials_same_user (keyring->credentials,
                                        credentials);
}

/**
 * Gets the hex-encoded secret key for the given ID.
 * Returns #FALSE if not enough memory. Returns #TRUE
 * but empty key on any other error such as unknown
 * key ID.
 *
 * @param keyring the keyring
 * @param key_id the key ID
 * @param hex_key string to append hex-encoded key to
 * @returns #TRUE if we had enough memory
 */
dbus_bool_t
_dbus_keyring_get_hex_key (DBusKeyring          *keyring,
                           int                  key_id,
                           DBusString          *hex_key)
{
    DBusKey *key;

    key = find_key_by_id (keyring->keys,
                          keyring->n_keys,

```

```

        key_id);
if (key == NULL)
    return TRUE; /* had enough memory, so TRUE */

return _dbus_string_hex_encode (&key->secret, 0,
                                hex_key,
                                _dbus_string_get_length (hex_key));
}

/** @} */ /* end of exposed API */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

dbus_bool_t
_dbus_keyring_test (void)
{
    DBusString context;
    DBusKeyring *ring1;
    DBusKeyring *ring2;
    int id;
    DBusError error;
    int i;

    ring1 = NULL;
    ring2 = NULL;

    /* Context validation */

    _dbus_string_init_const (&context, "foo");
    _dbus_assert (_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "org_freedesktop_blah");
    _dbus_assert (_dbus_keyring_validate_context (&context));

    _dbus_string_init_const (&context, "");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, ".foo");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "bar.foo");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "bar/foo");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "bar\\foo");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "foo\xfa\xfo");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "foo\x80");
    _dbus_assert (!_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "foo\x7f");
    _dbus_assert (_dbus_keyring_validate_context (&context));
    _dbus_string_init_const (&context, "foo bar");

```

```

_dbus_assert (!_dbus_keyring_validate_context (&context));

if (!_dbus_string_init (&context))
    _dbus_assert_not_reached ("no memory");
if (!_dbus_string_append_byte (&context, '\0'))
    _dbus_assert_not_reached ("no memory");
_dbus_assert (!_dbus_keyring_validate_context (&context));
_dbus_string_free (&context);

/* Now verify that if we create a key in keyring 1,
 * it is properly loaded in keyring 2
 */

_dbus_string_init_const (&context,
"org_freedesktop_dbus_testsuite");
dbus_error_init (&error);
ring1 = _dbus_keyring_new_for_credentials (NULL, &context,
                                           &error);

_dbus_assert (ring1 != NULL);
_dbus_assert (error.name == NULL);

id = _dbus_keyring_get_best_key (ring1, &error);
if (id < 0)
{
    fprintf (stderr, "Could not load keyring: %s\n", error.message);
    dbus_error_free (&error);
    goto failure;
}

ring2 = _dbus_keyring_new_for_credentials (NULL, &context, &error);
_dbus_assert (ring2 != NULL);
_dbus_assert (error.name == NULL);

if (ring1->n_keys != ring2->n_keys)
{
    fprintf (stderr, "Different number of keys in keyrings\n");
    goto failure;
}

/* We guarantee we load and save keeping keys in a fixed
 * order
 */
i = 0;
while (i < ring1->n_keys)
{
    if (ring1->keys[i].id != ring2->keys[i].id)
    {
        fprintf (stderr, "Keyring 1 has first key ID %d and keyring
2 has %d\n",
                ring1->keys[i].id, ring2->keys[i].id);
        goto failure;
    }
}

```

```

        if (ring1->keys[i].creation_time != ring2-
>keys[i].creation_time)
        {
            fprintf (stderr, "Keyring 1 has first key time %ld and
keyring 2 has %ld\n",
                    ring1->keys[i].creation_time, ring2-
>keys[i].creation_time);
            goto failure;
        }

        if (!_dbus_string_equal (&ring1->keys[i].secret,
                                &ring2->keys[i].secret))
        {
            fprintf (stderr, "Keyrings 1 and 2 have different secrets
for same ID/timestamp\n");
            goto failure;
        }

        ++i;
    }

    printf (" %d keys in test\n", ring1->n_keys);

    /* Test ref/unref */
    _dbus_keyring_ref (ring1);
    _dbus_keyring_ref (ring2);
    _dbus_keyring_unref (ring1);
    _dbus_keyring_unref (ring2);

    /* really unref */
    _dbus_keyring_unref (ring1);
    _dbus_keyring_unref (ring2);

    return TRUE;

failure:
    if (ring1)
        _dbus_keyring_unref (ring1);
    if (ring2)
        _dbus_keyring_unref (ring2);

    return FALSE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-keyring.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-keyring.h Store secret cookies in your homedir
 *
 * Copyright (C) 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifdef DBUS_KEYRING_H
#define DBUS_KEYRING_H

#include <dbus/dbus-macros.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-credentials.h>

DBUS_BEGIN_DECLS

typedef struct DbusKeyring DbusKeyring;

DBusKeyring* _dbus_keyring_new_for_credentials (DBusCredentials
*credentials,
                                           const DbusString
*context,
                                           DbusError
*error);
DBusKeyring* _dbus_keyring_ref (DBusKeyring
*keyring);
void _dbus_keyring_unref (DBusKeyring
*keyring);
dbus_bool_t _dbus_keyring_validate_context (const DbusString
*context);
int _dbus_keyring_get_best_key (DBusKeyring
*keyring,

```

```

DBusError
*error);
dbus_bool_t _dbus_keyring_is_for_credentials (DBusKeyring
*keyring,
DBusCredentials
*credentials);
dbus_bool_t _dbus_keyring_get_hex_key (DBusKeyring
*keyring,
int
key_id,
DBusString
*hex_key);

DBUS_END_DECLS

#endif /* DBUS_KEYRING_H */

File = dbus-launch-win.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-launch-win.c  dbus-launch utility
 *
 * Copyright (C) 2007 Ralf Habacker <ralf.habacker@freenet.de>
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#ifdef UNICOD
#define UNICOD 1
#endif

```

```

#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>

/* Save string functions.  Instead of figuring out the exact _MSC_VER
   that work, override for everybody.  */

#define errno_t int
#define wcsat_s my_wcsat_s
#define wcsncpy_s my_wcsncpy_s

static errno_t
wcsat_s (wchar_t *dest, size_t size, wchar_t *src)
{
    assert (sizeof (wchar_t) * (wcslen (dest) + wcslen (src) + 1) <=
size);
    wcsat (dest, src);
    return 0;
}

static errno_t
wcsncpy_s (wchar_t *dest, size_t size, wchar_t *src)
{
    assert (sizeof (wchar_t) * (wcslen (src) + 1) <= size);
    wcsncpy (dest, src);
    return 0;
}

/* TODO (tl): This Windows version of dbus-launch is currety rather
 * pointless as it doesn't take the same command-line options as the
 * UNIX dbus-launch does. A main point of the dbus-launch command is
 * to pass it for instance a --config-file option to make the started
 * dbus-daemon use that config file.
 *
 * This version also doesn't print out any information, which is a
 * main point of the UNIX one. It should at least support the
 * --sh-syntax option, and maybe also a --cmd-syntax to print out the
 * variable settings in cmd.exe syntax?
 *
 * NOTE (rh): The main task of dbus-launch is (from the man page) to
start
 * a session bus and this is achieved by the current implementation.
 *
 * Additional on windows the session bus starting in not integrated
 * into the logon process, so there is no need for any --syntax
option.
 * In fact (at least for kde on windows) the session bus is
autostarted
 * with the first application requesting a session bus.
 *

```

```

*/

#define AUTO_ACTIVATE_CONSOLE_WHEN_VERBOSE_MODE 1

#define DIM(x) (sizeof(x) / sizeof(x[0]))
#define WCSTRINGIFY_(x) L ## x
#define WCSTRINGIFY(x) WCSTRINGIFY_(x)

int
main (int argc, char **argv)
{
    wchar_t dbusDaemonPath[MAX_PATH * 2 + 1];
    wchar_t command[MAX_PATH * 2 + 1];
    wchar_t *p;
    wchar_t *daemon_name;
    int result;
    int showConsole = 0;
#ifdef DBUS_WINCE
    char *s = NULL;
#else
    char *s = getenv("DBUS_VERBOSE");
#endif
    int verbose = s && *s != '\0' ? 1 : 0;

    PROCESS_INFORMATION pi;
    STARTUPINFO si;
    BOOL inherit = TRUE;
    DWORD flags = 0;

#ifdef AUTO_ACTIVATE_CONSOLE_WHEN_VERBOSE_MODE
    if (verbose)
        showConsole = 1;
#endif
    GetModuleFileNameW (NULL, dbusDaemonPath, DIM (dbusDaemonPath));

    daemon_name = WCSTRINGIFY(DBUS_DAEMON_NAME) L".exe";

    if ((p = wcsrchr (dbusDaemonPath, L'\\'))
        {
            p[1] = L'\0';
            wcscat_s (dbusDaemonPath, sizeof (dbusDaemonPath), daemon_name);
        }
    else
        {
            if (verbose)
                fprintf (stderr, "error: could not extract path from current
"
                        "applications module filename\n");
            return 1;
        }
}

#ifdef DBUS_WINCE

```



```

    /* Windows CE has a different interpretation of cmdline: Start with
    argv[1]. */
    wcsncpy_s (command, sizeof (command), L"--session");
    if (verbose)
        fprintf (stderr, "%ls %ls\n", dbusDaemonPath, command);
#else
    command[0] = L'\0';
    /* Windows cmdline starts with path, which can contain spaces. */
    wcsncpy_s (command, sizeof (command), L"\");
    wscat_s (command, sizeof (command), dbusDaemonPath);
    wscat_s (command, sizeof (command), L"\ --session");
    if (verbose)
        fprintf (stderr, "%ls\n", command);
#endif

    memset (&si, 0, sizeof (si));
    memset (&pi, 0, sizeof (pi));
    si.cb = sizeof (si);

    if (verbose)
        flags |= CREATE_NEW_CONSOLE;

#ifdef DBUS_WINCE
    inherit = FALSE;
#else
    flags |= NORMAL_PRIORITY_CLASS;
    if (!verbose)
        flags |= DETACHED_PROCESS;
#endif

    result = CreateProcessW (dbusDaemonPath, command, 0, 0,
                            inherit, flags, 0, 0, &si, &pi);

    if (result == 0)
    {
        if (verbose)
            fprintf (stderr, "Could not start " DBUS_DAEMON_NAME ".
error=%d\n",
                    GetLastError ());
        return 4;
    }

    CloseHandle (pi.hProcess);
    CloseHandle (pi.hThread);

    return 0;
}

```

File = dbus-launch-x11.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-launch.h  dbus-launch utility
 *
 * Copyright (C) 2006 Thiago Macieira <thiago@kde.org>
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 *
 */
```

```
#include <config.h>
#include "dbus-launch.h"
```

```
#ifdef DBUS_BUILD_X11
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <stdio.h>
#include <string.h>
#include <pwd.h>
#include <X11/Xlib.h>
#include <X11/Xatom.h>
```

```
Display *xdisplay = NULL;
static Atom selection_atom;
static Atom address_atom;
static Atom pid_atom;
```

```
static int
x_io_error_handler (Display *xdisplay)
{
    verbose ("X IO error\n");
    kill_bus_and_exit (0);
}
```

```

    return 0;
}

static void
remove_prefix (char *s,
               char *prefix)
{
    int plen;

    plen = strlen (prefix);

    if (strncmp (s, prefix, plen) == 0)
        {
            memmove (s, s + plen, strlen (s) - plen + 1);
        }
}

static const char*
get_homedir (void)
{
    const char *home;

    home = getenv ("HOME");
    if (home == NULL)
        {
            /* try from the user database */
            struct passwd *user = getpwuid (getuid());
            if (user != NULL)
                home = user->pw_dir;
        }

    if (home == NULL)
        {
            fprintf (stderr, "Can't get user home directory\n");
            exit (1);
        }

    return home;
}

#define DBUS_DIR ".dbus"
#define DBUS_SESSION_BUS_DIR "session-bus"

static char *
get_session_file (void)
{
    static const char prefix[] = "/" DBUS_DIR "/" DBUS_SESSION_BUS_DIR
"/";
    const char *machine;
    const char *home;
    char *display;
    char *result;

```

```

char *p;

machine = get_machine_uuid ();
if (machine == NULL)
    return NULL;

display = xstrdup (getenv ("DISPLAY"));
if (display == NULL)
    {
        verbose ("X11 integration disabled because X11 is not
running\n");
        return NULL;
    }

/* remove the screen part of the display name */
p = strrchr (display, ':');
if (p != NULL)
    {
        for ( ; *p; ++p)
            {
                if (*p == '.')
                    {
                        *p = '\\0';
                        break;
                    }
            }
    }

/* Note that we leave the hostname in the display most of the
* time. The idea is that we want to be per-(machine,display,user)
* triplet to be extra-sure we get a bus we can connect to. Ideally
* we'd recognize when the hostname matches the machine we're on in
* all cases; we do try to drop localhost and localhost.localdomain
* as a special common case so that alternate spellings of DISPLAY
* don't result in extra bus instances.
*
* We also kill the ":" if there's nothing in front of it. This
* avoids an ugly double underscore in the filename.
*/
remove_prefix (display, "localhost.localdomain:");
remove_prefix (display, "localhost:");
remove_prefix (display, ":");

/* replace the : in the display with _ if the : is still there.
* use _ instead of - since it can't be in hostnames.
*/
for (p = display; *p; ++p)
    {
        if (*p == ':')
            *p = '_';
    }

```

```

home = get_homedir ();

result = malloc (strlen (home) + strlen (prefix) + strlen (machine)
+
                strlen (display) + 2);
if (result == NULL)
{
    /* out of memory */
    free (display);
    return NULL;
}

strcpy (result, home);
strcat (result, prefix);
strcat (result, machine);
strcat (result, "-");
strcat (result, display);
free (display);

verbose ("session file: %s\n", result);
return result;
}

static void
ensure_session_directory (void)
{
    const char *home;
    char *dir;

    home = get_homedir ();

    /* be sure we have space for / and nul */
    dir = malloc (strlen (home) + strlen (DBUS_DIR) + strlen
(DBUS_SESSION_BUS_DIR) + 3);
    if (dir == NULL)
    {
        fprintf (stderr, "no memory\n");
        exit (1);
    }

    strcpy (dir, home);
    strcat (dir, "/");
    strcat (dir, DBUS_DIR);

    if (mkdir (dir, 0700) < 0)
    {
        /* only print a warning here, writing the session file itself
will fail later */
        if (errno != EEXIST)
            fprintf (stderr, "Unable to create %s\n", dir);
    }
}

```

```

strcat (dir, "/");
strcat (dir, DBUS_SESSION_BUS_DIR);

if (mkdir (dir, 0700) < 0)
{
    /* only print a warning here, writing the session file itself
will fail later */
    if (errno != EEXIST)
        fprintf (stderr, "Unable to create %s\n", dir);
}

free (dir);
}

static Display *
open_x11 (void)
{
    if (xdisplay != NULL)
        return xdisplay;

    xdisplay = XOpenDisplay (NULL);
    if (xdisplay != NULL)
    {
        verbose ("Connected to X11 display '%s'\n", DisplayString
(xdisplay));
        XSetIOErrorHandler (x_io_error_handler);
    }
    return xdisplay;
}

static int
init_x_atoms (Display *display)
{
    static const char selection_prefix[] =
"_DBUS_SESSION_BUS_SELECTION_";
    static const char address_prefix[] = "_DBUS_SESSION_BUS_ADDRESS";
    static const char pid_prefix[] = "_DBUS_SESSION_BUS_PID";
    static int init = FALSE;
    char *atom_name;
    const char *machine;
    char *user_name;
    struct passwd *user;

    if (init)
        return TRUE;

    machine = get_machine_uid ();
    if (machine == NULL)
        return FALSE;

    user = getpwuid (getuid ());
    if (user == NULL)

```

```

        {
            verbose ("Could not determine user information; aborting X11
integration.\n");
            return FALSE;
        }
        user_name = xstrdup(user->pw_name);

        atom_name = malloc (strlen (machine) + strlen (user_name) + 2 +
                            MAX (strlen (selection_prefix),
                                MAX (strlen (address_prefix),
                                    strlen (pid_prefix))));

        if (atom_name == NULL)
        {
            verbose ("Could not create X11 atoms; aborting X11
integration.\n");
            free (user_name);
            return FALSE;
        }

        /* create the selection atom */
        strcpy (atom_name, selection_prefix);
        strcat (atom_name, user_name);
        strcat (atom_name, "_");
        strcat (atom_name, machine);
        selection_atom = XInternAtom (display, atom_name, FALSE);

        /* create the address property atom */
        strcpy (atom_name, address_prefix);
        address_atom = XInternAtom (display, atom_name, FALSE);

        /* create the PID property atom */
        strcpy (atom_name, pid_prefix);
        pid_atom = XInternAtom (display, atom_name, FALSE);

        free (atom_name);
        free (user_name);
        init = TRUE;
        return TRUE;
    }

    /*
    * Gets the daemon address from the X11 display.
    * Returns FALSE if there was an error. Returning
    * TRUE does not mean the address exists.
    */
    int
    x11_get_address (char **paddress, pid_t *pid, long *wid)
    {
        int result;
        Atom type;
        Window owner;
        int format;
    }

```

```

unsigned long items;
unsigned long after;
char *data;

*ppaddress = NULL;

/* locate the selection owner */
owner = XGetSelectionOwner (xdisplay, selection_atom);
if (owner == None)
    return TRUE;                /* no owner */
if (wid != NULL)
    *wid = (long) owner;

/* get the bus address */
result = XGetWindowProperty (xdisplay, owner, address_atom, 0, 1024,
False,
                                XA_STRING, &type, &format, &items,
&after,
                                (unsigned char **) &data);
if (result != Success || type == None || after != 0 || data == NULL
|| format != 8)
    return FALSE;                /* error */

*ppaddress = xstrdup (data);
XFree (data);

/* get the PID */
if (pid != NULL)
{
    *pid = 0;
    result = XGetWindowProperty (xdisplay, owner, pid_atom, 0,
sizeof pid, False,
                                XA_CARDINAL, &type, &format,
&items, &after,
                                (unsigned char **) &data);
    if (result == Success && type != None && after == 0 && data !=
NULL && format == 32)
        *pid = (pid_t) *(long*) data;
    XFree (data);
}

return TRUE;                /* success */
}

/*
 * Saves the address in the X11 display. Returns 0 on success.
 * If an error occurs, returns -1. If the selection already exists,
 * returns 1. (i.e. another daemon is already running)
 */
static Window
set_address_in_x11(char *address, pid_t pid)
{

```



```

char *current_address;
Window wid = None;
unsigned long pid32; /* Xlib property functions want _long_ not 32-
bit for format "32" */

/* lock the X11 display to make sure we're doing this atomically */
XGrabServer (xdisplay);

if (!x11_get_address (&current_address, NULL, NULL))
{
    /* error! */
    goto out;
}

if (current_address != NULL)
{
    /* someone saved the address in the meantime */
    free (current_address);
    goto out;
}

/* Create our window */
wid = XCreateWindow (xdisplay, RootWindow (xdisplay, 0), -20, -20,
10, 10,
                    0, CopyFromParent, InputOnly, CopyFromParent,
                    0, NULL);
verbose ("Created window %d\n", wid);

/* Save the property in the window */
XChangeProperty (xdisplay, wid, address_atom, XA_STRING, 8,
PropModeReplace,
                (unsigned char *)address, strlen (address));
pid32 = pid;
XChangeProperty (xdisplay, wid, pid_atom, XA_CARDINAL, 32,
PropModeReplace,
                (unsigned char *)&pid32, 1);

/* Now grab the selection */
XSetSelectionOwner (xdisplay, selection_atom, wid, CurrentTime);

out:
/* Ungrab the server to let other people use it too */
XUngrabServer (xdisplay);

/* And make sure that the ungrab gets sent to X11 */
XFlush (xdisplay);

return wid;
}

/*
* Saves the session address in session file. Returns TRUE on

```

```

* success, FALSE if an error occurs.
*/
static int
set_address_in_file (char *address, pid_t pid, Window wid)
{
    char *session_file;
    FILE *f;

    ensure_session_directory ();
    session_file = get_session_file();
    if (session_file == NULL)
        return FALSE;

    f = fopen (session_file, "w");
    free (session_file);
    if (f == NULL)
        return FALSE;          /* some kind of error */
    fprintf (f,
        "# This file allows processes on the machine with id %s
using \n"
        "# display %s to find the D-Bus session bus with the below
address.\n"
        "# If the DBUS_SESSION_BUS_ADDRESS environment variable is
set, it will\n"
        "# be used rather than this file.\n"
        "# See \"man dbus-launch\" for more details.\n"
        "DBUS_SESSION_BUS_ADDRESS=%s\n"
        "DBUS_SESSION_BUS_PID=%ld\n"
        "DBUS_SESSION_BUS_WINDOWID=%ld\n",
        get_machine_uuid (),
        getenv ("DISPLAY"),
        address, (long)pid, (long)wid);

    fclose (f);

    return TRUE;
}

int
x11_save_address (char *address, pid_t pid, long *wid)
{
    Window id = set_address_in_x11 (address, pid);
    if (id != None)
    {
        if (!set_address_in_file (address, pid, id))
            return FALSE;

        if (wid != NULL)
            *wid = (long) id;
        return TRUE;
    }
    return FALSE;
}

```

```

}

int
x11_init (void)
{
    return open_x11 () != NULL && init_x_atoms (xdisplay);
}

void
x11_handle_event (void)
{
    if (xdisplay != NULL)
    {
        while (XPending (xdisplay))
        {
            XEvent ignored;
            XNextEvent (xdisplay, &ignored);
        }
    }
}

#else
void dummy_dbus_launch_x11 (void);

void dummy_dbus_launch_x11 (void) { }
#endif

```

File = dbus-launch.1

```

.\"
.\" dbus\--launch manual page.
.\" Copyright (C) 2003 Red Hat, Inc.
.\"
.TH dbus\--launch 1
.SH NAME
dbus\--launch \-- Utility to start a message bus from a shell script
.SH SYNOPSIS
.SH PP
.B dbus\--launch [\--version] [\--sh\--syntax] [\--csh\--syntax] [\--\--
auto\--syntax] [\--\--exit\--with\--session] [\--\--autolaunch=MACHINEID] [\--
\--config\--file=FILENAME] [PROGRAM] [ARGS...]

```

.SH DESCRIPTION

The `\fIdbus\--launch\fp` command is used to start a session bus instance of `\fIdbus\--daemon\fp` from a shell script. It would normally be called from a user's login scripts. Unlike the daemon itself, `\fIdbus\--launch\fp` exits, so backticks or the `$()` construct can be used to read information from `\fIdbus\--launch\fp`.

With no arguments, `\fIdbus\-\launch\fp` will launch a session bus instance and print the address and PID of that instance to standard output.

You may specify a program to be run; in this case, `\fIdbus\-\launch\fp` will launch a session bus instance, set the appropriate environment variables so the specified program can find the bus, and then execute the specified program, with the specified arguments. See below for examples.

If you launch a program, `\fIdbus\-\launch\fp` will not print the information about the new bus to standard output.

When `\fIdbus\-\launch\fp` prints bus information to standard output, by default it is in a simple key-value pairs format. However, you may request several alternate syntaxes using the `\-\-sh\-\syntax`, `\-\-csh\-\syntax`, `\-\-binary\-\syntax`, or `\-\-auto\-\syntax` options. Several of these cause `\fIdbus\-\launch\fp` to emit shell code to set up the environment.

With the `\-\-auto\-\syntax` option, `\fIdbus\-\launch\fp` looks at the value of the SHELL environment variable to determine which shell syntax should be used. If SHELL ends in "csh", then csh-compatible code is emitted; otherwise Bourne shell code is emitted. Instead of passing `\-\-auto\-\syntax`, you may explicitly specify a particular one by using `\-\-sh\-\syntax` for Bourne syntax, or `\-\-csh\-\syntax` for csh syntax. In scripts, it's more robust to avoid `\-\-auto\-\syntax` and you hopefully know which shell your script is written in.

.PP

See <http://www.freedesktop.org/software/dbus/> for more information about D-Bus. See also the man page for `\fIdbus\-\daemon\fp`.

.SH EXAMPLES

Distributions running

`.B dbus\-\launch`

as part of a standard X session should run

`.B "dbus\-\launch \-\-exit\-\with\-\session"`

after the X server has started and become available, as a wrapper around

the "main" X client (typically a session manager or window manager), as in

these examples:

.RS

```
.B "dbus\--launch \--exit\--with\--session gnome\--session"
.B "dbus\--launch \--exit\--with\--session openbox"
.B "dbus\--launch \--exit\--with\--session ~/.xsession"
.RE
```

If your distribution does not do this, you can achieve similar results by running your session or window manager in the same way in a script run by your X session, such as

```
.BR ~/.xsession ,
.B ~/.xinitrc
or
.BR ~/.Xclients .
```

To start a D-Bus session within a text-mode session, you can run `dbus-launch` in the background. For instance, in a `sh-compatible` shell:

```
.nf
  ## test for an existing bus daemon, just to be safe
  if test \-z "$DBUS_SESSION_BUS_ADDRESS" ; then
    ## if not found, launch a new one
    eval `dbus\--launch \--sh\--syntax`
    echo "D\--Bus per\--session daemon address is:
$DBUS_SESSION_BUS_ADDRESS"
  fi
.fi
```

Note that in this case, `dbus-launch` will exit, and `dbus-daemon` will not be terminated automatically on logout.

.SH AUTOMATIC LAUNCHING

.PP

If `DBUS_SESSION_BUS_ADDRESS` is not set for a process that tries to use `D\--Bus`, by default the process will attempt to invoke `dbus\--launch` with the `\--autolaunch` option to start up a new session bus or find the existing bus address on the X display or in a file in `~/.dbus/session\--bus/`

.PP

Whenever an autolaunch occurs, the application that had to start a new bus will be in its own little world; it can effectively end up starting a whole new session if it tries to use a lot of bus services. This can be suboptimal or even totally broken, depending on the app and what it tries to do.

.PP

There are two common reasons for autolaunch. One is `ssh` to a remote machine. The ideal fix for that would be forwarding of `DBUS_SESSION_BUS_ADDRESS` in the same way that `DISPLAY` is forwarded.

In the meantime, you can edit the session.conf config file to have your session bus listen on TCP, and manually set DBUS_SESSION_BUS_ADDRESS, if you like.

.PP

The second common reason for autolaunch is an su to another user, and display of X applications running as the second user on the display belonging to the first user. Perhaps the ideal fix in this case would be to allow the second user to connect to the session bus of the first user, just as they can connect to the first user's display. However, a mechanism for that has not been coded.

.PP

You can always avoid autolaunch by manually setting DBUS_SESSION_BUS_ADDRESS. Autolaunch happens because the default address if none is set is "autolaunch:", so if any other address is set there will be no autolaunch. You can however include autolaunch in an explicit session bus address as a fallback, for example DBUS_SESSION_BUS_ADDRESS="something:,autolaunch:" \- in that case if the first address doesn't work, processes will autolaunch. (The bus address variable contains a comma\-separated list of addresses to try.)

.PP

The \-\-autolaunch option is considered an internal implementation detail of libdbus, and in fact there are plans to change it. There's no real reason to use it outside of the libdbus implementation anyhow.

.SH OPTIONS

The following options are supported:

.TP

.I "\-\-auto\-\-syntax"

Choose \-\-csh\-\-syntax or \-\-sh\-\-syntax based on the SHELL environment variable.

.I "\-\-binary\-\-syntax"

Write to stdout a nul\-terminated bus address, then the bus PID as a binary integer of size sizeof(pid_t), then the bus X window ID as a binary integer of size sizeof(long). Integers are in the machine's byte order, not network byte order or any other canonical byte order.

.TP

.I "\-\-close\-\-stderr"

Close the standard error output stream before starting the D\-\-Bus daemon. This is useful if you want to capture dbus\-\-launch error messages but you don't want dbus\-\-daemon to keep the stream open to your application.

.TP

.I "\-\-config\-\-file=FILENAME"

Pass \-\-config\-\-file=FILENAME to the bus daemon, instead of passing it

the `--session` argument. See the man page for `dbus-daemon`

.TP

.I "--csh-syntax"

Emit `cs` compatible code to set up environment variables.

.TP

.I "--exit-with-session"

If this option is provided, a persistent "babysitter" process will be created that watches `stdin` for HUP and tries to connect to the X server. If this process gets a HUP on `stdin` or loses its X connection, it kills the message bus daemon.

.TP

.I "--autolaunch=MACHINEID"

This option implies that `dbus-launch` should scan for a previously-started session and reuse the values found there. If no session is found, it will start a new session. The `--exit-with-session` option is implied if `--autolaunch` is given. This option is for the exclusive use of `libdbus`, you do not want to use it manually. It may change in the future.

.TP

.I "--sh-syntax"

Emit Bourne-shell compatible code to set up environment variables.

.TP

.I "--version"

Print the version of `dbus-launch`

.SH NOTES

If you run

.B "dbus-launch myapp"

(with any other options), `dbus-daemon` will

.I not

exit when

.B myapp

terminates: this is because

.B myapp

is assumed to be part of a larger session, rather than a session in its

own right.

.SH AUTHOR

See <http://www.freedesktop.org/software/dbus/doc/AUTHORS>

.SH BUGS

Please send bug reports to the D-Bus mailing list or bug tracker, see <http://www.freedesktop.org/software/dbus/>

File = dbus-launch.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-launch.c  dbus-launch utility
 *
 * Copyright (C) 2003, 2006 Red Hat, Inc.
 * Copyright (C) 2006 Thiago Macieira <thiago@kde.org>
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 *
 */

#include <config.h>
#include "dbus-launch.h"
#include <stdlib.h>
#include <ctype.h>
#include <unistd.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/wait.h>
#include <errno.h>
#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <stdarg.h>
#include <sys/select.h>
#include <time.h>

#ifdef DBUS_BUILD_X11
#include <X11/Xlib.h>
extern Display *xdisplay;
#endif

/* PROCESSES
```



```

*
* If you are in a shell and run "dbus-launch myapp", here is what
happens:
*
* shell [*]
*   \- main()                --exec--> myapp[*]
*     \- "intermediate parent"
*       \- bus-runner        --exec--> dbus-daemon --fork
*         \- babysitter[*]   \- final dbus-daemon[*]
*
* Processes marked [*] survive the initial flurry of activity.
*
* If you run "dbus-launch --sh-syntax" then the diagram is the same,
except
* that main() prints variables and exits 0 instead of exec'ing myapp.
*
* PIPES
*
* dbus-daemon --print-pid      -> bus_pid_to_launcher_pipe      -> main
* dbus-daemon --print-address -> bus_address_to_launcher_pipe -> main
* main                        -> bus_pid_to_babysitter_pipe  ->
babysitter
*
* The intermediate parent looks pretty useless at first glance. Its
purpose
* is to avoid the bus-runner becoming a zombie: when the intermediate
parent
* terminates, the bus-runner and babysitter are reparented to init,
which
* reaps them if they have finished. We can't rely on main() to reap
arbitrary
* children because it might exec myapp, after which it can't be
relied on to
* reap its children. We can rely on main() to reap the intermediate
parent,
* because that happens before it execs myapp.
*
* It's unclear why dbus-daemon needs to fork, but we explicitly tell
it to
* for some reason, then wait for it. If we left it undefined, a
forking
* dbus-daemon would get the parent process reparented to init and
reaped
* when the intermediate parent terminated, and a non-forking dbus-
daemon
* would get reparented to init and carry on there.
*
* myapp is exec'd by the process that initially ran main() so that
it's
* the shell's child, so the shell knows how to do job control and
stuff.

```

```
 * This is desirable for the "dbus-launch an application" use-case,  
less so  
 * for the "dbus-launch a test suite in an isolated session" use-case.  
 */
```

```
static char* machine_uuid = NULL;
```

```
const char*  
get_machine_uuid (void)  
{  
    return machine_uuid;  
}
```

```
static void  
save_machine_uuid (const char *uuid_arg)  
{  
    if (strlen (uuid_arg) != 32)  
    {  
        fprintf (stderr, "machine ID '%s' looks like it's the wrong  
length, should be 32 hex digits",  
                uuid_arg);  
        exit (1);  
    }  
}
```

```
    machine_uuid = xstrdup (uuid_arg);  
}
```

```
#define UUID_MAXLEN 40
```

```
/* Read the machine uuid from file if needed. Returns TRUE if  
machine_uuid is
```

```
 * set after this function */
```

```
static int  
read_machine_uuid_if_needed (void)  
{
```

```
    FILE *f;  
    char uuid[UUID_MAXLEN];  
    size_t len;  
    int ret = FALSE;
```

```
    if (machine_uuid != NULL)  
        return TRUE;
```

```
    f = fopen (DBUS_MACHINE_UUID_FILE, "r");  
    if (f == NULL)  
        return FALSE;
```

```
    if (fgets (uuid, UUID_MAXLEN, f) == NULL)  
        goto out;
```

```
    len = strlen (uuid);  
    if (len < 32)  
        goto out;
```

```

/* rstrip the read uuid */
while (len > 31 && isspace(uuid[len - 1]))
    len--;

if (len != 32)
    goto out;

uuid[len] = '\0';
machine_uuid = xstrdup (uuid);
verbose ("UID: %s\n", machine_uuid);
ret = TRUE;

out:
    fclose(f);
    return ret;
}

void
verbose (const char *format,
        ...)
{
    va_list args;
    static int verbose = TRUE;
    static int verbose_initted = FALSE;

    /* things are written a bit oddly here so that
     * in the non-verbose case we just have the one
     * conditional and return immediately.
     */
    if (!verbose)
        return;

    if (!verbose_initted)
    {
        verbose = getenv ("DBUS_VERBOSE") != NULL;
        verbose_initted = TRUE;
        if (!verbose)
            return;
    }

    fprintf (stderr, "%lu: ", (unsigned long) getpid ());

    va_start (args, format);
    vfprintf (stderr, format, args);
    va_end (args);
}

static void
usage (int ecode)
{

```

```

    fprintf (stderr, "dbus-launch [--version] [--help] [--sh-syntax] [--
csh-syntax] [--auto-syntax] [--exit-with-session]\n");
    exit (ecode);
}

static void
version (void)
{
    printf ("D-Bus Message Bus Launcher %s\n"
           "Copyright (C) 2003 Red Hat, Inc.\n"
           "This is free software; see the source for copying
conditions.\n"
           "There is NO warranty; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.\n",
           VERSION);
    exit (0);
}

char *
xstrdup (const char *str)
{
    int len;
    char *copy;

    if (str == NULL)
        return NULL;

    len = strlen (str);

    copy = malloc (len + 1);
    if (copy == NULL)
        return NULL;

    memcpy (copy, str, len + 1);

    return copy;
}

typedef enum
{
    READ_STATUS_OK,      /**< Read succeeded */
    READ_STATUS_ERROR,  /**< Some kind of error */
    READ_STATUS_EOF     /**< EOF returned */
} ReadStatus;

static ReadStatus
read_line (int      fd,
           char     *buf,
           size_t   maxlen)
{
    size_t bytes = 0;
    ReadStatus retval;

```

```

memset (buf, '\0', maxlen);
maxlen -= 1; /* ensure nul term */

retval = READ_STATUS_OK;

while (TRUE)
{
    ssize_t chunk;
    size_t to_read;

again:
    to_read = maxlen - bytes;

    if (to_read == 0)
        break;

    chunk = read (fd,
                  buf + bytes,
                  to_read);
    if (chunk < 0 && errno == EINTR)
        goto again;

    if (chunk < 0)
    {
        retval = READ_STATUS_ERROR;
        break;
    }
    else if (chunk == 0)
    {
        retval = READ_STATUS_EOF;
        break; /* EOF */
    }
    else /* chunk > 0 */
        bytes += chunk;
}

if (retval == READ_STATUS_EOF &&
    bytes > 0)
    retval = READ_STATUS_OK;

/* whack newline */
if (retval != READ_STATUS_ERROR &&
    bytes > 0 &&
    buf[bytes-1] == '\n')
    buf[bytes-1] = '\0';

return retval;
}

static ReadStatus
read_pid (int          fd,

```

```

        pid_t      *buf)
{
    size_t bytes = 0;
    ReadStatus retval;

    retval = READ_STATUS_OK;

    while (TRUE)
    {
        ssize_t chunk;
        size_t to_read;

    again:
        to_read = sizeof (pid_t) - bytes;

        if (to_read == 0)
            break;

        chunk = read (fd,
                     ((char*)buf) + bytes,
                     to_read);
        if (chunk < 0 && errno == EINTR)
            goto again;

        if (chunk < 0)
        {
            retval = READ_STATUS_ERROR;
            break;
        }
        else if (chunk == 0)
        {
            retval = READ_STATUS_EOF;
            break; /* EOF */
        }
        else /* chunk > 0 */
            bytes += chunk;
    }

    return retval;
}

static void
do_write (int fd, const void *buf, size_t count)
{
    size_t bytes_written;
    int ret;

    bytes_written = 0;

    again:

```

```

    ret = write (fd, ((const char*)buf) + bytes_written, count -
bytes_written);

    if (ret < 0)
    {
        if (errno == EINTR)
            goto again;
        else
        {
            fprintf (stderr, "Failed to write data to pipe! %s\n",
                strerror (errno));
            exit (1); /* give up, we suck */
        }
    }
    else
        bytes_written += ret;

    if (bytes_written < count)
        goto again;
}

static void
write_pid (int    fd,
           pid_t pid)
{
    do_write (fd, &pid, sizeof (pid));
}

static int
do_waitpid (pid_t pid)
{
    int ret;

again:
    ret = waitpid (pid, NULL, 0);

    if (ret < 0 &&
        errno == EINTR)
        goto again;

    return ret;
}

static pid_t bus_pid_to_kill = -1;

static void
kill_bus(void)
{
    verbose ("Killing message bus and exiting babysitter\n");
    kill (bus_pid_to_kill, SIGTERM);
    sleep (3);
    kill (bus_pid_to_kill, SIGKILL);
}

```

```

}

void
kill_bus_and_exit (int exitcode)
{
    /* in case these point to any NFS mounts, get rid of them
    immediately */
    close (0);
    close (1);
    close (2);

    kill_bus();

    exit (exitcode);
}

static void
print_variables (const char *bus_address, pid_t bus_pid, long bus_wid,
                int c_shell_syntax, int bourne_shell_syntax,
                int binary_syntax)
{
    if (binary_syntax)
    {
        do_write (1, bus_address, strlen (bus_address) + 1);
        do_write (1, &bus_pid, sizeof bus_pid);
        do_write (1, &bus_wid, sizeof bus_wid);
        return;
    }
    else if (c_shell_syntax)
    {
        printf ("setenv DBUS_SESSION_BUS_ADDRESS '%s';\n", bus_address);
        printf ("set DBUS_SESSION_BUS_PID=%ld;\n", (long) bus_pid);
        if (bus_wid)
            printf ("set DBUS_SESSION_BUS_WINDOWID=%ld;\n", (long)
bus_wid);
        fflush (stdout);
    }
    else if (bourne_shell_syntax)
    {
        printf ("DBUS_SESSION_BUS_ADDRESS='%s';\n", bus_address);
        printf ("export DBUS_SESSION_BUS_ADDRESS;\n");
        printf ("DBUS_SESSION_BUS_PID=%ld;\n", (long) bus_pid);
        if (bus_wid)
            printf ("DBUS_SESSION_BUS_WINDOWID=%ld;\n", (long) bus_wid);
        fflush (stdout);
    }
    else
    {
        printf ("DBUS_SESSION_BUS_ADDRESS=%s\n", bus_address);
        printf ("DBUS_SESSION_BUS_PID=%ld\n", (long) bus_pid);
        if (bus_wid)
            printf ("DBUS_SESSION_BUS_WINDOWID=%ld\n", (long) bus_wid);
    }
}

```



```

        fflush (stdout);
    }
}

static int got_sighup = FALSE;

static void
signal_handler (int sig)
{
    switch (sig)
    {
#ifdef SIGHUP
        case SIGHUP:
#endif
        case SIGINT:
        case SIGTERM:
            got_sighup = TRUE;
            break;
    }
}

static void
kill_bus_when_session_ends (void)
{
    int tty_fd;
    int x_fd;
    fd_set read_set;
    fd_set err_set;
    struct sigaction act;
    sigset_t empty_mask;

    /* install SIGHUP handler */
    got_sighup = FALSE;
    sigemptyset (&empty_mask);
    act.sa_handler = signal_handler;
    act.sa_mask     = empty_mask;
    act.sa_flags    = 0;
    sigaction (SIGHUP, &act, NULL);
    sigaction (SIGTERM, &act, NULL);
    sigaction (SIGINT, &act, NULL);

#ifdef DBUS_BUILD_X11
    x11_init();
    if (xdisplay != NULL)
    {
        x_fd = ConnectionNumber (xdisplay);
    }
    else
        x_fd = -1;
#else
    x_fd = -1;
#endif
}

```

```

if (isatty (0))
    tty_fd = 0;
else
    tty_fd = -1;

if (x_fd >= 0)
{
    verbose ("session lifetime is defined by X, not monitoring
stdin\n");
    tty_fd = -1;
}
else if (tty_fd >= 0)
{
    verbose ("stdin isatty(), monitoring it\n");
}
else
{
    verbose ("stdin was not a TTY, not monitoring it\n");
}

if (tty_fd < 0 && x_fd < 0)
{
    fprintf (stderr, "No terminal on standard input and no X
display; cannot attach message bus to session lifetime\n");
    exit (1);
}

while (TRUE)
{
#ifdef DBUS_BUILD_X11
    /* Dump events on the floor, and let
    * IO error handler run if we lose
    * the X connection. It's important to
    * run this before going into select() since
    * we might have queued outgoing messages or
    * events.
    */
    x11_handle_event ();
#endif

    FD_ZERO (&read_set);
    FD_ZERO (&err_set);

    if (tty_fd >= 0)
    {
        FD_SET (tty_fd, &read_set);
        FD_SET (tty_fd, &err_set);
    }

    if (x_fd >= 0)
    {

```

```

        FD_SET (x_fd, &read_set);
        FD_SET (x_fd, &err_set);
    }

    select (MAX (tty_fd, x_fd) + 1,
            &read_set, NULL, &err_set, NULL);

    if (got_sighup)
    {
        verbose ("Got SIGHUP, exiting\n");
        kill_bus_and_exit (0);
    }

#ifdef DBUS_BUILD_X11
    /* Events will be processed before we select again
    */
    if (x_fd >= 0)
        verbose ("X fd condition reading = %d error = %d\n",
                FD_ISSET (x_fd, &read_set),
                FD_ISSET (x_fd, &err_set));
#endif

    if (tty_fd >= 0)
    {
        if (FD_ISSET (tty_fd, &read_set))
        {
            int bytes_read;
            char discard[512];

            verbose ("TTY ready for reading\n");

            bytes_read = read (tty_fd, discard, sizeof (discard));

            verbose ("Read %d bytes from TTY errno = %d\n",
                    bytes_read, errno);

            if (bytes_read == 0)
                kill_bus_and_exit (0); /* EOF */
            else if (bytes_read < 0 && errno != EINTR)
            {
                /* This shouldn't happen I don't think; to avoid
                * spinning on the fd forever we exit.
                */
                fprintf (stderr, "dbus-launch: error reading from
stdin: %s\n",
                        strerror (errno));
                kill_bus_and_exit (0);
            }
        }
        else if (FD_ISSET (tty_fd, &err_set))
        {
            verbose ("TTY has error condition\n");

```

```

        kill_bus_and_exit (0);
    }
}

static void
babysit (int    exit_with_session,
         pid_t  child_pid,
         int    read_bus_pid_fd) /* read pid from here */
{
    int ret;
    int dev_null_fd;
    const char *s;

    verbose ("babysitting, exit_with_session = %d, child_pid = %ld,
read_bus_pid_fd = %d\n",
            exit_with_session, (long) child_pid, read_bus_pid_fd);

    /* We chdir ("/") since we are persistent and daemon-like, and fork
    * again so dbus-launch can reap the parent.  However, we don't
    * setsid() or close fd 0 because the idea is to remain attached
    * to the tty and the X server in order to kill the message bus
    * when the session ends.
    */

    if (chdir ("/") < 0)
    {
        fprintf (stderr, "Could not change to root directory: %s\n",
                strerror (errno));
        exit (1);
    }

    /* Close stdout/stderr so we don't block an "eval" or otherwise
    * lock up.  stdout is still chaining through to dbus-launch
    * and in turn to the parent shell.
    */
    dev_null_fd = open ("/dev/null", O_RDWR);
    if (dev_null_fd >= 0)
    {
        if (!exit_with_session)
            dup2 (dev_null_fd, 0);
        dup2 (dev_null_fd, 1);
        s = getenv ("DBUS_DEBUG_OUTPUT");
        if (s == NULL || *s == '\0')
            dup2 (dev_null_fd, 2);
    }
    else
    {
        fprintf (stderr, "Failed to open /dev/null: %s\n",
                strerror (errno));
    }
}

```

```

    /* continue, why not */
}

ret = fork ();

if (ret < 0)
{
    fprintf (stderr, "fork() failed in babysitter: %s\n",
            strerror (errno));
    exit (1);
}

if (ret > 0)
{
    /* Parent reaps pre-fork part of bus daemon, then exits and is
    * reaped so the babysitter isn't a zombie
    */

    verbose ("=== Babysitter's intermediate parent continues
again\n");

    if (do_waitpid (child_pid) < 0)
    {
        /* shouldn't happen */
        fprintf (stderr, "Failed waitpid() waiting for bus daemon's
parent\n");
        exit (1);
    }

    verbose ("Babysitter's intermediate parent exiting\n");

    exit (0);
}

/* Child continues */
verbose ("=== Babysitter process created\n");

verbose ("Reading PID from bus\n");

switch (read_pid (read_bus_pid_fd, &bus_pid_to_kill))
{
    case READ_STATUS_OK:
        break;
    case READ_STATUS_EOF:
        fprintf (stderr, "EOF in dbus-launch reading PID from bus
daemon\n");
        exit (1);
        break;
    case READ_STATUS_ERROR:
        fprintf (stderr, "Error in dbus-launch reading PID from bus
daemon: %s\n",
                strerror (errno));

```

```

        exit (1);
        break;
    }

    verbose ("Got PID %ld from daemon\n",
            (long) bus_pid_to_kill);

    if (exit_with_session)
    {
        /* Bus is now started and launcher has needed info;
         * we connect to X display and tty and wait to
         * kill bus if requested.
         */

        kill_bus_when_session_ends ();
    }

    verbose ("Babysitter exiting\n");

    exit (0);
}

static void
do_close_stderr (void)
{
    int fd;

    fflush (stderr);

    /* dbus-launch is a Unix-only program, so we can rely on /dev/null
    being there.
     * We're including unistd.h and we're dealing with sh/csh launch
    sequences...
     */
    fd = open ("/dev/null", O_RDWR);
    if (fd == -1)
    {
        fprintf (stderr, "Internal error: cannot open /dev/null: %s",
                strerror (errno));
        exit (1);
    }

    close (2);
    if (dup2 (fd, 2) == -1)
    {
        /* error; we can't report an error anymore... */
        exit (1);
    }
    close (fd);
}

static void

```

```

pass_info (const char *runprog, const char *bus_address, pid_t
bus_pid,
           long bus_wid, int c_shell_syntax, int bourne_shell_syntax,
           int binary_syntax,
           int argc, char **argv, int remaining_args)
{
    char *envvar = NULL;
    char **args = NULL;

    if (runprog)
    {
        int i;

        envvar = malloc (strlen ("DBUS_SESSION_BUS_ADDRESS=") +
            strlen (bus_address) + 1);
        args = malloc (sizeof (char *) * ((argc-remaining_args)+2));

        if (envvar == NULL || args == NULL)
            goto oom;

        args[0] = xstrdup (runprog);
        if (!args[0])
            goto oom;
        for (i = 1; i <= (argc-remaining_args); i++)
        {
            size_t len = strlen (argv[remaining_args+i-1])+1;
            args[i] = malloc (len);
            if (!args[i])
                goto oom;
            strncpy (args[i], argv[remaining_args+i-1], len);
        }
        args[i] = NULL;

        strcpy (envvar, "DBUS_SESSION_BUS_ADDRESS=");
        strcat (envvar, bus_address);
        putenv (envvar);

        execvp (runprog, args);
        fprintf (stderr, "Couldn't exec %s: %s\n", runprog, strerror
(errno));
        exit (1);
    }
    else
    {
        print_variables (bus_address, bus_pid, bus_wid, c_shell_syntax,
            bourne_shell_syntax, binary_syntax);
    }
    verbose ("dbus-launch exiting\n");

    fflush (stdout);
    fflush (stderr);
    close (1);
}

```

```

    close (2);
    exit (0);
oom:
    if (envvar)
        free (envvar);

    if (args)
        free (args);

    fprintf (stderr, "Out of memory!");
    exit (1);
}

#define READ_END 0
#define WRITE_END 1

int
main (int argc, char **argv)
{
    const char *prev_arg;
    const char *shname;
    const char *runprog = NULL;
    int remaining_args = 0;
    int exit_with_session;
    int binary_syntax = FALSE;
    int c_shell_syntax = FALSE;
    int bourne_shell_syntax = FALSE;
    int auto_shell_syntax = FALSE;
    int autolaunch = FALSE;
    int requires_arg = FALSE;
    int close_stderr = FALSE;
    int i;
    int ret;
    int bus_pid_to_launcher_pipe[2];
    int bus_pid_to_babysitter_pipe[2];
    int bus_address_to_launcher_pipe[2];
    char *config_file;

    exit_with_session = FALSE;
    config_file = NULL;

    prev_arg = NULL;
    i = 1;
    while (i < argc)
    {
        const char *arg = argv[i];

        if (strcmp (arg, "--help") == 0 ||
            strcmp (arg, "-h") == 0 ||
            strcmp (arg, "-?") == 0)
            usage (0);
        else if (strcmp (arg, "--auto-syntax") == 0)

```



```

    auto_shell_syntax = TRUE;
else if (strcmp (arg, "-c") == 0 ||
        strcmp (arg, "--csh-syntax") == 0)
    c_shell_syntax = TRUE;
else if (strcmp (arg, "-s") == 0 ||
        strcmp (arg, "--sh-syntax") == 0)
    bourne_shell_syntax = TRUE;
else if (strcmp (arg, "--binary-syntax") == 0)
    binary_syntax = TRUE;
else if (strcmp (arg, "--version") == 0)
    version ();
else if (strcmp (arg, "--exit-with-session") == 0)
    exit_with_session = TRUE;
else if (strcmp (arg, "--close-stderr") == 0)
    close_stderr = TRUE;
else if (strstr (arg, "--autolaunch=") == arg)
    {
        const char *s;

        if (autolaunch)
            {
                fprintf (stderr, "--autolaunch given twice\n");
                exit (1);
            }

        autolaunch = TRUE;

        s = strchr (arg, '=');
        ++s;

        save_machine_uuid (s);
    }
else if (prev_arg &&
        strcmp (prev_arg, "--autolaunch") == 0)
    {
        if (autolaunch)
            {
                fprintf (stderr, "--autolaunch given twice\n");
                exit (1);
            }

        autolaunch = TRUE;

        save_machine_uuid (arg);
        requires_arg = FALSE;
    }
else if (strcmp (arg, "--autolaunch") == 0)
    requires_arg = TRUE;
else if (strstr (arg, "--config-file=") == arg)
    {
        const char *file;

```

```

    if (config_file != NULL)
    {
        fprintf (stderr, "--config-file given twice\n");
        exit (1);
    }

    file = strchr (arg, '=');
    ++file;

    config_file = xstrdup (file);
}
else if (prev_arg &&
        strcmp (prev_arg, "--config-file") == 0)
{
    if (config_file != NULL)
    {
        fprintf (stderr, "--config-file given twice\n");
        exit (1);
    }

    config_file = xstrdup (arg);
    requires_arg = FALSE;
}
else if (strcmp (arg, "--config-file") == 0)
requires_arg = TRUE;
else if (arg[0] == '-')
{
    if (strcmp (arg, "--") != 0)
    {
        fprintf (stderr, "Option `%s' is unknown.\n", arg);
        exit (1);
    }
    else
    {
        runprog = argv[i+1];
        remaining_args = i+2;
        break;
    }
}
else
{
    runprog = arg;
    remaining_args = i+1;
    break;
}

prev_arg = arg;

++i;
}
if (requires_arg)
{

```

```

        fprintf (stderr, "Option `%s' requires an argument.\n",
prev_arg);
        exit (1);
    }

    if (auto_shell_syntax)
    {
        if ((shname = getenv ("SHELL")) != NULL)
        {
            if (!strncmp (shname + strlen (shname) - 3, "csh", 3))
                c_shell_syntax = TRUE;
            else
                bourne_shell_syntax = TRUE;
        }
        else
            bourne_shell_syntax = TRUE;
    }

    if (exit_with_session)
        verbose ("--exit-with-session enabled\n");

    if (autolaunch)
    {
#ifdef DBUS_BUILD_X11
        fprintf (stderr, "Autolaunch requested, but X11 support not
compiled in.\n"
                "Cannot continue.\n");
        exit (1);
#else /* DBUS_BUILD_X11 */
#ifdef DBUS_ENABLE_X11_AUTOLAUNCH
        fprintf (stderr, "X11 autolaunch support disabled at compile
time.\n");
        exit (1);
#else /* DBUS_ENABLE_X11_AUTOLAUNCH */
        char *address;
        pid_t pid;
        long wid;

        if (get_machine_uid () == NULL)
        {
            fprintf (stderr, "Machine UUID not provided as arg to --
autolaunch\n");
            exit (1);
        }

        verbose ("Autolaunch enabled (using X11).\n");
        if (!exit_with_session)
        {
            verbose ("--exit-with-session automatically enabled\n");
            exit_with_session = TRUE;
        }
#endif
#endif
#endif
    }

```

```

        if (!x11_init ())
        {
            fprintf (stderr, "Autolaunch error: X11 initialization
failed.\n");
            exit (1);
        }

        if (!x11_get_address (&address, &pid, &wid))
        {
            fprintf (stderr, "Autolaunch error: X11 communication
error.\n");
            exit (1);
        }

        if (address != NULL)
        {
            verbose ("dbus-daemon is already running. Returning existing
parameters.\n");
            pass_info (runprog, address, pid, wid, c_shell_syntax,
                bourne_shell_syntax, binary_syntax, argc, argv,
remaining_args);
            exit (0);
        }
#endif /* DBUS_ENABLE_X11_AUTOLAUNCH */
    }
    else if (read_machine_uuid_if_needed())
    {
        x11_init();
#endif /* DBUS_BUILD_X11 */
    }

if (pipe (bus_pid_to_launcher_pipe) < 0 ||
    pipe (bus_address_to_launcher_pipe) < 0 ||
    pipe (bus_pid_to_babysitter_pipe) < 0)
{
    fprintf (stderr,
            "Failed to create pipe: %s\n",
            strerror (errno));
    exit (1);
}

ret = fork ();
if (ret < 0)
{
    fprintf (stderr, "Failed to fork: %s\n",
            strerror (errno));
    exit (1);
}

if (ret == 0)
{

```

```

    /* Child */
#define MAX_FD_LEN 64
    char write_pid_fd_as_string[MAX_FD_LEN];
    char write_address_fd_as_string[MAX_FD_LEN];

#ifdef DBUS_BUILD_X11
    xdisplay = NULL;
#endif

    if (close_stderr)
do_close_stderr ();

    verbose ("=== Babysitter's intermediate parent created\n");

    /* Fork once more to create babysitter */

    ret = fork ();
    if (ret < 0)
    {
        fprintf (stderr, "Failed to fork: %s\n",
                strerror (errno));
        exit (1);
    }

    if (ret > 0)
    {
        /* In babysitter */
        verbose ("=== Babysitter's intermediate parent
continues\n");

        close (bus_pid_to_launcher_pipe[READ_END]);
        close (bus_pid_to_launcher_pipe[WRITE_END]);
        close (bus_address_to_launcher_pipe[READ_END]);
        close (bus_address_to_launcher_pipe[WRITE_END]);
        close (bus_pid_to_babysitter_pipe[WRITE_END]);

        /* babysit() will fork *again*
        * and will also reap the pre-forked bus
        * daemon
        */
        babysit (exit_with_session, ret,
                bus_pid_to_babysitter_pipe[READ_END]);
        exit (0);
    }

    verbose ("=== Bus exec process created\n");

    /* Now we are the bus process (well, almost;
    * dbus-daemon itself forks again)
    */
    close (bus_pid_to_launcher_pipe[READ_END]);
    close (bus_address_to_launcher_pipe[READ_END]);

```

```

close (bus_pid_to_babysitter_pipe[READ_END]);
close (bus_pid_to_babysitter_pipe[WRITE_END]);

sprintf (write_pid_fd_as_string,
         "%d", bus_pid_to_launcher_pipe[WRITE_END]);

sprintf (write_address_fd_as_string,
         "%d", bus_address_to_launcher_pipe[WRITE_END]);

verbose ("Calling exec()\n");

#ifdef DBUS_BUILD_TESTS
/* exec from testdir */
if (getenv("DBUS_USE_TEST_BINARY") != NULL)
{
    execl (TEST_BUS_BINARY,
          TEST_BUS_BINARY,
          "--fork",
          "--print-pid", write_pid_fd_as_string,
          "--print-address", write_address_fd_as_string,
          config_file ? "--config-file" : "--session",
          config_file, /* has to be last in this varargs list
*/
          NULL);

    fprintf (stderr,
            "Failed to execute test message bus daemon %s: %s.
Will try again with the system path.\n",
            TEST_BUS_BINARY, strerror (errno));
}
#endif /* DBUS_BUILD_TESTS */

execl (DBUS_DAEMONDIR"/dbus-daemon",
       DBUS_DAEMONDIR"/dbus-daemon",
       "--fork",
       "--print-pid", write_pid_fd_as_string,
       "--print-address", write_address_fd_as_string,
       config_file ? "--config-file" : "--session",
       config_file, /* has to be last in this varargs list */
       NULL);

fprintf (stderr,
        "Failed to execute message bus daemon %s: %s. Will try
again without full path.\n",
        DBUS_DAEMONDIR"/dbus-daemon", strerror (errno));

/*
 * If it failed, try running without full PATH. Note this is
needed
 * because the build process builds the run-with-tmp-session-
bus.conf

```

```

    * file and the dbus-daemon will not be in the install location
during
    * build time.
    */
    execlp ("dbus-daemon",
           "dbus-daemon",
           "--fork",
           "--print-pid", write_pid_fd_as_string,
           "--print-address", write_address_fd_as_string,
           config_file ? "--config-file" : "--session",
           config_file, /* has to be last in this varargs list */
           NULL);

    fprintf (stderr,
            "Failed to execute message bus daemon: %s\n",
            strerror (errno));
    exit (1);
}
else
{
    /* Parent */
#define MAX_PID_LEN 64
    pid_t bus_pid;
    char bus_address[MAX_ADDR_LEN];
    char buf[MAX_PID_LEN];
    char *end;
    long wid = 0;
    long val;
    int ret2;

    verbose ("=== Parent dbus-launch continues\n");

    close (bus_pid_to_launcher_pipe[WRITE_END]);
    close (bus_address_to_launcher_pipe[WRITE_END]);
    close (bus_pid_to_babysitter_pipe[READ_END]);

    verbose ("Waiting for babysitter's intermediate parent\n");

    /* Immediately reap parent of babysitter
     * (which was created just for us to reap)
     */
    if (do_waitpid (ret) < 0)
    {
        fprintf (stderr, "Failed to waitpid() for babysitter
intermediate process: %s\n",
                strerror (errno));
        exit (1);
    }

    verbose ("Reading address from bus\n");

    /* Read the pipe data, print, and exit */

```

```

switch (read_line (bus_address_to_launcher_pipe[READ_END],
                  bus_address, MAX_ADDR_LEN))
{
case READ_STATUS_OK:
    break;
case READ_STATUS_EOF:
    fprintf (stderr, "EOF in dbus-launch reading address from
bus daemon\n");
    exit (1);
    break;
case READ_STATUS_ERROR:
    fprintf (stderr, "Error in dbus-launch reading address from
bus daemon: %s\n",
            strerror (errno));
    exit (1);
    break;
}

close (bus_address_to_launcher_pipe[READ_END]);

verbose ("Reading PID from daemon\n");
/* Now read data */
switch (read_line (bus_pid_to_launcher_pipe[READ_END], buf,
MAX_PID_LEN))
{
case READ_STATUS_OK:
    break;
case READ_STATUS_EOF:
    fprintf (stderr, "EOF reading PID from bus daemon\n");
    exit (1);
    break;
case READ_STATUS_ERROR:
    fprintf (stderr, "Error reading PID from bus daemon: %s\n",
            strerror (errno));
    exit (1);
    break;
}

end = NULL;
val = strtol (buf, &end, 0);
if (buf == end || end == NULL)
{
    fprintf (stderr, "Failed to parse bus PID \"%s\": %s\n",
            buf, strerror (errno));
    exit (1);
}

bus_pid = val;

close (bus_pid_to_launcher_pipe[READ_END]);

#ifdef DBUS_ENABLE_X11_AUTOLAUNCH

```



```

if (xdisplay != NULL)
{
    verbose("Saving x11 address\n");
    ret2 = x11_save_address (bus_address, bus_pid, &wid);
    /* Only get an existing dbus session when autolaunching */
    if (autolaunch)
    {
        if (ret2 == 0)
        {
            char *address = NULL;
            /* another window got added. Return its address */
            bus_pid_to_kill = bus_pid;
            if (x11_get_address (&address, &bus_pid, &wid)
                && address != NULL)
            {
                verbose ("dbus-daemon is already running.
Returning existing parameters.\n");
                /* Kill the old bus */
                kill_bus();
                pass_info (runprog, address, bus_pid, wid,
                    c_shell_syntax, bourne_shell_syntax,
binary_syntax,
                    argc, argv, remaining_args);
            }
        }
        if (ret2 < 0)
        {
            fprintf (stderr, "Error saving bus information.\n");
            bus_pid_to_kill = bus_pid;
            kill_bus_and_exit (1);
        }
    }
}
#endif

/* Forward the pid to the babysitter */
write_pid (bus_pid_to_babysitter_pipe[WRITE_END], bus_pid);
close (bus_pid_to_babysitter_pipe[WRITE_END]);

    pass_info (runprog, bus_address, bus_pid, wid, c_shell_syntax,
        bourne_shell_syntax, binary_syntax, argc, argv,
remaining_args);
}

return 0;
}

```

File = dbus-launch.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */

```
/* dbus-launch.h  dbus-launch utility
 *
 * Copyright (C) 2006 Thiago Macieira <thiago@kde.org>
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */
```

```
#ifndef DBUS_LAUNCH_H
#define DBUS_LAUNCH_H
```

```
#include <sys/types.h>
```

```
#ifndef TRUE
#define TRUE (1)
#endif
```

```
#ifndef FALSE
#define FALSE (0)
#endif
```

```
#undef MAX
#define MAX(a, b)  (((a) > (b)) ? (a) : (b))
```

```
#define MAX_ADDR_LEN 512
```

```
/* defined in dbus-launch.c */
void verbose (const char *format, ...);
char *xstrdup (const char *str);
void kill_bus_and_exit (int exitcode);
```

```
const char* get_machine_uuid (void);
```

```
#ifdef DBUS_BUILD_X11
/* defined in dbus-launch-x11.c */
```

```

int xll_init (void);
int xll_get_address (char **paddress, pid_t *pid, long *wid);
int xll_save_address (char *address, pid_t pid, long *wid);
void xll_handle_event (void);
#endif

#endif

```

File = dbus-launch.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://www.oasis-
open.org/docbook/xml/4.1.2/docbookx.dtd">
<!-- lifted from troff+man by doclifter -->
<refentry id='dbuslaunch1'>
<!-- -->
<!-- dbus\ -launch manual page. -->
<!-- Copyright (C) 2003 Red Hat, Inc. -->

<refmeta>
<refentrytitle>dbus-launch</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>
<refnamediv id='name'>
<refname>dbus-launch</refname>
<refpurpose>Utility to start a message bus from a shell
script</refpurpose>
</refnamediv>
<!-- body begins here -->
<refsynopsisdiv id='synopsis'>
<cmdsynopsis>
    <command>dbus-launch</command>    <arg choice='opt'>--version </arg>
    <arg choice='opt'>--sh-syntax </arg>
    <arg choice='opt'>--csh-syntax </arg>
    <arg choice='opt'>--auto-syntax </arg>
    <arg choice='opt'>--exit-with-session </arg>
    <arg choice='opt'>--
autolaunch=<replaceable>MACHINEID</replaceable></arg>
    <arg choice='opt'>--config-
file=<replaceable>FILENAME</replaceable></arg>
    <arg choice='opt'><replaceable>PROGRAM</replaceable></arg>
    <arg choice='opt'
rep='repeat'><replaceable>ARGS</replaceable></arg>
    <sbr/>
</cmdsynopsis>
</refsynopsisdiv>

<refsect1 id='description'><title>DESCRIPTION</title>

```

<para>The <command>dbus-launch</command> command is used to start a session bus instance of <emphasis remap='I'>dbus-daemon</emphasis> from a shell script.

It would normally be called from a user's login scripts. Unlike the daemon itself, <command>dbus-launch</command> exits, so backticks or the \$() construct can be used to read information from <command>dbus-launch</command>.</para>

<para>With no arguments, <command>dbus-launch</command> will launch a session bus instance and print the address and pid of that instance to standard output.</para>

<para>You may specify a program to be run; in this case, <command>dbus-launch</command> will launch a session bus instance, set the appropriate environment variables so the specified program can find the bus, and then execute the specified program, with the specified arguments. See below for examples.</para>

<para>If you launch a program, <command>dbus-launch</command> will not print the information about the new bus to standard output.</para>

<para>When <command>dbus-launch</command> prints bus information to standard output, by default it is in a simple key-value pairs format. However, you may request several alternate syntaxes using the --sh-syntax, --csh-syntax, --binary-syntax, or --auto-syntax options. Several of these cause <command>dbus-launch</command> to emit shell code to set up the environment.</para>

<para>With the --auto-syntax option, <command>dbus-launch</command> looks at the value of the SHELL environment variable to determine which shell syntax should be used. If SHELL ends in "csh", then csh-compatible code is emitted; otherwise Bourne shell code is emitted. Instead of passing --auto-syntax, you may explicitly specify a particular one by using --sh-syntax for Bourne syntax, or --csh-syntax for csh syntax. In scripts, it's more robust to avoid --auto-syntax and you hopefully know which shell your script is written in.</para>

<para>See <ulink url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop.org/software/dbus/</ulink> for more information

about D-Bus. See also the man page for `dbus-daemon`.

Here is an example of how to use `dbus-launch` with an sh-compatible shell to start the per-session bus daemon:

```
## test for an existing bus daemon, just to be safe
if test -z "$DBUS_SESSION_BUS_ADDRESS" ; then
    ## if not found, launch a new one
    eval `dbus-launch --sh-syntax --exit-with-session`
    echo "D-Bus per-session daemon address is:
$DBUS_SESSION_BUS_ADDRESS"
fi
```

`<!-- .fi -->`

You might run something like that in your login scripts.

Another way to use `dbus-launch` is to run your main session program, like so:

```
dbus-launch gnome-session
```

`<!-- .fi -->`

The above would likely be appropriate for `~/.xsession` or `~/.Xclients`.

`</refsect1>`

`<refsect1 id='automatic_launching'><title>AUTOMATIC LAUNCHING</title>`

`<para>If DBUS_SESSION_BUS_ADDRESS is not set for a process that tries to use`

`D-Bus, by default the process will attempt to invoke dbus-launch with the --autolaunch option to start up a new session bus or find the existing bus address on the X display or in a file in ~/.dbus/session-bus/`

Whenever an autolaunch occurs, the application that had to start a new bus will be in its own little world; it can effectively end up starting a whole new session if it tries to use a lot of bus services. This can be suboptimal or even totally broken, depending on the app and what it tries to do.

There are two common reasons for autolaunch. One is ssh to a remote

machine. The ideal fix for that would be forwarding of DBUS_SESSION_BUS_ADDRESS in the same way that DISPLAY is forwarded. In the meantime, you can edit the session.conf config file to have your session bus listen on TCP, and manually set DBUS_SESSION_BUS_ADDRESS, if you like.</para>

<para>The second common reason for autolaunch is an su to another user, and display of X applications running as the second user on the display belonging to the first user. Perhaps the ideal fix in this case would be to allow the second user to connect to the session bus of the first user, just as they can connect to the first user's display. However, a mechanism for that has not been coded.</para>

<para>You can always avoid autolaunch by manually setting DBUS_SESSION_BUS_ADDRESS. Autolaunch happens because the default address if none is set is "autolaunch:", so if any other address is set there will be no autolaunch. You can however include autolaunch in an explicit session bus address as a fallback, for example DBUS_SESSION_BUS_ADDRESS="something:,autolaunch:" - in that case if the first address doesn't work, processes will autolaunch. (The bus address variable contains a comma-separated list of addresses to try.)</para>

<para>The --autolaunch option is considered an internal implementation detail of libdbus, and in fact there are plans to change it. There's no real reason to use it outside of the libdbus implementation anyhow.</para>

</refsect1>

<refsect1 id='options'><title>OPTIONS</title>

<para>The following options are supported:</para>

<variablelist remap='TP'>

<varlistentry>

<term><option>--auto-syntax</option></term>

<listitem>

<para>Choose --csh-syntax or --sh-syntax based on the SHELL environment variable.</para>

<para><option>--binary-syntax</option>

Write to stdout a nul-terminated bus address, then the bus PID as a binary integer of size sizeof(pid_t), then the bus X window ID as a binary integer of size sizeof(long). Integers are in the machine's byte order, not network byte order or any other canonical byte order.</para>

</listitem>

</varlistentry>

```
<varlistentry>
<term><option>--close-stderr</option></term>
<listitem>
```

<para>Close the standard error output stream before starting the D-Bus daemon. This is useful if you want to capture dbus-launch error messages but you don't want dbus-daemon to keep the stream open to your application.</para>

```
</listitem>
</varlistentry>
<varlistentry>
<term><option>--config-file=FILENAME</option></term>
<listitem>
```

<para>Pass --config-file=FILENAME to the bus daemon, instead of passing it the --session argument. See the man page for dbus-daemon</para>

```
</listitem>
</varlistentry>
<varlistentry>
<term><option>--csh-syntax</option></term>
<listitem>
```

<para>Emit csh compatible code to set up environment variables.</para>

```
</listitem>
</varlistentry>
<varlistentry>
<term><option>--exit-with-session</option></term>
<listitem>
```

<para>If this option is provided, a persistent "babysitter" process will be created that watches stdin for HUP and tries to connect to the X server. If this process gets a HUP on stdin or loses its X connection, it kills the message bus daemon.</para>

```
</listitem>
</varlistentry>
<varlistentry>
<term><option>--autolaunch=MACHINEID</option></term>
<listitem>
```

<para>This option implies that <command>dbus-launch</command> should scan for a previously-started session and reuse the values found there. If no session is found, it will start a new session. The --exit-with-session option is implied if --autolaunch is given. This option is for the exclusive use of libdbus, you do not want to use it manually. It may change in the future.</para>

```
</listitem>
</varlistentry>
<varlistentry>
<term><option>--sh-syntax</option></term>
```

```

    <listitem>
<para>Emit Bourne-shell compatible code to set up environment
variables.</para>

    </listitem>
</varlistentry>
<varlistentry>
    <term><option>--version</option></term>
    <listitem>
<para>Print the version of dbus-launch</para>

    </listitem>
</varlistentry>
</variablelist>
</refsect1>

<refsect1 id='author'><title>AUTHOR</title>
<para>See <ulink
url='http://www.freedesktop.org/software/dbus/doc/AUTHORS'>http://www.
freedesktop.org/software/dbus/doc/AUTHORS</ulink></para>

</refsect1>

<refsect1 id='bugs'><title>BUGS</title>
<para>Please send bug reports to the D-Bus mailing list or bug
tracker,
see <ulink
url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop
.org/software/dbus/</ulink></para>
</refsect1>
</refentry>

```

File = dbus-list.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-list.c Generic linked list utility (internal to D-Bus
implementation)
*
* Copyright (C) 2002 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
*/

```



```
* This program is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program; if not, write to the Free Software  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA  
*  
*/
```

```
#include <config.h>  
#include "dbus-internals.h"  
#include "dbus-list.h"  
#include "dbus-mempool.h"  
#include "dbus-threads-internal.h"
```

```
/**  
 * @defgroup DBusList Linked list  
 * @ingroup DBusInternals  
 * @brief DBusList data structure  
 *  
 * Types and functions related to DBusList.  
 */
```

```
static DBusMemPool *list_pool;  
_DBUS_DEFINE_GLOBAL_LOCK (list);
```

```
/**  
 * @defgroup DBusListInternals Linked list implementation details  
 * @ingroup DBusInternals  
 * @brief DBusList implementation details  
 *  
 * The guts of DBusList.  
 *  
 * @{  
 */
```

```
/* the mem pool is probably a speed hit, with the thread  
 * lock, though it does still save memory - unknown.  
 */
```

```
static DBusList*  
alloc_link (void *data)  
{  
    DBusList *link;  
  
    _DBUS_LOCK (list);  
  
    if (list_pool == NULL)  
    {  
        list_pool = _dbus_mem_pool_new (sizeof (DBusList), TRUE);
```

```

    if (list_pool == NULL)
    {
        _DBUS_UNLOCK (list);
        return NULL;
    }

    link = _dbus_mem_pool_alloc (list_pool);
    if (link == NULL)
    {
        _dbus_mem_pool_free (list_pool);
        list_pool = NULL;
        _DBUS_UNLOCK (list);
        return NULL;
    }
}
else
{
    link = _dbus_mem_pool_alloc (list_pool);
}

if (link)
    link->data = data;

_DBUS_UNLOCK (list);

return link;
}

static void
free_link (DBusList *link)
{
    _DBUS_LOCK (list);
    if (_dbus_mem_pool_dealloc (list_pool, link))
    {
        _dbus_mem_pool_free (list_pool);
        list_pool = NULL;
    }

    _DBUS_UNLOCK (list);
}

static void
link_before (DBusList **list,
             DBusList *before_this_link,
             DBusList *link)
{
    if (*list == NULL)
    {
        link->prev = link;
        link->next = link;
        *list = link;
    }
}

```

```

    }
else
    {
        link->next = before_this_link;
        link->prev = before_this_link->prev;
        before_this_link->prev = link;
        link->prev->next = link;

        if (before_this_link == *list)
            *list = link;
    }
}

static void
link_after (DBusList **list,
            DBusList *after_this_link,
            DBusList *link)
{
    if (*list == NULL)
    {
        link->prev = link;
        link->next = link;
        *list = link;
    }
else
    {
        link->prev = after_this_link;
        link->next = after_this_link->next;
        after_this_link->next = link;
        link->next->prev = link;
    }
}

#ifdef DBUS_ENABLE_STATS
void
_dbus_list_get_stats (dbus_uint32_t *in_use_p,
                     dbus_uint32_t *in_free_list_p,
                     dbus_uint32_t *allocated_p)
{
    _DBUS_LOCK (list);
    _dbus_mem_pool_get_stats (list_pool, in_use_p, in_free_list_p,
allocated_p);
    _DBUS_UNLOCK (list);
}
#endif

/** @} */

/**
 * @addtogroup DBusList
 * @{
 */

```

```

/**
 * @struct DBusList
 *
 * A node in a linked list.
 *
 * DBusList is a circular list; that is, the tail of the list
 * points back to the head of the list. The empty list is
 * represented by a #NULL pointer.
 */

```

```

/**
 * @def _dbus_list_get_next_link
 *
 * Gets the next link in the list, or #NULL if
 * there are no more links. Used for iteration.
 *
 * @code
 * DBusList *link;
 * link = _dbus_list_get_first_link (&list);
 * while (link != NULL)
 * {
 *     printf ("value is %p\n", link->data);
 *     link = _dbus_list_get_next_link (&link);
 * }
 * @endcode
 *
 * @param list address of the list head.
 * @param link current link.
 * @returns the next link, or %NULL if none.
 */

```

```

/**
 * @def _dbus_list_get_prev_link
 *
 * Gets the previous link in the list, or #NULL if
 * there are no more links. Used for iteration.
 *
 * @code
 * DBusList *link;
 * link = _dbus_list_get_last_link (&list);
 * while (link != NULL)
 * {
 *     printf ("value is %p\n", link->data);
 *     link = _dbus_list_get_prev_link (&link);
 * }
 * @endcode
 *
 * @param list address of the list head.
 * @param link current link.
 * @returns the previous link, or %NULL if none.
 */

```

```

*
*/

/**
 * Allocates a linked list node. Useful for preallocating
 * nodes and using _dbus_list_append_link() to avoid
 * allocations.
 *
 * @param data the value to store in the link.
 * @returns a newly allocated link.
 */
DBusList*
_dbus_list_alloc_link (void *data)
{
    return alloc_link (data);
}

/**
 * Frees a linked list node allocated with _dbus_list_alloc_link.
 * Does not free the data in the node.
 *
 * @param link the list node
 */
void
_dbus_list_free_link (DBusList *link)
{
    free_link (link);
}

/**
 * Appends a value to the list. May return #FALSE
 * if insufficient memory exists to add a list link.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @param data the value to append.
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_list_append (DBusList **list,
                  void *data)
{
    if (!_dbus_list_prepend (list, data))
        return FALSE;

    /* Now cycle the list forward one so the prepended node is the tail
    */
    *list = (*list)->next;

    return TRUE;
}

```

```

/**
 * Prepends a value to the list. May return #FALSE
 * if insufficient memory exists to add a list link.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @param data the value to prepend.
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_list_prepend (DBusList **list,
                   void      *data)
{
    DBusList *link;

    link = alloc_link (data);
    if (link == NULL)
        return FALSE;

    link_before (list, *list, link);

    return TRUE;
}

/**
 * Appends a link to the list.
 * Cannot fail due to out of memory.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @param link the link to append.
 */
void
_dbus_list_append_link (DBusList **list,
                       DBusList *link)
{
    _dbus_list_prepend_link (list, link);

    /* Now cycle the list forward one so the prepended node is the tail
 */
    *list = (*list)->next;
}

/**
 * Prepends a link to the list.
 * Cannot fail due to out of memory.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @param link the link to prepend.
 */

```

```

void
_dbus_list_prepend_link (DBusList **list,
                        DBusList *link)
{
    link_before (list, *list, link);
}

/**
 * Inserts data into the list after the given existing link.
 *
 * @param list the list to modify
 * @param after_this_link existing link to insert after, or #NULL to
prepend
 * @param data the value to insert
 * @returns #TRUE on success, #FALSE if memory allocation fails
 */
dbus_bool_t
_dbus_list_insert_after (DBusList **list,
                        DBusList *after_this_link,
                        void *data)
{
    DBusList *link;

    if (after_this_link == NULL)
        return _dbus_list_prepend (list, data);
    else
    {
        link = alloc_link (data);
        if (link == NULL)
            return FALSE;

        link_after (list, after_this_link, link);
    }

    return TRUE;
}

/**
 * Inserts a link into the list before the given existing link.
 *
 * @param list the list to modify
 * @param before_this_link existing link to insert before, or #NULL to
append
 * @param link the link to insert
 */
void
_dbus_list_insert_before_link (DBusList **list,
                              DBusList *before_this_link,
                              DBusList *link)
{
    if (before_this_link == NULL)
        _dbus_list_append_link (list, link);
}

```

```

    else
        link_before (list, before_this_link, link);
}

/**
 * Inserts a link into the list after the given existing link.
 *
 * @param list the list to modify
 * @param after_this_link existing link to insert after, or #NULL to
prepend
 * @param link the link to insert
 */
void
_dbus_list_insert_after_link (DBusList **list,
                             DBusList *after_this_link,
                             DBusList *link)
{
    if (after_this_link == NULL)
        _dbus_list_prepend_link (list, link);
    else
        link_after (list, after_this_link, link);
}

/**
 * Removes a value from the list. Only removes the
 * first value equal to the given data pointer,
 * even if multiple values exist which match.
 * This is a linear-time operation.
 *
 * @param list address of the list head.
 * @param data the value to remove.
 * @returns #TRUE if a value was found to remove.
 */
dbus_bool_t
_dbus_list_remove (DBusList **list,
                  void *data)
{
    DBusList *link;

    link = *list;
    while (link != NULL)
    {
        if (link->data == data)
        {
            _dbus_list_remove_link (list, link);
            return TRUE;
        }

        link = _dbus_list_get_next_link (list, link);
    }

    return FALSE;
}

```



```

}

/**
 * Removes a value from the list. Only removes the
 * last value equal to the given data pointer,
 * even if multiple values exist which match.
 * This is a linear-time operation.
 *
 * @param list address of the list head.
 * @param data the value to remove.
 * @returns #TRUE if a value was found to remove.
 */
dbus_bool_t
_dbus_list_remove_last (DBusList **list,
                       void      *data)
{
    DBusList *link;

    link = _dbus_list_find_last (list, data);
    if (link)
        {
            _dbus_list_remove_link (list, link);
            return TRUE;
        }
    else
        return FALSE;
}

/**
 * Finds a value in the list. Returns the last link
 * with value equal to the given data pointer.
 * This is a linear-time operation.
 * Returns #NULL if no value found that matches.
 *
 * @param list address of the list head.
 * @param data the value to find.
 * @returns the link if found
 */
DBusList*
_dbus_list_find_last (DBusList **list,
                     void      *data)
{
    DBusList *link;

    link = _dbus_list_get_last_link (list);

    while (link != NULL)
        {
            if (link->data == data)
                return link;

            link = _dbus_list_get_prev_link (list, link);
        }
}

```

```

    }

    return NULL;
}

/**
 * Removes the given link from the list, but doesn't
 * free it. _dbus_list_remove_link() both removes the
 * link and also frees it.
 *
 * @param list the list
 * @param link the link in the list
 */
void
_dbus_list_unlink (DBusList **list,
                  DBusList *link)
{
    if (link->next == link)
    {
        /* one-element list */
        *list = NULL;
    }
    else
    {
        link->prev->next = link->next;
        link->next->prev = link->prev;

        if (*list == link)
            *list = link->next;
    }

    link->next = NULL;
    link->prev = NULL;
}

/**
 * Removes a link from the list. This is a constant-time operation.
 *
 * @param list address of the list head.
 * @param link the list link to remove.
 */
void
_dbus_list_remove_link (DBusList **list,
                       DBusList *link)
{
    _dbus_list_unlink (list, link);
    free_link (link);
}

/**
 * Frees all links in the list and sets the list head to #NULL. Does
 * not free the data in each link, for obvious reasons. This is a

```

```

* linear-time operation.
*
* @param list address of the list head.
*/
void
_dbus_list_clear (DBusList **list)
{
    DBusList *link;

    link = *list;
    while (link != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (list, link);

        free_link (link);

        link = next;
    }

    *list = NULL;
}

/**
 * Gets the first link in the list.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @returns the first link, or #NULL for an empty list.
 */
DBusList*
_dbus_list_get_first_link (DBusList **list)
{
    return *list;
}

/**
 * Gets the last link in the list.
 * This is a constant-time operation.
 *
 * @param list address of the list head.
 * @returns the last link, or #NULL for an empty list.
 */
DBusList*
_dbus_list_get_last_link (DBusList **list)
{
    if (*list == NULL)
        return NULL;
    else
        return (*list)->prev;
}

/**

```

```

* Gets the last data in the list.
* This is a constant-time operation.
*
* @param list address of the list head.
* @returns the last data in the list, or #NULL for an empty list.
*/
void*
_dbus_list_get_last (DBusList **list)
{
    if (*list == NULL)
        return NULL;
    else
        return (*list)->prev->data;
}

/**
* Gets the first data in the list.
* This is a constant-time operation.
*
* @param list address of the list head.
* @returns the first data in the list, or #NULL for an empty list.
*/
void*
_dbus_list_get_first (DBusList **list)
{
    if (*list == NULL)
        return NULL;
    else
        return (*list)->data;
}

/**
* Removes the first link in the list and returns it. This is a
* constant-time operation.
*
* @param list address of the list head.
* @returns the first link in the list, or #NULL for an empty list.
*/
DBusList*
_dbus_list_pop_first_link (DBusList **list)
{
    DBusList *link;

    link = _dbus_list_get_first_link (list);
    if (link == NULL)
        return NULL;

    _dbus_list_unlink (list, link);

    return link;
}

```

```

/**
 * Removes the first value in the list and returns it. This is a
 * constant-time operation.
 *
 * @param list address of the list head.
 * @returns the first data in the list, or #NULL for an empty list.
 */
void*
_dbus_list_pop_first (DBusList **list)
{
    DBusList *link;
    void *data;

    link = _dbus_list_get_first_link (list);
    if (link == NULL)
        return NULL;

    data = link->data;
    _dbus_list_remove_link (list, link);

    return data;
}

/**
 * Removes the last value in the list and returns it. This is a
 * constant-time operation.
 *
 * @param list address of the list head.
 * @returns the last data in the list, or #NULL for an empty list.
 */
void*
_dbus_list_pop_last (DBusList **list)
{
    DBusList *link;
    void *data;

    link = _dbus_list_get_last_link (list);
    if (link == NULL)
        return NULL;

    data = link->data;
    _dbus_list_remove_link (list, link);

    return data;
}

/**
 * Copies a list. This is a linear-time operation. If there isn't
 * enough memory to copy the entire list, the destination list will be
 * set to #NULL.
 *
 * @param list address of the head of the list to copy.

```

```

* @param dest address where the copied list should be placed.
* @returns #TRUE on success, #FALSE if not enough memory.
*/
dbus_bool_t
_dbus_list_copy (DBusList **list,
                 DBusList **dest)
{
    DBusList *link;

    _dbus_assert (list != dest);

    *dest = NULL;

    link = *list;
    while (link != NULL)
    {
        if (!_dbus_list_append (dest, link->data))
        {
            /* free what we have so far */
            _dbus_list_clear (dest);
            return FALSE;
        }

        link = _dbus_list_get_next_link (list, link);
    }

    return TRUE;
}

/**
 * Gets the length of a list. This is a linear-time
 * operation.
 *
 * @param list address of the head of the list
 * @returns number of elements in the list.
 */
int
_dbus_list_get_length (DBusList **list)
{
    DBusList *link;
    int length;

    length = 0;

    link = *list;
    while (link != NULL)
    {
        ++length;

        link = _dbus_list_get_next_link (list, link);
    }
}

```

```

    return length;
}

/**
 * Calls the given function for each element in the list. The
 * function is passed the list element as its first argument, and the
 * given data as its second argument.
 *
 * @param list address of the head of the list.
 * @param function function to call for each element.
 * @param data extra data for the function.
 */
void
_dbus_list_foreach (DBusList **list,
                   DBusForeachFunction function,
                   void *data)
{
    DBusList *link;

    link = *list;
    while (link != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (list, link);

        (* function) (link->data, data);

        link = next;
    }
}

/**
 * Check whether length is exactly one.
 *
 * @param list the list
 * @returns #TRUE if length is exactly one
 */
dbus_bool_t
_dbus_list_length_is_one (DBusList **list)
{
    return (*list != NULL &&
           (*list)->next == *list);
}

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

static void
verify_list (DBusList **list)

```

```

{
    DBusList *link;
    int length;

    link = *list;

    if (link == NULL)
        return;

    if (link->next == link)
    {
        _dbus_assert (link->prev == link);
        _dbus_assert (*list == link);
        return;
    }

    length = 0;
    do
    {
        length += 1;
        _dbus_assert (link->prev->next == link);
        _dbus_assert (link->next->prev == link);
        link = link->next;
    }
    while (link != *list);

    _dbus_assert (length == _dbus_list_get_length (list));

    if (length == 1)
        _dbus_assert (_dbus_list_length_is_one (list));
    else
        _dbus_assert (!_dbus_list_length_is_one (list));
}

static dbus_bool_t
is_ascending_sequence (DBusList **list)
{
    DBusList *link;
    int prev;

    prev = _DBUS_INT_MIN;

    link = _dbus_list_get_first_link (list);
    while (link != NULL)
    {
        int v = _DBUS_POINTER_TO_INT (link->data);

        if (v <= prev)
            return FALSE;

        prev = v;
    }
}

```



```

        link = _dbus_list_get_next_link (list, link);
    }

    return TRUE;
}

static dbus_bool_t
is_descending_sequence (DBusList **list)
{
    DBusList *link;
    int prev;

    prev = _DBUS_INT_MAX;

    link = _dbus_list_get_first_link (list);
    while (link != NULL)
    {
        int v = _DBUS_POINTER_TO_INT (link->data);

        if (v >= prev)
            return FALSE;

        prev = v;

        link = _dbus_list_get_next_link (list, link);
    }

    return TRUE;
}

static dbus_bool_t
all_even_values (DBusList **list)
{
    DBusList *link;

    link = _dbus_list_get_first_link (list);
    while (link != NULL)
    {
        int v = _DBUS_POINTER_TO_INT (link->data);

        if ((v % 2) != 0)
            return FALSE;

        link = _dbus_list_get_next_link (list, link);
    }

    return TRUE;
}

static dbus_bool_t
all_odd_values (DBusList **list)
{

```

```

DBusList *link;

link = _dbus_list_get_first_link (list);
while (link != NULL)
{
    int v = _DBUS_POINTER_TO_INT (link->data);

    if ((v % 2) == 0)
        return FALSE;

    link = _dbus_list_get_next_link (list, link);
}

return TRUE;
}

static dbus_bool_t
lists_equal (DBusList **list1,
            DBusList **list2)
{
    DBusList *link1;
    DBusList *link2;

    link1 = _dbus_list_get_first_link (list1);
    link2 = _dbus_list_get_first_link (list2);
    while (link1 && link2)
    {
        if (link1->data != link2->data)
            return FALSE;

        link1 = _dbus_list_get_next_link (list1, link1);
        link2 = _dbus_list_get_next_link (list2, link2);
    }

    if (link1 || link2)
        return FALSE;

    return TRUE;
}

/**
 * @ingroup DBusListInternals
 * Unit test for DBusList
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_list_test (void)
{
    DBusList *list1;
    DBusList *list2;
    DBusList *link1;
    DBusList *link2;

```

```

DBusList *copy1;
DBusList *copy2;
int i;

list1 = NULL;
list2 = NULL;

/* Test append and prepend */

i = 0;
while (i < 10)
{
    if (!_dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i)))
        _dbus_assert_not_reached ("could not allocate for append");

    if (!_dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i)))
        _dbus_assert_not_reached ("count not allocate for prepend");
    ++i;

    verify_list (&list1);
    verify_list (&list2);

    _dbus_assert (_dbus_list_get_length (&list1) == i);
    _dbus_assert (_dbus_list_get_length (&list2) == i);
}

_dbus_assert (is_ascending_sequence (&list1));
_dbus_assert (is_descending_sequence (&list2));

/* Test list clear */
_dbus_list_clear (&list1);
_dbus_list_clear (&list2);

verify_list (&list1);
verify_list (&list2);

/* Test get_first, get_last, pop_first, pop_last */

i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;
}

--i;
while (i >= 0)
{
    void *got_data1;
    void *got_data2;

```

```

void *data1;
void *data2;

got_data1 = _dbus_list_get_last (&list1);
got_data2 = _dbus_list_get_first (&list2);

data1 = _dbus_list_pop_last (&list1);
data2 = _dbus_list_pop_first (&list2);

_dbus_assert (got_data1 == data1);
_dbus_assert (got_data2 == data2);

_dbus_assert (_DBUS_POINTER_TO_INT (data1) == i);
_dbus_assert (_DBUS_POINTER_TO_INT (data2) == i);

verify_list (&list1);
verify_list (&list2);

_dbus_assert (is_ascending_sequence (&list1));
_dbus_assert (is_descending_sequence (&list2));

--i;
}

_dbus_assert (list1 == NULL);
_dbus_assert (list2 == NULL);

/* Test get_first_link, get_last_link, pop_first_link, pop_last_link
*/

i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;
}

--i;
while (i >= 0)
{
    DBusList *got_link1;
    DBusList *got_link2;

    DBusList *link2;

    void *data1_indirect;
    void *data1;
    void *data2;

    got_link1 = _dbus_list_get_last_link (&list1);
    got_link2 = _dbus_list_get_first_link (&list2);

```

```

link2 = _dbus_list_pop_first_link (&list2);

_dbus_assert (got_link2 == link2);

data1_indirect = got_link1->data;
/* this call makes got_link1 invalid */
data1 = _dbus_list_pop_last (&list1);
_dbus_assert (data1 == data1_indirect);
data2 = link2->data;

_dbus_list_free_link (link2);

_dbus_assert (_DBUS_POINTER_TO_INT (data1) == i);
_dbus_assert (_DBUS_POINTER_TO_INT (data2) == i);

verify_list (&list1);
verify_list (&list2);

_dbus_assert (is_ascending_sequence (&list1));
_dbus_assert (is_descending_sequence (&list2));

--i;
}

_dbus_assert (list1 == NULL);
_dbus_assert (list2 == NULL);

/* Test iteration */

i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;

    verify_list (&list1);
    verify_list (&list2);

    _dbus_assert (_dbus_list_get_length (&list1) == i);
    _dbus_assert (_dbus_list_get_length (&list2) == i);
}

_dbus_assert (is_ascending_sequence (&list1));
_dbus_assert (is_descending_sequence (&list2));

--i;
link2 = _dbus_list_get_first_link (&list2);
while (link2 != NULL)
{
    verify_list (&link2); /* pretend this link is the head */
}

```

```

    _dbus_assert (_DBUS_POINTER_TO_INT (link2->data) == i);

    link2 = _dbus_list_get_next_link (&list2, link2);
    --i;
}

i = 0;
link1 = _dbus_list_get_first_link (&list1);
while (link1 != NULL)
{
    verify_list (&link1); /* pretend this link is the head */

    _dbus_assert (_DBUS_POINTER_TO_INT (link1->data) == i);

    link1 = _dbus_list_get_next_link (&list1, link1);
    ++i;
}

--i;
link1 = _dbus_list_get_last_link (&list1);
while (link1 != NULL)
{
    verify_list (&link1); /* pretend this link is the head */

    _dbus_assert (_DBUS_POINTER_TO_INT (link1->data) == i);

    link1 = _dbus_list_get_prev_link (&list1, link1);
    --i;
}

_dbus_list_clear (&list1);
_dbus_list_clear (&list2);

/* Test remove */

i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;
}

--i;
while (i >= 0)
{
    if ((i % 2) == 0)
    {
        if (!_dbus_list_remove (&list1, _DBUS_INT_TO_POINTER (i)))
            _dbus_assert_not_reached ("element should have been in
list");
    }
}

```

```

        if (!_dbus_list_remove (&list2, _DBUS_INT_TO_POINTER (i)))
            _dbus_assert_not_reached ("element should have been in
list");

        verify_list (&list1);
        verify_list (&list2);
    }
    --i;
}

_dbus_assert (all_odd_values (&list1));
_dbus_assert (all_odd_values (&list2));

_dbus_list_clear (&list1);
_dbus_list_clear (&list2);

/* test removing the other half of the elements */

i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;
}

--i;
while (i >= 0)
{
    if ((i % 2) != 0)
    {
        if (!_dbus_list_remove (&list1, _DBUS_INT_TO_POINTER (i)))
            _dbus_assert_not_reached ("element should have been in
list");
        if (!_dbus_list_remove (&list2, _DBUS_INT_TO_POINTER (i)))
            _dbus_assert_not_reached ("element should have been in
list");

        verify_list (&list1);
        verify_list (&list2);
    }
    --i;
}

_dbus_assert (all_even_values (&list1));
_dbus_assert (all_even_values (&list2));

/* clear list using remove_link */
while (list1 != NULL)
{
    _dbus_list_remove_link (&list1, list1);
    verify_list (&list1);
}

```

```

    }
while (list2 != NULL)
    {
        _dbus_list_remove_link (&list2, list2);
        verify_list (&list2);
    }

/* Test remove link more generally */
i = 0;
while (i < 10)
    {
        _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
        _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
        ++i;
    }

--i;
link2 = _dbus_list_get_first_link (&list2);
while (link2 != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (&list2, link2);

        _dbus_assert (_DBUS_POINTER_TO_INT (link2->data) == i);

        if ((i % 2) == 0)
            _dbus_list_remove_link (&list2, link2);

        verify_list (&list2);

        link2 = next;
        --i;
    }

_dbus_assert (all_odd_values (&list2));
_dbus_list_clear (&list2);

i = 0;
link1 = _dbus_list_get_first_link (&list1);
while (link1 != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (&list1, link1);

        _dbus_assert (_DBUS_POINTER_TO_INT (link1->data) == i);

        if ((i % 2) != 0)
            _dbus_list_remove_link (&list1, link1);

        verify_list (&list1);

        link1 = next;
        ++i;
    }

```



```

_dbus_assert (all_even_values (&list1));
_dbus_list_clear (&list1);

/* Test copying a list */
i = 0;
while (i < 10)
{
    _dbus_list_append (&list1, _DBUS_INT_TO_POINTER (i));
    _dbus_list_prepend (&list2, _DBUS_INT_TO_POINTER (i));
    ++i;
}

/* bad pointers, because they are allowed in the copy dest */
copy1 = _DBUS_INT_TO_POINTER (0x342234);
copy2 = _DBUS_INT_TO_POINTER (23);

_dbus_list_copy (&list1, &copy1);
verify_list (&list1);
verify_list (&copy1);
_dbus_assert (lists_equal (&list1, &copy1));

_dbus_list_copy (&list2, &copy2);
verify_list (&list2);
verify_list (&copy2);
_dbus_assert (lists_equal (&list2, &copy2));

/* Now test copying empty lists */
_dbus_list_clear (&list1);
_dbus_list_clear (&list2);
_dbus_list_clear (&copy1);
_dbus_list_clear (&copy2);

/* bad pointers, because they are allowed in the copy dest */
copy1 = _DBUS_INT_TO_POINTER (0x342234);
copy2 = _DBUS_INT_TO_POINTER (23);

_dbus_list_copy (&list1, &copy1);
verify_list (&list1);
verify_list (&copy1);
_dbus_assert (lists_equal (&list1, &copy1));

_dbus_list_copy (&list2, &copy2);
verify_list (&list2);
verify_list (&copy2);
_dbus_assert (lists_equal (&list2, &copy2));

_dbus_list_clear (&list1);
_dbus_list_clear (&list2);

/* insert_after on empty list */
_dbus_list_insert_after (&list1, NULL,

```

```

        _DBUS_INT_TO_POINTER (0));
verify_list (&list1);

/* inserting after first element */
_dbus_list_insert_after (&list1, list1,
        _DBUS_INT_TO_POINTER (1));
verify_list (&list1);
_dbus_assert (is_ascending_sequence (&list1));

/* inserting at the end */
_dbus_list_insert_after (&list1, list1->next,
        _DBUS_INT_TO_POINTER (2));
verify_list (&list1);
_dbus_assert (is_ascending_sequence (&list1));

/* using insert_after to prepend */
_dbus_list_insert_after (&list1, NULL,
        _DBUS_INT_TO_POINTER (-1));
verify_list (&list1);
_dbus_assert (is_ascending_sequence (&list1));

_dbus_list_clear (&list1);

/* using remove_last */
_dbus_list_append (&list1, _DBUS_INT_TO_POINTER (2));
_dbus_list_append (&list1, _DBUS_INT_TO_POINTER (1));
_dbus_list_append (&list1, _DBUS_INT_TO_POINTER (3));

_dbus_list_remove_last (&list1, _DBUS_INT_TO_POINTER (2));

verify_list (&list1);
_dbus_assert (is_ascending_sequence (&list1));

_dbus_list_clear (&list1);

return TRUE;
}

```

```
#endif
```

```
File = dbus-list.h
```

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-list.h Generic linked list utility (internal to D-Bus
implementation)
*
* Copyright (C) 2002, 2003 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*

```



```

dbus_bool_t _dbus_list_remove          (DBusList *list,
                                         void *data);
dbus_bool_t _dbus_list_remove_last    (DBusList *list,
                                         void *data);
void        _dbus_list_remove_link    (DBusList *list,
                                         DBusList *link);
DBusList*   _dbus_list_find_last      (DBusList *list,
                                         void *data);
void        _dbus_list_clear          (DBusList *list);
DBusList*   _dbus_list_get_first_link (DBusList *list);
DBusList*   _dbus_list_get_last_link  (DBusList *list);
void*       _dbus_list_get_last       (DBusList *list);
void*       _dbus_list_get_first      (DBusList *list);
void*       _dbus_list_pop_first      (DBusList *list);
void*       _dbus_list_pop_last       (DBusList *list);
DBusList*   _dbus_list_pop_first_link (DBusList *list);
dbus_bool_t _dbus_list_copy           (DBusList *list,
                                         DBusList **dest);

int         _dbus_list_get_length     (DBusList *list);
DBusList*   _dbus_list_alloc_link     (void *data);
void        _dbus_list_free_link      (DBusList *link);
void        _dbus_list_unlink         (DBusList *list,
                                         DBusList *link);

void        _dbus_list_append_link    (DBusList *list,
                                         DBusList *link);
void        _dbus_list_prepend_link   (DBusList *list,
                                         DBusList *link);
dbus_bool_t _dbus_list_length_is_one  (DBusList *list);

void _dbus_list_foreach (DBusList **list,
                        DBusForeachFunction function,
                        void *data);

#define _dbus_list_get_next_link(list, link) ((link)->next == *(list)
? NULL : (link)->next)
#define _dbus_list_get_prev_link(list, link) ((link) == *(list) ? NULL
: (link)->prev)

/* if DBUS_ENABLE_STATS */
void        _dbus_list_get_stats      (dbus_uint32_t *in_use_p,
                                         dbus_uint32_t
*in_free_list_p,
                                         dbus_uint32_t
*allocated_p);

DBUS_END_DECLS

#endif /* DBUS_LIST_H */

```

File = dbus-macros.h

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-macros.h generic macros
 *
 * Copyright (C) 2002 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_MACROS_H
#define DBUS_MACROS_H

#ifdef __cplusplus
# define DBUS_BEGIN_DECLS extern "C" {
# define DBUS_END_DECLS }
#else
# define DBUS_BEGIN_DECLS
# define DBUS_END_DECLS
#endif

#ifndef TRUE
# define TRUE 1
#endif

#ifndef FALSE
# define FALSE 0
#endif

#ifndef NULL
```

```

# ifdef __cplusplus
#   define NULL          (0L)
# else /* !__cplusplus */
#   define NULL          ((void*) 0)
# endif /* !__cplusplus */
#endif

#if __GNUC__ > 3 || (__GNUC__ == 3 && __GNUC_MINOR__ >= 1)
# define DBUS_DEPRECATED __attribute__((__deprecated__))
#elif defined(_MSC_VER) && (_MSC_VER >= 1300)
# define DBUS_DEPRECATED __declspec(deprecated)
#else
# define DBUS_DEPRECATED
#endif

#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__ >= 8)
# define _DBUS_GNUC_EXTENSION __extension__
#else
# define _DBUS_GNUC_EXTENSION
#endif

#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__ > 4)
#define _DBUS_GNUC_PRINTF( format_idx, arg_idx ) \
    __attribute__((__format__(__printf__, format_idx, arg_idx)))
#define _DBUS_GNUC_NORETURN \
    __attribute__((__noreturn__))
#else /* !__GNUC__ */
#define _DBUS_GNUC_PRINTF( format_idx, arg_idx )
#define _DBUS_GNUC_NORETURN
#endif /* !__GNUC__ */

#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__ >= 96)
#define DBUS_MALLOC __attribute__((__malloc__))
#else
#define DBUS_MALLOC
#endif

#if (__GNUC__ > 4) || (__GNUC__ == 4 && __GNUC_MINOR__ >= 3)
#define DBUS_ALLOC_SIZE(x) __attribute__((__alloc_size__(x)))
#define DBUS_ALLOC_SIZE2(x,y) __attribute__((__alloc_size__(x,y)))
#else
#define DBUS_ALLOC_SIZE(x)
#define DBUS_ALLOC_SIZE2(x,y)
#endif

/** @def _DBUS_GNUC_PRINTF
 * used to tell gcc about printf format strings
 */
/** @def _DBUS_GNUC_NORETURN
 * used to tell gcc about functions that never return, such as
_dbus_abort()
 */

```

```

/* Normally docs are in .c files, but there isn't a .c file for this.
*/
/**
 * @defgroup DBusMacros Utility macros
 * @ingroup DBus
 * @brief #TRUE, #FALSE, #NULL, and so on
 *
 * Utility macros.
 *
 * @{
 */

/**
 * @def DBUS_BEGIN_DECLS
 *
 * Macro used prior to declaring functions in the D-Bus header
 * files. Expands to "extern "C"" when using a C++ compiler,
 * and expands to nothing when using a C compiler.
 *
 * Please don't use this in your own code, consider it
 * D-Bus internal.
 */
/**
 * @def DBUS_END_DECLS
 *
 * Macro used after declaring functions in the D-Bus header
 * files. Expands to "}" when using a C++ compiler,
 * and expands to nothing when using a C compiler.
 *
 * Please don't use this in your own code, consider it
 * D-Bus internal.
 */
/**
 * @def TRUE
 *
 * Expands to "1"
 */
/**
 * @def FALSE
 *
 * Expands to "0"
 */
/**
 * @def NULL
 *
 * A null pointer, defined appropriately for C or C++.
 */
/**
 * @def DBUS_DEPRECATED
 *

```

```

* Tells the compiler to warn about a function or type if it's used.
* Code marked in this way should also be enclosed in
* @code
* #ifndef DBUS_DISABLE_DEPRECATED
*   deprecated stuff here
* #endif
* @endcode
*
* Please don't use this in your own code, consider it
* D-Bus internal.
*/
/**
* @def _DBUS_GNUC_EXTENSION
*
* Tells gcc not to warn about extensions to the C standard in the
* following expression, even if compiling with -pedantic. Do not use
* this macro in your own code; please consider it to be internal to
libdbus.
*/

/*
* @def DBUS_EXPORT
*
* Declare the following symbol as public. This is currently a noop
on
* platforms other than Windows.
*/

#if defined(_WIN32)
# if defined(DBUS_STATIC_BUILD)
# define DBUS_EXPORT
# elif defined(dbus_1_EXPORTS)
# define DBUS_EXPORT __declspec(dllexport)
# else
# define DBUS_EXPORT __declspec(dllimport)
# endif
#else
#define DBUS_EXPORT
#endif

/** @} */

#endif /* DBUS_MACROS_H */

File = dbus-mainloop.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-mainloop.c Main loop utility
*
* Copyright © 2003, 2004 Red Hat, Inc.

```



```
* Copyright Â© 2011 Nokia Corporation
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-mainloop.h"

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include <dbus/dbus-hash.h>
#include <dbus/dbus-list.h>
#include <dbus/dbus-socket-set.h>
#include <dbus/dbus-watch.h>

#define MAINLOOP_SPEW 0

#if MAINLOOP_SPEW
#ifdef DBUS_ENABLE_VERBOSE_MODE
static const char*
watch_flags_to_string (int flags)
{
    const char *watch_type;

    if ((flags & DBUS_WATCH_READABLE) &&
        (flags & DBUS_WATCH_WRITABLE))
        watch_type = "readwrite";
    else if (flags & DBUS_WATCH_READABLE)
        watch_type = "read";
    else if (flags & DBUS_WATCH_WRITABLE)
        watch_type = "write";
    else
        watch_type = "not read or write";
    return watch_type;
}
#endif
#endif
```

```

}
#endif /* DBUS_ENABLE_VERBOSE_MODE */
#endif /* MAINLOOP_SPEW */

struct DBusLoop
{
    int refcount;
    /** fd => dbus_malloc'd DBusList ** of references to DBusWatch */
    DBusHashTable *watches;
    DBusSocketSet *socket_set;
    DBusList *timeouts;
    int callback_list_serial;
    int watch_count;
    int timeout_count;
    int depth; /**< number of recursive runs */
    DBusList *need_dispatch;
    /** TRUE if we will skip a watch next time because it was OOM;
becomes
    * FALSE between polling, and dealing with the results of the poll
*/
    unsigned oom_watch_pending : 1;
};

typedef struct
{
    DBusTimeout *timeout;
    unsigned long last_tv_sec;
    unsigned long last_tv_usec;
} TimeoutCallback;

#define TIMEOUT_CALLBACK(callback) ((TimeoutCallback*)callback)

static TimeoutCallback*
timeout_callback_new (DBusTimeout          *timeout)
{
    TimeoutCallback *cb;

    cb = dbus_new (TimeoutCallback, 1);
    if (cb == NULL)
        return NULL;

    cb->timeout = timeout;
    _dbus_get_monotonic_time (&cb->last_tv_sec,
                             &cb->last_tv_usec);

    return cb;
}

static void
timeout_callback_free (TimeoutCallback *cb)
{
    dbus_free (cb);
}

```

```

static void
free_watch_table_entry (void *data)
{
    DBusList **watches = data;
    DBusWatch *watch;

    /* DBusHashTable sometimes calls free_function(NULL) even if you
never
    * have NULL as a value */
    if (watches == NULL)
        return;

    for (watch = _dbus_list_pop_first (watches);
        watch != NULL;
        watch = _dbus_list_pop_first (watches))
    {
        _dbus_watch_unref (watch);
    }

    _dbus_assert (*watches == NULL);
    dbus_free (watches);
}

DBusLoop*
_dbus_loop_new (void)
{
    DBusLoop *loop;

    loop = dbus_new0 (DBusLoop, 1);
    if (loop == NULL)
        return NULL;

    loop->watches = _dbus_hash_table_new (DBUS_HASH_INT, NULL,
                                         free_watch_table_entry);

    loop->socket_set = _dbus_socket_set_new (0);

    if (loop->watches == NULL || loop->socket_set == NULL)
    {
        if (loop->watches != NULL)
            _dbus_hash_table_unref (loop->watches);

        if (loop->socket_set != NULL)
            _dbus_socket_set_free (loop->socket_set);

        dbus_free (loop);
        return NULL;
    }

    loop->refcount = 1;
}

```

```

    return loop;
}

DBusLoop *
_dbus_loop_ref (DBusLoop *loop)
{
    _dbus_assert (loop != NULL);
    _dbus_assert (loop->refcount > 0);

    loop->refcount += 1;

    return loop;
}

void
_dbus_loop_unref (DBusLoop *loop)
{
    _dbus_assert (loop != NULL);
    _dbus_assert (loop->refcount > 0);

    loop->refcount -= 1;
    if (loop->refcount == 0)
    {
        while (loop->need_dispatch)
        {
            DBusConnection *connection = _dbus_list_pop_first (&loop-
>need_dispatch);

            dbus_connection_unref (connection);
        }

        _dbus_hash_table_unref (loop->watches);
        _dbus_socket_set_free (loop->socket_set);
        dbus_free (loop);
    }
}

static DBusList **
ensure_watch_table_entry (DBusLoop *loop,
                          int        fd)
{
    DBusList **watches;

    watches = _dbus_hash_table_lookup_int (loop->watches, fd);

    if (watches == NULL)
    {
        watches = dbus_new0 (DBusList *, 1);

        if (watches == NULL)
            return watches;
    }
}

```

```

        if (!_dbus_hash_table_insert_int (loop->watches, fd, watches))
        {
            dbus_free (watches);
            watches = NULL;
        }
    }

    return watches;
}

static void
cull_watches_for_invalid_fd (DBusLoop *loop,
                             int fd)
{
    DBusList *link;
    DBusList **watches;

    _dbus_warn ("invalid request, socket fd %d not open\n", fd);
    watches = _dbus_hash_table_lookup_int (loop->watches, fd);

    if (watches != NULL)
    {
        for (link = _dbus_list_get_first_link (watches);
             link != NULL;
             link = _dbus_list_get_next_link (watches, link))
            _dbus_watch_invalidate (link->data);
    }

    _dbus_hash_table_remove_int (loop->watches, fd);
}

static dbus_bool_t
gc_watch_table_entry (DBusLoop *loop,
                     DBusList **watches,
                     int fd)
{
    /* If watches is already NULL we have nothing to do */
    if (watches == NULL)
        return FALSE;

    /* We can't GC hash table entries if they're non-empty lists */
    if (*watches != NULL)
        return FALSE;

    _dbus_hash_table_remove_int (loop->watches, fd);
    return TRUE;
}

static void
refresh_watches_for_fd (DBusLoop *loop,
                       DBusList **watches,
                       int fd)

```

```

{
    DBusList *link;
    unsigned int flags = 0;
    dbus_bool_t interested = FALSE;

    _dbus_assert (fd != -1);

    if (watches == NULL)
        watches = _dbus_hash_table_lookup_int (loop->watches, fd);

    /* we allocated this in the first _dbus_loop_add_watch for the fd,
and keep
    * it until there are none left */
    _dbus_assert (watches != NULL);

    for (link = _dbus_list_get_first_link (watches);
        link != NULL;
        link = _dbus_list_get_next_link (watches, link))
    {
        if (dbus_watch_get_enabled (link->data) &&
            !_dbus_watch_get_oom_last_time (link->data))
        {
            flags |= dbus_watch_get_flags (link->data);
            interested = TRUE;
        }
    }

    if (interested)
        _dbus_socket_set_enable (loop->socket_set, fd, flags);
    else
        _dbus_socket_set_disable (loop->socket_set, fd);
}

dbus_bool_t
_dbus_loop_add_watch (DBusLoop *loop,
                     DBusWatch *watch)
{
    int fd;
    DBusList **watches;

    fd = dbus_watch_get_socket (watch);
    _dbus_assert (fd != -1);

    watches = ensure_watch_table_entry (loop, fd);

    if (watches == NULL)
        return FALSE;

    if (!_dbus_list_append (watches, _dbus_watch_ref (watch)))
    {
        _dbus_watch_unref (watch);
        gc_watch_table_entry (loop, watches, fd);
    }
}

```

```

        return FALSE;
    }

    if (_dbus_list_length_is_one (watches))
    {
        if (!_dbus_socket_set_add (loop->socket_set, fd,
                                   dbus_watch_get_flags (watch),
                                   dbus_watch_get_enabled (watch)))
        {
            _dbus_hash_table_remove_int (loop->watches, fd);
            return FALSE;
        }
    }
    else
    {
        /* we're modifying, not adding, which can't fail with OOM */
        refresh_watches_for_fd (loop, watches, fd);
    }

    loop->callback_list_serial += 1;
    loop->watch_count += 1;
    return TRUE;
}

void
_dbus_loop_toggle_watch (DBusLoop      *loop,
                        DBusWatch     *watch)
{
    refresh_watches_for_fd (loop, NULL, dbus_watch_get_socket (watch));
}

void
_dbus_loop_remove_watch (DBusLoop      *loop,
                        DBusWatch     *watch)
{
    DBusList **watches;
    DBusList *link;
    int fd;

    /* This relies on people removing watches before they invalidate
    them,
    * which has been safe since fd.o #33336 was fixed. Assert about it
    * so we don't regress. */
    fd = dbus_watch_get_socket (watch);
    _dbus_assert (fd != -1);

    watches = _dbus_hash_table_lookup_int (loop->watches, fd);

    if (watches != NULL)
    {
        link = _dbus_list_get_first_link (watches);

```

```

while (link != NULL)
{
    DBusList *next = _dbus_list_get_next_link (watches, link);
    DBusWatch *this = link->data;

    if (this == watch)
    {
        _dbus_list_remove_link (watches, link);
        loop->callback_list_serial += 1;
        loop->watch_count -= 1;
        _dbus_watch_unref (this);

        /* if that was the last watch for that fd, drop the hash
table
set */
        if (gc_watch_table_entry (loop, watches, fd))
        {
            _dbus_socket_set_remove (loop->socket_set, fd);
        }

        return;
    }

    link = next;
}

_dbus_warn ("could not find watch %p to remove\n", watch);
}

dbus_bool_t
_dbus_loop_add_timeout (DBusLoop          *loop,
                       DBusTimeout       *timeout)
{
    TimeoutCallback *tcb;

    tcb = timeout_callback_new (timeout);
    if (tcb == NULL)
        return FALSE;

    if (_dbus_list_append (&loop->timeouts, tcb))
    {
        loop->callback_list_serial += 1;
        loop->timeout_count += 1;
    }
    else
    {
        timeout_callback_free (tcb);
        return FALSE;
    }
}

```



```

    return TRUE;
}

void
_dbus_loop_remove_timeout (DBusLoop          *loop,
                           DBusTimeout      *timeout)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&loop->timeouts);
    while (link != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (&loop->timeouts,
link);
        TimeoutCallback *this = link->data;

        if (this->timeout == timeout)
        {
            _dbus_list_remove_link (&loop->timeouts, link);
            loop->callback_list_serial += 1;
            loop->timeout_count -= 1;
            timeout_callback_free (this);

            return;
        }

        link = next;
    }

    _dbus_warn ("could not find timeout %p to remove\n", timeout);
}

/* Convolutions from GLib, there really must be a better way
 * to do this.
 */
static dbus_bool_t
check_timeout (unsigned long    tv_sec,
               unsigned long    tv_usec,
               TimeoutCallback *tcb,
               int              *timeout)
{
    long sec_remaining;
    long msec_remaining;
    unsigned long expiration_tv_sec;
    unsigned long expiration_tv_usec;
    long interval_seconds;
    long interval_milliseconds;
    int interval;

    /* I'm pretty sure this function could suck (a lot) less */

    interval = dbus_timeout_get_interval (tcb->timeout);

```

```

interval_seconds = interval / 1000L;
interval_milliseconds = interval % 1000L;

expiration_tv_sec = tcb->last_tv_sec + interval_seconds;
expiration_tv_usec = tcb->last_tv_usec + interval_milliseconds *
1000;
if (expiration_tv_usec >= 1000000)
{
    expiration_tv_usec -= 1000000;
    expiration_tv_sec += 1;
}

sec_remaining = expiration_tv_sec - tv_sec;
/* need to force this to be signed, as it is intended to sometimes
 * produce a negative result
 */
msec_remaining = ((long) expiration_tv_usec - (long) tv_usec) /
1000L;

#if MAINLOOP_SPEW
    _dbus_verbose ("Interval is %ld seconds %ld msec\n",
        interval_seconds,
        interval_milliseconds);
    _dbus_verbose ("Now is %lu seconds %lu usecs\n",
        tv_sec, tv_usec);
    _dbus_verbose ("Last is %lu seconds %lu usecs\n",
        tcb->last_tv_sec, tcb->last_tv_usec);
    _dbus_verbose ("Exp is %lu seconds %lu usecs\n",
        expiration_tv_sec, expiration_tv_usec);
    _dbus_verbose ("Pre-correction, sec_remaining %ld msec_remaining
%ld\n",
        sec_remaining, msec_remaining);
#endif

/* We do the following in a rather convoluted fashion to deal with
 * the fact that we don't have an integral type big enough to hold
 * the difference of two timevals in milliseconds.
 */
if (sec_remaining < 0 || (sec_remaining == 0 && msec_remaining < 0))
{
    *timeout = 0;
}
else
{
    if (msec_remaining < 0)
    {
        msec_remaining += 1000;
        sec_remaining -= 1;
    }

    if (sec_remaining > (_DBUS_INT_MAX / 1000) ||

```

```

        msec_remaining > _DBUS_INT_MAX)
        *timeout = _DBUS_INT_MAX;
    else
        *timeout = sec_remaining * 1000 + msec_remaining;
    }

    if (*timeout > interval)
    {
        /* This indicates that the system clock probably moved backward
*/
        _dbus_verbose ("System clock set backward! Resetting
timeout.\n");

        tcb->last_tv_sec = tv_sec;
        tcb->last_tv_usec = tv_usec;

        *timeout = interval;
    }

#ifdef MAINLOOP_SPEW
    _dbus_verbose (" timeout expires in %d milliseconds\n", *timeout);
#endif

    return *timeout == 0;
}

dbus_bool_t
_dbus_loop_dispatch (DBusLoop *loop)
{
#ifdef MAINLOOP_SPEW
    _dbus_verbose (" %d connections to dispatch\n",
_dbus_list_get_length (&loop->need_dispatch));
#endif

    if (loop->need_dispatch == NULL)
        return FALSE;

next:
    while (loop->need_dispatch != NULL)
    {
        DBusConnection *connection = _dbus_list_pop_first (&loop-
>need_dispatch);

        while (TRUE)
        {
            DBusDispatchStatus status;

            status = dbus_connection_dispatch (connection);

            if (status == DBUS_DISPATCH_COMPLETE)
            {

```

```

        dbus_connection_unref (connection);
        goto next;
    }
    else
    {
        if (status == DBUS_DISPATCH_NEED_MEMORY)
            _dbus_wait_for_memory ();
    }
}

return TRUE;
}

dbus_bool_t
_dbus_loop_queue_dispatch (DBusLoop      *loop,
                          DBusConnection *connection)
{
    if (_dbus_list_append (&loop->need_dispatch, connection))
    {
        dbus_connection_ref (connection);
        return TRUE;
    }
    else
        return FALSE;
}

/* Returns TRUE if we invoked any timeouts or have ready file
 * descriptors, which is just used in test code as a debug hack
 */

dbus_bool_t
_dbus_loop_iterate (DBusLoop      *loop,
                   dbus_bool_t    block)
{
#define N_STACK_DESCRIPTOR 64
    dbus_bool_t retval;
    DBusSocketEvent ready_fds[N_STACK_DESCRIPTOR];
    int i;
    DBusList *link;
    int n_ready;
    int initial_serial;
    long timeout;
    int orig_depth;

    retval = FALSE;

    orig_depth = loop->depth;

#ifdef MAINLOOP_SPEW
    _dbus_verbose ("Iteration block=%d depth=%d timeout_count=%d
watch_count=%d\n",

```

```

        block, loop->depth, loop->timeout_count, loop-
>watch_count);
#endif

    if (_dbus_hash_table_get_n_entries (loop->watches) == 0 &&
        loop->timeouts == NULL)
        goto next_iteration;

    timeout = -1;
    if (loop->timeout_count > 0)
    {
        unsigned long tv_sec;
        unsigned long tv_usec;

        _dbus_get_monotonic_time (&tv_sec, &tv_usec);

        link = _dbus_list_get_first_link (&loop->timeouts);
        while (link != NULL)
        {
            DBusList *next = _dbus_list_get_next_link (&loop->timeouts,
link);
            TimeoutCallback *tcb = link->data;

            if (dbus_timeout_get_enabled (tcb->timeout))
            {
                int msecs_remaining;

                check_timeout (tv_sec, tv_usec, tcb, &msecs_remaining);

                if (timeout < 0)
                    timeout = msecs_remaining;
                else
                    timeout = MIN (msecs_remaining, timeout);

#if MAINLOOP_SPEW
                _dbus_verbose (" timeout added, %d remaining, aggregate
timeout %ld\n",
                                msecs_remaining, timeout);
#endif
            }

            _dbus_assert (timeout >= 0);

            if (timeout == 0)
                break; /* it's not going to get shorter... */
        }
    }
#if MAINLOOP_SPEW
    else
    {
        _dbus_verbose (" skipping disabled timeout\n");
    }
#endif
#endif

```

```

        link = next;
    }
}

/* Never block if we have stuff to dispatch */
if (!block || loop->need_dispatch != NULL)
{
    timeout = 0;
#ifdef MAINLOOP_SPEW
    _dbus_verbose (" timeout is 0 as we aren't blocking\n");
#endif
}

/* if a watch was OOM last time, don't wait longer than the OOM
 * wait to re-enable it
 */
if (loop->oom_watch_pending)
    timeout = MIN (timeout, _dbus_get_oom_wait ());

#ifdef MAINLOOP_SPEW
    _dbus_verbose (" polling on %d descriptors timeout %ld\n", n_fds,
timeout);
#endif

n_ready = _dbus_socket_set_poll (loop->socket_set, ready_fds,
                                _DBUS_N_ELEMENTS (ready_fds),
timeout);

/* re-enable any watches we skipped this time */
if (loop->oom_watch_pending)
{
    DBusHashIter hash_iter;

    loop->oom_watch_pending = FALSE;

    _dbus_hash_iter_init (loop->watches, &hash_iter);

    while (_dbus_hash_iter_next (&hash_iter))
    {
        DBusList **watches;
        int fd;
        dbus_bool_t changed;

        changed = FALSE;
        fd = _dbus_hash_iter_get_int_key (&hash_iter);
        watches = _dbus_hash_iter_get_value (&hash_iter);

        for (link = _dbus_list_get_first_link (watches);
            link != NULL;
            link = _dbus_list_get_next_link (watches, link))
        {
            DBusWatch *watch = link->data;

```

```

        if (_dbus_watch_get_oom_last_time (watch))
        {
            _dbus_watch_set_oom_last_time (watch, FALSE);
            changed = TRUE;
        }
    }

    if (changed)
        refresh_watches_for_fd (loop, watches, fd);
}

retval = TRUE; /* return TRUE here to keep the loop going,
                * since we don't know the watch was inactive */
}

initial_serial = loop->callback_list_serial;

if (loop->timeout_count > 0)
{
    unsigned long tv_sec;
    unsigned long tv_usec;

    _dbus_get_monotonic_time (&tv_sec, &tv_usec);

    /* It'd be nice to avoid this O(n) thingy here */
    link = _dbus_list_get_first_link (&loop->timeouts);
    while (link != NULL)
    {
        DBusList *next = _dbus_list_get_next_link (&loop->timeouts,
link);
        TimeoutCallback *tcb = link->data;

        if (initial_serial != loop->callback_list_serial)
            goto next_iteration;

        if (loop->depth != orig_depth)
            goto next_iteration;

        if (dbus_timeout_get_enabled (tcb->timeout))
        {
            int msecs_remaining;

            if (check_timeout (tv_sec, tv_usec,
                              tcb, &msecs_remaining))
            {
                /* Save last callback time and fire this timeout */
                tcb->last_tv_sec = tv_sec;
                tcb->last_tv_usec = tv_usec;
            }
        }
    }
}

#ifdef MAINLOOP_SPEW
    _dbus_verbose (" invoking timeout\n");
#endif

```

```

#endif

just          /* can theoretically return FALSE on OOM, but we
              * let it fire again later - in practice that's what
              * every wrapper callback in dbus-daemon used to do
*/
              dbus_timeout_handle (tcb->timeout);

              retval = TRUE;
              }
              else
              {
#if MAINLOOP_SPEW
              _dbus_verbose (" timeout has not expired\n");
#endif
              }
              }
#if MAINLOOP_SPEW
              else
              {
timeout\n");
              }
#endif

              link = next;
              }
              }

if (n_ready > 0)
{
    for (i = 0; i < n_ready; i++)
    {
        DBusList **watches;
        DBusList *next;
        unsigned int condition;
        dbus_bool_t any_oom;

        /* FIXME I think this "restart if we change the watches"
         * approach could result in starving watches
         * toward the end of the list.
         */
        if (initial_serial != loop->callback_list_serial)
            goto next_iteration;

        if (loop->depth != orig_depth)
            goto next_iteration;

        _dbus_assert (ready_fds[i].flags != 0);

        if (_DBUS_UNLIKELY (ready_fds[i].flags & _DBUS_WATCH_NVAL))

```



```

    {
        cull_watches_for_invalid_fd (loop, ready_fds[i].fd);
        goto next_iteration;
    }

condition = ready_fds[i].flags;
_dbus_assert ((condition & _DBUS_WATCH_NVAL) == 0);

/* condition may still be 0 if we got some
 * weird POLLFOO thing like POLLWRBAND
 */
if (condition == 0)
    continue;

watches = _dbus_hash_table_lookup_int (loop->watches,
                                       ready_fds[i].fd);

if (watches == NULL)
    continue;

any_oom = FALSE;

for (link = _dbus_list_get_first_link (watches);
     link != NULL;
     link = next)
    {
        DBusWatch *watch = link->data;

        next = _dbus_list_get_next_link (watches, link);

        if (dbus_watch_get_enabled (watch))
            {
                dbus_bool_t oom;

                oom = !dbus_watch_handle (watch, condition);

                if (oom)
                    {
                        _dbus_watch_set_oom_last_time (watch, TRUE);
                        loop->oom_watch_pending = TRUE;
                        any_oom = TRUE;
                    }
            }

#ifdef MAINLOOP_SPEW
        _dbus_verbose ("  Invoked watch, oom = %d\n", oom);
#endif

        retval = TRUE;

        /* We re-check this every time, in case the callback
         * added/removed watches, which might make our
         * the linked list invalid. See the FIXME above. */
    }
}

```

```

        if (initial_serial != loop->callback_list_serial ||
            loop->depth != orig_depth)
        {
            if (any_oom)
                refresh_watches_for_fd (loop, NULL,
ready_fds[i].fd);

                goto next_iteration;
        }
    }

    if (any_oom)
        refresh_watches_for_fd (loop, watches, ready_fds[i].fd);
}

next_iteration:
#ifdef MAINLOOP_SPEW
    _dbus_verbose (" moving to next iteration\n");
#endif

    if (_dbus_loop_dispatch (loop))
        retval = TRUE;

#ifdef MAINLOOP_SPEW
    _dbus_verbose ("Returning %d\n", retval);
#endif

    return retval;
}

void
_dbus_loop_run (DBusLoop *loop)
{
    int our_exit_depth;

    _dbus_assert (loop->depth >= 0);

    _dbus_loop_ref (loop);

    our_exit_depth = loop->depth;
    loop->depth += 1;

    _dbus_verbose ("Running main loop, depth %d -> %d\n",
                    loop->depth - 1, loop->depth);

    while (loop->depth != our_exit_depth)
        _dbus_loop_iterate (loop, TRUE);

    _dbus_loop_unref (loop);
}

```

```

void
_dbus_loop_quit (DBusLoop *loop)
{
    _dbus_assert (loop->depth > 0);

    loop->depth -= 1;

    _dbus_verbose ("Quit main loop, depth %d -> %d\n",
                  loop->depth + 1, loop->depth);
}

int
_dbus_get_oom_wait (void)
{
#ifdef DBUS_BUILD_TESTS
    /* make tests go fast */
    return 0;
#else
    return 500;
#endif
}

void
_dbus_wait_for_memory (void)
{
    _dbus_verbose ("Waiting for more memory\n");
    _dbus_sleep_milliseconds (_dbus_get_oom_wait ());
}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

```

File = dbus-mainloop.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-mainloop.h  Main loop utility
 *
 * Copyright (C) 2003  Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of

```

```

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef DBUS_MAINLOOP_H
#define DBUS_MAINLOOP_H

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include <dbus/dbus.h>

typedef struct DBusLoop DBusLoop;

typedef dbus_bool_t (* DBusWatchFunction) (DBusWatch *watch,
                                           unsigned int condition,
                                           void *data);

DBusLoop* _dbus_loop_new (void);
DBusLoop* _dbus_loop_ref (DBusLoop *loop);
void _dbus_loop_unref (DBusLoop *loop);
dbus_bool_t _dbus_loop_add_watch (DBusLoop *loop,
                                  DBusWatch *watch);
void _dbus_loop_remove_watch (DBusLoop *loop,
                              DBusWatch *watch);
void _dbus_loop_toggle_watch (DBusLoop *loop,
                              DBusWatch *watch);
dbus_bool_t _dbus_loop_add_timeout (DBusLoop *loop,
                                   DBusTimeout *timeout);
void _dbus_loop_remove_timeout (DBusLoop *loop,
                                DBusTimeout *timeout);

dbus_bool_t _dbus_loop_queue_dispatch (DBusLoop *loop,
                                       DBusConnection
                                       *connection);

void _dbus_loop_run (DBusLoop *loop);
void _dbus_loop_quit (DBusLoop *loop);
dbus_bool_t _dbus_loop_iterate (DBusLoop *loop,
                               dbus_bool_t block);
dbus_bool_t _dbus_loop_dispatch (DBusLoop *loop);

int _dbus_get_oom_wait (void);
void _dbus_wait_for_memory (void);

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

```

```
#endif /* DBUS_MAINLOOP_H */
```

```
File = dbus-marshal-basic.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* dbus-marshal-basic.c Marshalling routines for basic (primitive)  
types  
*  
* Copyright (C) 2002 CodeFactory AB  
* Copyright (C) 2003, 2004, 2005 Red Hat, Inc.  
*  
* Licensed under the Academic Free License version 2.1  
*  
* This program is free software; you can redistribute it and/or  
modify  
* it under the terms of the GNU General Public License as published  
by  
* the Free Software Foundation; either version 2 of the License, or  
* (at your option) any later version.  
*  
* This program is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program; if not, write to the Free Software  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA  
*  
*/
```

```
#include <config.h>  
#include "dbus-internals.h"  
#include "dbus-marshal-basic.h"  
#include "dbus-signature.h"
```

```
#include <string.h>
```

```
#if defined(__GNUC__) && (__GNUC__ >= 4)  
# define _DBUS_ASSERT_ALIGNMENT(type, op, val) \  
_DBUS_STATIC_ASSERT (__extension__ __alignof__ (type) op val)  
#else  
/* not gcc, so probably no alignof operator: just use a no-op  
statement  
* that's valid in the same contexts */  
# define _DBUS_ASSERT_ALIGNMENT(type, op, val) \  
_DBUS_STATIC_ASSERT (TRUE)  
#endif
```

```

/* True by definition, but just for completeness... */
_DBUS_STATIC_ASSERT (sizeof (char) == 1);
_DBUS_ASSERT_ALIGNMENT (char, ==, 1);

_DBUS_STATIC_ASSERT (sizeof (dbus_int16_t) == 2);
_DBUS_ASSERT_ALIGNMENT (dbus_int16_t, <=, 2);
_DBUS_STATIC_ASSERT (sizeof (dbus_uint16_t) == 2);
_DBUS_ASSERT_ALIGNMENT (dbus_uint16_t, <=, 2);

_DBUS_STATIC_ASSERT (sizeof (dbus_int32_t) == 4);
_DBUS_ASSERT_ALIGNMENT (dbus_int32_t, <=, 4);
_DBUS_STATIC_ASSERT (sizeof (dbus_uint32_t) == 4);
_DBUS_ASSERT_ALIGNMENT (dbus_uint32_t, <=, 4);
_DBUS_STATIC_ASSERT (sizeof (dbus_bool_t) == 4);
_DBUS_ASSERT_ALIGNMENT (dbus_bool_t, <=, 4);

_DBUS_STATIC_ASSERT (sizeof (double) == 8);
_DBUS_ASSERT_ALIGNMENT (double, <=, 8);

#ifdef DBUS_HAVE_INT64
_DBUS_STATIC_ASSERT (sizeof (dbus_int64_t) == 8);
_DBUS_ASSERT_ALIGNMENT (dbus_int64_t, <=, 8);
_DBUS_STATIC_ASSERT (sizeof (dbus_uint64_t) == 8);
_DBUS_ASSERT_ALIGNMENT (dbus_uint64_t, <=, 8);
#endif

_DBUS_STATIC_ASSERT (sizeof (DBusBasicValue) >= 8);
/* The alignment of a DBusBasicValue might conceivably be > 8 because
of the
* pointer, so we don't assert about it */

_DBUS_STATIC_ASSERT (sizeof (DBus8ByteStruct) == 8);
_DBUS_ASSERT_ALIGNMENT (DBus8ByteStruct, <=, 8);

/**
 * @defgroup DBusMarshal marshaling and unmarshaling
 * @ingroup DBusInternals
 * @brief functions to marshal/unmarshal data from the wire
 *
 * Types and functions related to converting primitive data types from
 * wire format to native machine format, and vice versa.
 *
 * A signature is just a string with multiple types one after the
 * other.
 * for example a type is "i" or "(ii)", a signature is "i(ii)"
 * where i is int and (ii) is struct { int; int; }
 *
 * @{
 */

static void

```

```

pack_2_octets (dbus_uint16_t    value,
               int              byte_order,
               unsigned char    *data)
{
    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 2) == data);

    if ((byte_order) == DBUS_LITTLE_ENDIAN)
        *((dbus_uint16_t*)(data)) = DBUS_UINT16_TO_LE (value);
    else
        *((dbus_uint16_t*)(data)) = DBUS_UINT16_TO_BE (value);
}

static void
pack_4_octets (dbus_uint32_t    value,
               int              byte_order,
               unsigned char    *data)
{
    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 4) == data);

    if ((byte_order) == DBUS_LITTLE_ENDIAN)
        *((dbus_uint32_t*)(data)) = DBUS_UINT32_TO_LE (value);
    else
        *((dbus_uint32_t*)(data)) = DBUS_UINT32_TO_BE (value);
}

static void
pack_8_octets (DBusBasicValue    value,
               int              byte_order,
               unsigned char    *data)
{
    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 8) == data);

#ifdef DBUS_HAVE_INT64
    if ((byte_order) == DBUS_LITTLE_ENDIAN)
        *((dbus_uint64_t*)(data)) = DBUS_UINT64_TO_LE (value.u64);
    else
        *((dbus_uint64_t*)(data)) = DBUS_UINT64_TO_BE (value.u64);
#else
    *(DBus8ByteStruct*)data = value.eight;
    swap_8_octets ((DBusBasicValue*)data, byte_order);
#endif
}

/**
 * Packs a 32 bit unsigned integer into a data pointer.
 *
 * @param value the value
 * @param byte_order the byte order to use
 * @param data the data pointer
 */
void
_dbus_pack_uint32 (dbus_uint32_t    value,

```

```

        int            byte_order,
        unsigned char *data)
{
    pack_4_octets (value, byte_order, data);
}

#ifdef DBUS_HAVE_INT64
/* from ORBit */
static void
swap_bytes (unsigned char *data,
            unsigned int  len)
{
    unsigned char *p1 = data;
    unsigned char *p2 = data + len - 1;

    while (p1 < p2)
    {
        unsigned char tmp = *p1;
        *p1 = *p2;
        *p2 = tmp;

        --p2;
        ++p1;
    }
}
#endif /* !DBUS_HAVE_INT64 */

static void
swap_8_octets (DBusBasicValue *value,
               int            byte_order)
{
    if (byte_order != DBUS_COMPILER_BYTE_ORDER)
    {
#ifdef DBUS_HAVE_INT64
        value->u64 = DBUS_UINT64_SWAP_LE_BE (value->u64);
#else
        swap_bytes (&value->bytes, 8);
#endif
    }
}

#ifdef 0
static DBusBasicValue
unpack_8_octets (int            byte_order,
                const unsigned char *data)
{
    DBusBasicValue r;

    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 8) == data);
    _dbus_assert (sizeof (r) == 8);
}
#endif

#ifdef DBUS_HAVE_INT64

```



```

    if (byte_order == DBUS_LITTLE_ENDIAN)
        r.u64 = DBUS_UINT64_FROM_LE (*(dbus_uint64_t*)data);
    else
        r.u64 = DBUS_UINT64_FROM_BE (*(dbus_uint64_t*)data);
#else
    r.eight = *(DBus8ByteStruct*)data;
    swap_8_octets (&r, byte_order);
#endif

    return r;
}
#endif

#ifndef _dbus_unpack_uint16
/**
 * Unpacks a 16 bit unsigned integer from a data pointer
 *
 * @param byte_order The byte order to use
 * @param data the data pointer
 * @returns the integer
 */
dbus_uint16_t
_dbus_unpack_uint16 (int byte_order,
                    const unsigned char *data)
{
    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 2) == data);

    if (byte_order == DBUS_LITTLE_ENDIAN)
        return DBUS_UINT16_FROM_LE (*(dbus_uint16_t*)data);
    else
        return DBUS_UINT16_FROM_BE (*(dbus_uint16_t*)data);
}
#endif /* _dbus_unpack_uint16 */

#ifndef _dbus_unpack_uint32
/**
 * Unpacks a 32 bit unsigned integer from a data pointer
 *
 * @param byte_order The byte order to use
 * @param data the data pointer
 * @returns the integer
 */
dbus_uint32_t
_dbus_unpack_uint32 (int byte_order,
                    const unsigned char *data)
{
    _dbus_assert (_DBUS_ALIGN_ADDRESS (data, 4) == data);

    if (byte_order == DBUS_LITTLE_ENDIAN)
        return DBUS_UINT32_FROM_LE (*(dbus_uint32_t*)data);
    else
        return DBUS_UINT32_FROM_BE (*(dbus_uint32_t*)data);
}

```

```

}
#endif /* _dbus_unpack_uint32 */

static void
set_2_octets (DBusString      *str,
              int              offset,
              dbus_uint16_t    value,
              int              byte_order)
{
    char *data;

    _dbus_assert (byte_order == DBUS_LITTLE_ENDIAN ||
                  byte_order == DBUS_BIG_ENDIAN);

    data = _dbus_string_get_data_len (str, offset, 2);

    pack_2_octets (value, byte_order, data);
}

static void
set_4_octets (DBusString      *str,
              int              offset,
              dbus_uint32_t    value,
              int              byte_order)
{
    char *data;

    _dbus_assert (byte_order == DBUS_LITTLE_ENDIAN ||
                  byte_order == DBUS_BIG_ENDIAN);

    data = _dbus_string_get_data_len (str, offset, 4);

    pack_4_octets (value, byte_order, data);
}

static void
set_8_octets (DBusString      *str,
              int              offset,
              DBusBasicValue    value,
              int              byte_order)
{
    char *data;

    _dbus_assert (byte_order == DBUS_LITTLE_ENDIAN ||
                  byte_order == DBUS_BIG_ENDIAN);

    data = _dbus_string_get_data_len (str, offset, 8);

    pack_8_octets (value, byte_order, data);
}

/**

```

```

* Sets the 4 bytes at the given offset to a marshaled unsigned
* integer, replacing anything found there previously.
*
* @param str the string to write the marshalled int to
* @param pos the byte offset where int should be written
* @param value the value
* @param byte_order the byte order to use
*
*/
void
_dbus_marshal_set_uint32 (DBusString      *str,
                        int               pos,
                        dbus_uint32_t     value,
                        int               byte_order)
{
    set_4_octets (str, pos, value, byte_order);
}

/**
* Sets the existing marshaled string at the given offset with
* a new marshaled string. The given offset must point to
* an existing string or the wrong length will be deleted
* and replaced with the new string.
*
* Note: no attempt is made by this function to re-align
* any data which has been already marshalled after this
* string. Use with caution.
*
* @param str the string to write the marshalled string to
* @param pos the position of the marshaled string length
* @param value the value
* @param byte_order the byte order to use
* @param old_end_pos place to store byte after the nul byte of the
old value
* @param new_end_pos place to store byte after the nul byte of the
new value
* @returns #TRUE on success, #FALSE if no memory
*
*/
static dbus_bool_t
set_string (DBusString      *str,
           int               pos,
           const char       *value,
           int               byte_order,
           int               *old_end_pos,
           int               *new_end_pos)
{
    int old_len, new_len;
    DBusString dstr;

    _dbus_string_init_const (&dstr, value);

```

```

    _dbus_assert (_DBUS_ALIGN_VALUE (pos, 4) == (unsigned) pos);
    old_len = _dbus_unpack_uint32 (byte_order,
                                   _dbus_string_get_const_data_len (str,
pos, 4));

    new_len = _dbus_string_get_length (&dstr);

    if (!_dbus_string_replace_len (&dstr, 0, new_len,
                                   str, pos + 4, old_len))
        return FALSE;

    _dbus_marshal_set_uint32 (str, pos, new_len, byte_order);

    if (old_end_pos)
        *old_end_pos = pos + 4 + old_len + 1;
    if (new_end_pos)
        *new_end_pos = pos + 4 + new_len + 1;

    return TRUE;
}

/**
 * Sets the existing marshaled signature at the given offset to a new
 * marshaled signature. Same basic ideas as set_string().
 *
 * @param str the string to write the marshalled signature to
 * @param pos the position of the marshaled signature length
 * @param value the value
 * @param byte_order the byte order to use
 * @param old_end_pos place to store byte after the nul byte of the
old value
 * @param new_end_pos place to store byte after the nul byte of the
new value
 * @returns #TRUE on success, #FALSE if no memory
 *
 */
static dbus_bool_t
set_signature (DBusString *str,
              int pos,
              const char *value,
              int byte_order,
              int *old_end_pos,
              int *new_end_pos)
{
    int old_len, new_len;
    DBusString dstr;

    _dbus_string_init_const (&dstr, value);

    old_len = _dbus_string_get_byte (str, pos);
    new_len = _dbus_string_get_length (&dstr);

```

```

    if (!_dbus_string_replace_len (&dstr, 0, new_len,
                                   str, pos + 1, old_len))
        return FALSE;

    _dbus_string_set_byte (str, pos, new_len);

    if (old_end_pos)
        *old_end_pos = pos + 1 + old_len + 1;
    if (new_end_pos)
        *new_end_pos = pos + 1 + new_len + 1;

    return TRUE;
}

/**
 * Sets an existing basic type value to a new value.
 * Arguments work the same way as _dbus_marshal_basic_type().
 *
 * @param str the string
 * @param pos location of the current value
 * @param type the type of the current and new values
 * @param value the address of the new value
 * @param byte_order byte order for marshaling
 * @param old_end_pos location to store end position of the old value,
or #NULL
 * @param new_end_pos location to store end position of the new value,
or #NULL
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_marshal_set_basic (DBusString      *str,
                        int               pos,
                        int               type,
                        const void       *value,
                        int               byte_order,
                        int               *old_end_pos,
                        int               *new_end_pos)
{
    const DBusBasicValue *vp;

    vp = value;

    switch (type)
    {
        case DBUS_TYPE_BYTE:
            _dbus_string_set_byte (str, pos, vp->byt);
            if (old_end_pos)
                *old_end_pos = pos + 1;
            if (new_end_pos)
                *new_end_pos = pos + 1;
            return TRUE;
        break;
    }
}

```

```

case DBUS_TYPE_INT16:
case DBUS_TYPE_UINT16:
    pos = _DBUS_ALIGN_VALUE (pos, 2);
    set_2_octets (str, pos, vp->u16, byte_order);
    if (old_end_pos)
        *old_end_pos = pos + 2;
    if (new_end_pos)
        *new_end_pos = pos + 2;
    return TRUE;
    break;
case DBUS_TYPE_BOOLEAN:
case DBUS_TYPE_INT32:
case DBUS_TYPE_UINT32:
case DBUS_TYPE_UNIX_FD:
    pos = _DBUS_ALIGN_VALUE (pos, 4);
    set_4_octets (str, pos, vp->u32, byte_order);
    if (old_end_pos)
        *old_end_pos = pos + 4;
    if (new_end_pos)
        *new_end_pos = pos + 4;
    return TRUE;
    break;
case DBUS_TYPE_INT64:
case DBUS_TYPE_UINT64:
case DBUS_TYPE_DOUBLE:
    pos = _DBUS_ALIGN_VALUE (pos, 8);
    set_8_octets (str, pos, *vp, byte_order);
    if (old_end_pos)
        *old_end_pos = pos + 8;
    if (new_end_pos)
        *new_end_pos = pos + 8;
    return TRUE;
    break;
case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
    pos = _DBUS_ALIGN_VALUE (pos, 4);
    _dbus_assert (vp->str != NULL);
    return set_string (str, pos, vp->str, byte_order,
                      old_end_pos, new_end_pos);

    break;
case DBUS_TYPE_SIGNATURE:
    _dbus_assert (vp->str != NULL);
    return set_signature (str, pos, vp->str, byte_order,
                          old_end_pos, new_end_pos);

    break;
default:
    _dbus_assert_not_reached ("not a basic type");
    return FALSE;
    break;
}
}

```

```

/**
 * Convenience function to demarshal a 32 bit unsigned integer.
 *
 * @param str the string containing the data
 * @param byte_order the byte order
 * @param pos the position in the string
 * @param new_pos the new position of the string
 * @returns the demarshaled integer.
 */
dbus_uint32_t
_dbus_marshal_read_uint32 (const DBusString *str,
                          int pos,
                          int byte_order,
                          int *new_pos)
{
    pos = _DBUS_ALIGN_VALUE (pos, 4);

    if (new_pos)
        *new_pos = pos + 4;

    _dbus_assert (pos + 4 <= _dbus_string_get_length (str));

    return _dbus_unpack_uint32 (byte_order,
                                _dbus_string_get_const_data (str) +
pos);
}

/**
 * Demarshals a basic-typed value. The "value" pointer is always
 * the address of a variable of the basic type. So e.g.
 * if the basic type is "double" then the pointer is
 * a double*, and if it's "char*" then the pointer is
 * a "char**".
 *
 * A value of type #DBusBasicValue is guaranteed to be large enough to
 * hold any of the types that may be returned, which is handy if you
 * are trying to do things generically. For example you can pass
 * a DBusBasicValue* in to this function, and then pass the same
 * DBusBasicValue* in to _dbus_marshal_basic_type() in order to
 * move a value from one place to another.
 *
 * @param str the string containing the data
 * @param pos position in the string
 * @param type type of value to demarshal
 * @param value pointer to return value data
 * @param byte_order the byte order
 * @param new_pos pointer to update with new position, or #NULL
 */
void
_dbus_marshal_read_basic (const DBusString *str,
                        int pos,
                        int type,

```

```

                                void                *value,
                                int                 byte_order,
                                int                 *new_pos)
{
    const char *str_data;

    _dbus_assert (dbus_type_is_basic (type));

    str_data = _dbus_string_get_const_data (str);

    /* Below we volatile types to avoid aliasing issues;
     * see http://bugs.freedesktop.org/show\_bug.cgi?id=20137
     */

    switch (type)
    {
        case DBUS_TYPE_BYTE:
        {
            volatile unsigned char *vp = value;
            *vp = (unsigned char) _dbus_string_get_byte (str, pos);
            (pos)++;
        }
        break;
        case DBUS_TYPE_INT16:
        case DBUS_TYPE_UINT16:
        {
            volatile dbus_uint16_t *vp = value;
            pos = _DBUS_ALIGN_VALUE (pos, 2);
            *vp = *(dbus_uint16_t *) (str_data + pos);
            if (byte_order != DBUS_COMPILER_BYTE_ORDER)
                *vp = DBUS_UINT16_SWAP_LE_BE (*vp);
            pos += 2;
        }
        break;
        case DBUS_TYPE_INT32:
        case DBUS_TYPE_UINT32:
        case DBUS_TYPE_BOOLEAN:
        case DBUS_TYPE_UNIX_FD:
        {
            volatile dbus_uint32_t *vp = value;
            pos = _DBUS_ALIGN_VALUE (pos, 4);
            *vp = *(dbus_uint32_t *) (str_data + pos);
            if (byte_order != DBUS_COMPILER_BYTE_ORDER)
                *vp = DBUS_UINT32_SWAP_LE_BE (*vp);
            pos += 4;
        }
        break;
        case DBUS_TYPE_INT64:
        case DBUS_TYPE_UINT64:
        case DBUS_TYPE_DOUBLE:
        {
            volatile dbus_uint64_t *vp = value;

```



```

        pos = _DBUS_ALIGN_VALUE (pos, 8);
#ifdef DBUS_HAVE_INT64
        if (byte_order != DBUS_COMPILER_BYTE_ORDER)
            *vp = DBUS_UINT64_SWAP_LE_BE (*(dbus_uint64_t*) (str_data +
pos));
        else
            *vp = *(dbus_uint64_t*) (str_data + pos);
#else
        *vp = *(DBus8ByteStruct*) (str_data + pos);
        swap_8_octets (vp, byte_order);
#endif
        pos += 8;
    }
    break;
case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
    {
        int len;
        volatile char **vp = value;

        len = _dbus_marshal_read_uint32 (str, pos, byte_order, &pos);

        *vp = (char*) str_data + pos;

        pos += len + 1; /* length plus nul */
    }
    break;
case DBUS_TYPE_SIGNATURE:
    {
        int len;
        volatile char **vp = value;

        len = _dbus_string_get_byte (str, pos);
        pos += 1;

        *vp = (char*) str_data + pos;

        pos += len + 1; /* length plus nul */
    }
    break;
default:
    _dbus_warn_check_failed ("type %s %d not a basic type\n",
                            _dbus_type_to_string (type), type);
    _dbus_assert_not_reached ("not a basic type");
    break;
}

if (new_pos)
    *new_pos = pos;
}

static dbus_bool_t

```

```

marshal_2_octets (DBusString  *str,
                  int          insert_at,
                  dbus_uint16_t value,
                  int          byte_order,
                  int          *pos_after)
{
    dbus_bool_t retval;
    int orig_len;

    _dbus_assert (sizeof (value) == 2);

    if (byte_order != DBUS_COMPILER_BYTE_ORDER)
        value = DBUS_UINT16_SWAP_LE_BE (value);

    orig_len = _dbus_string_get_length (str);

    retval = _dbus_string_insert_2_aligned (str, insert_at,
                                           (const unsigned char
*)&value);

    if (pos_after)
    {
        *pos_after = insert_at + (_dbus_string_get_length (str) -
orig_len);
        _dbus_assert (*pos_after <= _dbus_string_get_length (str));
    }

    return retval;
}

static dbus_bool_t
marshal_4_octets (DBusString  *str,
                  int          insert_at,
                  dbus_uint32_t value,
                  int          byte_order,
                  int          *pos_after)
{
    dbus_bool_t retval;
    int orig_len;

    _dbus_assert (sizeof (value) == 4);

    if (byte_order != DBUS_COMPILER_BYTE_ORDER)
        value = DBUS_UINT32_SWAP_LE_BE (value);

    orig_len = _dbus_string_get_length (str);

    retval = _dbus_string_insert_4_aligned (str, insert_at,
                                           (const unsigned char
*)&value);

    if (pos_after)

```

```

    {
        *pos_after = insert_at + (_dbus_string_get_length (str) -
orig_len);
        _dbus_assert (*pos_after <= _dbus_string_get_length (str));
    }

    return retval;
}

static dbus_bool_t
marshal_8_octets (DBusString *str,
                 int insert_at,
                 DBusBasicValue value,
                 int byte_order,
                 int *pos_after)
{
    dbus_bool_t retval;
    int orig_len;

    _dbus_assert (sizeof (value) == 8);

    swap_8_octets (&value, byte_order);

    orig_len = _dbus_string_get_length (str);

    retval = _dbus_string_insert_8_aligned (str, insert_at,
                                           (const unsigned char
*)&value);

    if (pos_after)
        *pos_after = insert_at + _dbus_string_get_length (str) - orig_len;

    return retval;
}

enum
{
    MARSHAL_AS_STRING,
    MARSHAL_AS_SIGNATURE,
    MARSHAL_AS_BYTE_ARRAY
};

static dbus_bool_t
marshal_len_followed_by_bytes (int marshal_as,
                               DBusString *str,
                               int insert_at,
                               const unsigned char *value,
                               int data_len, /*
doesn't include nul if any */
                               int byte_order,
                               int *pos_after)
{

```

```

int pos;
DBusString value_str;
int value_len;

_dbus_assert (byte_order == DBUS_LITTLE_ENDIAN || byte_order ==
DBus_BIG_ENDIAN);
if (insert_at > _dbus_string_get_length (str))
    _dbus_warn ("insert_at = %d string len = %d data_len = %d\n",
                insert_at, _dbus_string_get_length (str), data_len);

if (marshal_as == MARSHAL_AS_BYTE_ARRAY)
    value_len = data_len;
else
    value_len = data_len + 1; /* value has a nul */

_dbus_string_init_const_len (&value_str, value, value_len);

pos = insert_at;

if (marshal_as == MARSHAL_AS_SIGNATURE)
{
    _dbus_assert (data_len <= DBUS_MAXIMUM_SIGNATURE_LENGTH);
    _dbus_assert (data_len <= 255); /* same as max sig len right now
*/

    if (!_dbus_string_insert_byte (str, pos, data_len))
        goto oom;

    pos += 1;
}
else
{
    if (!marshal_4_octets (str, pos, data_len,
                          byte_order, &pos))
        goto oom;
}

if (!_dbus_string_copy_len (&value_str, 0, value_len,
                           str, pos))
    goto oom;

#if 0
/* too expensive */
_dbus_assert (_dbus_string_equal_substring (&value_str, 0,
value_len,
                                           str, pos));
_dbus_verbose_bytes_of_string (str, pos, value_len);
#endif

pos += value_len;

if (pos_after)

```

```

    *pos_after = pos;

return TRUE;

oom:
/* Delete what we've inserted */
_dbus_string_delete (str, insert_at, pos - insert_at);

return FALSE;
}

static dbus_bool_t
marshal_string (DBusString      *str,
                int              insert_at,
                const char      *value,
                int              byte_order,
                int              *pos_after)
{
    return marshal_len_followed_by_bytes (MARSHAL_AS_STRING,
                                         str, insert_at, value,
                                         strlen (value),
                                         byte_order, pos_after);
}

static dbus_bool_t
marshal_signature (DBusString      *str,
                  int              insert_at,
                  const char      *value,
                  int              *pos_after)
{
    return marshal_len_followed_by_bytes (MARSHAL_AS_SIGNATURE,
                                         str, insert_at, value,
                                         strlen (value),
                                         DBUS_COMPILER_BYTE_ORDER, /*
irrelevant */
                                         pos_after);
}

/**
 * Marshals a basic-typed value. The "value" pointer is always the
 * address of a variable containing the basic type value.
 * So for example for int32 it will be dbus_int32_t*, and
 * for string it will be const char**. This is for symmetry
 * with _dbus_marshal_read_basic() and to have a simple
 * consistent rule.
 *
 * @param str string to marshal to
 * @param insert_at where to insert the value
 * @param type type of value
 * @param value pointer to a variable containing the value
 * @param byte_order byte order
 * @param pos_after #NULL or the position after the type

```



```

        break;
    case DBUS_TYPE_SIGNATURE:
        _dbus_assert (vp->str != NULL);
        return marshal_signature (str, insert_at, vp->str, pos_after);
        break;
    default:
        _dbus_assert_not_reached ("not a basic type");
        return FALSE;
        break;
    }
}

```

```

static dbus_bool_t
marshal_1_octets_array (DBusString *str,
                       int insert_at,
                       const unsigned char *value,
                       int n_elements,
                       int byte_order,
                       int *pos_after)
{
    int pos;
    DBusString value_str;

    _dbus_string_init_const_len (&value_str, value, n_elements);

    pos = insert_at;

    if (!_dbus_string_copy_len (&value_str, 0, n_elements,
                               str, pos))
        return FALSE;

    pos += n_elements;

    if (pos_after)
        *pos_after = pos;

    return TRUE;
}

```

```

/**
 * Swaps the elements of an array to the opposite byte order
 *
 * @param data start of array
 * @param n_elements number of elements
 * @param alignment size of each element
 */
void
_dbus_swap_array (unsigned char *data,
                 int n_elements,
                 int alignment)
{
    unsigned char *d;

```

```

unsigned char *end;

_dbus_assert (_DBUS_ALIGN_ADDRESS (data, alignment) == data);

/* we use const_data and cast it off so DBusString can be a const
string
 * for the unit tests. don't ask.
 */
d = data;
end = d + (n_elements * alignment);

if (alignment == 8)
{
    while (d != end)
    {
#ifdef DBUS_HAVE_INT64
        *((dbus_uint64_t*)d) = DBUS_UINT64_SWAP_LE_BE
(*((dbus_uint64_t*)d));
#else
        swap_8_bytes ((DBusBasicValue*) d);
#endif
        d += 8;
    }
}
else if (alignment == 4)
{
    while (d != end)
    {
        *((dbus_uint32_t*)d) = DBUS_UINT32_SWAP_LE_BE
(*((dbus_uint32_t*)d));
        d += 4;
    }
}
else
{
    _dbus_assert (alignment == 2);

    while (d != end)
    {
        *((dbus_uint16_t*)d) = DBUS_UINT16_SWAP_LE_BE
(*((dbus_uint16_t*)d));
        d += 2;
    }
}
}

static void
swap_array (DBusString *str,
            int         array_start,
            int         n_elements,
            int         byte_order,
            int         alignment)

```



```

{
    _dbus_assert (_DBUS_ALIGN_VALUE (array_start, alignment) ==
(unsigned) array_start);

    if (byte_order != DBUS_COMPILER_BYTE_ORDER)
    {
        /* we use const_data and cast it off so DbusString can be a
const string
        * for the unit tests. don't ask.
        */
        _dbus_swap_array ((unsigned char*) (_dbus_string_get_const_data
(str) + array_start),
                        n_elements, alignment);
    }
}

static dbus_bool_t
marshal_fixed_multi (DBusString      *str,
                    int              insert_at,
                    const DBusBasicValue *value,
                    int              n_elements,
                    int              byte_order,
                    int              alignment,
                    int              *pos_after)
{
    int old_string_len;
    int array_start;
    DBusString t;
    int len_in_bytes;

    _dbus_assert (n_elements <= DBUS_MAXIMUM_ARRAY_LENGTH / alignment);

    old_string_len = _dbus_string_get_length (str);

    len_in_bytes = n_elements * alignment;
    array_start = insert_at;

    /* Note that we do alignment padding unconditionally
    * even if the array is empty; this means that
    * padding + len is always equal to the number of bytes
    * in the array.
    */

    if (!_dbus_string_insert_alignment (str, &array_start, alignment))
        goto error;

    _dbus_string_init_const_len (&t,
                                (const unsigned char*) value,
                                len_in_bytes);

    if (!_dbus_string_copy (&t, 0,
                          str, array_start))

```

```

        goto error;

    swap_array (str, array_start, n_elements, byte_order, alignment);

    if (pos_after)
        *pos_after = array_start + len_in_bytes;

    return TRUE;

error:
    _dbus_string_delete (str, insert_at,
                        _dbus_string_get_length (str) -
old_string_len);

    return FALSE;
}

/**
 * Marshals a block of values of fixed-length type all at once, as an
 * optimization.  dbus_type_is_fixed() returns #TRUE for fixed-length
 * types, which are the basic types minus the string-like types.
 *
 * The value argument should be the address of an
 * array, so e.g. "const dbus_uint32_t**"
 *
 * @param str string to marshal to
 * @param insert_at where to insert the value
 * @param element_type type of array elements
 * @param value address of an array to marshal
 * @param n_elements number of elements in the array
 * @param byte_order byte order
 * @param pos_after #NULL or the position after the type
 * @returns #TRUE on success
 */
dbus_bool_t
_dbus_marshal_write_fixed_multi (DBusString *str,
                                int          insert_at,
                                int          element_type,
                                const void *value,
                                int          n_elements,
                                int          byte_order,
                                int          *pos_after)
{
    const void* vp = *(const DBusBasicValue**)value;

    _dbus_assert (dbus_type_is_fixed (element_type));
    _dbus_assert (n_elements >= 0);

#ifdef DBUS_VERBOSE
    _dbus_verbose ("writing %d elements of %s\n",
                  n_elements, _dbus_type_to_string (element_type));
#endif
}

```

```

switch (element_type)
{
    case DBUS_TYPE_BYTE:
        return marshal_1_octets_array (str, insert_at, vp, n_elements,
byte_order, pos_after);
        break;
    case DBUS_TYPE_INT16:
    case DBUS_TYPE_UINT16:
        return marshal_fixed_multi (str, insert_at, vp, n_elements,
byte_order, 2, pos_after);
    case DBUS_TYPE_BOOLEAN:
    case DBUS_TYPE_INT32:
    case DBUS_TYPE_UINT32:
    case DBUS_TYPE_UNIX_FD:
        return marshal_fixed_multi (str, insert_at, vp, n_elements,
byte_order, 4, pos_after);
        break;
    case DBUS_TYPE_INT64:
    case DBUS_TYPE_UINT64:
    case DBUS_TYPE_DOUBLE:
        return marshal_fixed_multi (str, insert_at, vp, n_elements,
byte_order, 8, pos_after);
        break;

    default:
        _dbus_assert_not_reached ("non fixed type in array write");
        break;
}

return FALSE;
}

```

```

/**
 * Skips over a basic-typed value, reporting the following position.
 *
 * @param str the string containing the data
 * @param type type of value to read
 * @param byte_order the byte order
 * @param pos pointer to position in the string,
 *             updated on return to new position
 */
void
_dbus_marshal_skip_basic (const DBusString      *str,
                          int                  type,
                          int                  byte_order,
                          int                  *pos)
{
    _dbus_assert (byte_order == DBUS_LITTLE_ENDIAN ||
byte_order == DBUS_BIG_ENDIAN);
}

```

```

switch (type)
{
case DBUS_TYPE_BYTE:
    (*pos)++;
    break;
case DBUS_TYPE_INT16:
case DBUS_TYPE_UINT16:
    *pos = _DBUS_ALIGN_VALUE (*pos, 2);
    *pos += 2;
    break;
case DBUS_TYPE_BOOLEAN:
case DBUS_TYPE_INT32:
case DBUS_TYPE_UINT32:
case DBUS_TYPE_UNIX_FD:
    *pos = _DBUS_ALIGN_VALUE (*pos, 4);
    *pos += 4;
    break;
case DBUS_TYPE_INT64:
case DBUS_TYPE_UINT64:
case DBUS_TYPE_DOUBLE:
    *pos = _DBUS_ALIGN_VALUE (*pos, 8);
    *pos += 8;
    break;
case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
    {
        int len;

        len = _dbus_marshal_read_uint32 (str, *pos, byte_order, pos);

        *pos += len + 1; /* length plus nul */
    }
    break;
case DBUS_TYPE_SIGNATURE:
    {
        int len;

        len = _dbus_string_get_byte (str, *pos);

        *pos += len + 2; /* length byte plus length plus nul */
    }
    break;
default:
    _dbus_warn ("type %s not a basic type\n",
                _dbus_type_to_string (type));
    _dbus_assert_not_reached ("not a basic type");
    break;
}
}

/**
 * Skips an array, returning the next position.

```

```

*
* @param str the string containing the data
* @param element_type the type of array elements
* @param byte_order the byte order
* @param pos pointer to position in the string,
*             updated on return to new position
*/
void
_dbus_marshal_skip_array (const DBusString *str,
                          int             element_type,
                          int             byte_order,
                          int             *pos)
{
    dbus_uint32_t array_len;
    int i;
    int alignment;

    i = _DBUS_ALIGN_VALUE (*pos, 4);

    array_len = _dbus_marshal_read_uint32 (str, i, byte_order, &i);

    alignment = _dbus_type_get_alignment (element_type);

    i = _DBUS_ALIGN_VALUE (i, alignment);

    *pos = i + array_len;
}

/**
 * Gets the alignment requirement for the given type;
 * will be 1, 4, or 8.
 *
 * @param typecode the type
 * @returns alignment of 1, 4, or 8
 */
int
_dbus_type_get_alignment (int typecode)
{
    switch (typecode)
    {
        {
            case DBUS_TYPE_BYTE:
            case DBUS_TYPE_VARIANT:
            case DBUS_TYPE_SIGNATURE:
                return 1;
            case DBUS_TYPE_INT16:
            case DBUS_TYPE_UINT16:
                return 2;
            case DBUS_TYPE_BOOLEAN:
            case DBUS_TYPE_INT32:
            case DBUS_TYPE_UINT32:
            case DBUS_TYPE_UNIX_FD:
                /* this stuff is 4 since it starts with a length */

```

```

case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
case DBUS_TYPE_ARRAY:
    return 4;
case DBUS_TYPE_INT64:
case DBUS_TYPE_UINT64:
case DBUS_TYPE_DOUBLE:
    /* struct is 8 since it could contain an 8-aligned item
     * and it's simpler to just always align structs to 8;
     * we want the amount of padding in a struct of a given
     * type to be predictable, not location-dependent.
     * DICT_ENTRY is always the same as struct.
     */
case DBUS_TYPE_STRUCT:
case DBUS_TYPE_DICT_ENTRY:
    return 8;

default:
    _dbus_assert_not_reached ("unknown typecode in
_dbus_type_get_alignment()");
    return 0;
}
}

/**
 * Returns a string describing the given type.
 *
 * @param typecode the type to describe
 * @returns a constant string describing the type
 */
const char *
_dbus_type_to_string (int typecode)
{
    switch (typecode)
    {
        case DBUS_TYPE_INVALID:
            return "invalid";
        case DBUS_TYPE_BOOLEAN:
            return "boolean";
        case DBUS_TYPE_BYTE:
            return "byte";
        case DBUS_TYPE_INT16:
            return "int16";
        case DBUS_TYPE_UINT16:
            return "uint16";
        case DBUS_TYPE_INT32:
            return "int32";
        case DBUS_TYPE_UINT32:
            return "uint32";
        case DBUS_TYPE_INT64:
            return "int64";
        case DBUS_TYPE_UINT64:

```

```

        return "uint64";
    case DBUS_TYPE_DOUBLE:
        return "double";
    case DBUS_TYPE_STRING:
        return "string";
    case DBUS_TYPE_OBJECT_PATH:
        return "object_path";
    case DBUS_TYPE_SIGNATURE:
        return "signature";
    case DBUS_TYPE_STRUCT:
        return "struct";
    case DBUS_TYPE_DICT_ENTRY:
        return "dict_entry";
    case DBUS_TYPE_ARRAY:
        return "array";
    case DBUS_TYPE_VARIANT:
        return "variant";
    case DBUS_STRUCT_BEGIN_CHAR:
        return "begin_struct";
    case DBUS_STRUCT_END_CHAR:
        return "end_struct";
    case DBUS_DICT_ENTRY_BEGIN_CHAR:
        return "begin_dict_entry";
    case DBUS_DICT_ENTRY_END_CHAR:
        return "end_dict_entry";
    case DBUS_TYPE_UNIX_FD:
        return "unix_fd";
    default:
        return "unknown";
    }
}

/**
 * If in verbose mode, print a block of binary data.
 *
 * @param data the data
 * @param len the length of the data
 * @param offset where to start counting for byte indexes
 */
void
_dbus_verbose_bytes (const unsigned char *data,
                    int len,
                    int offset)
{
    int i;
    const unsigned char *aligned;

    _dbus_assert (len >= 0);

    if (!_dbus_is_verbose())
        return;

```

```

/* Print blanks on first row if appropriate */
aligned = _DBUS_ALIGN_ADDRESS (data, 4);
if (aligned > data)
    aligned -= 4;
_dbus_assert (aligned <= data);

if (aligned != data)
{
    _dbus_verbose ("%4ld\t%p: ", - (long)(data - aligned), aligned);
    while (aligned != data)
    {
        _dbus_verbose ("    ");
        ++aligned;
    }
}

/* now print the bytes */
i = 0;
while (i < len)
{
    if (_DBUS_ALIGN_ADDRESS (&data[i], 4) == &data[i])
    {
        _dbus_verbose ("%4d\t%p: ",
            offset + i, &data[i]);
    }

    if (data[i] >= 32 &&
        data[i] <= 126)
        _dbus_verbose (" '%c' ", data[i]);
    else
        _dbus_verbose ("0x%s%x ",
            data[i] <= 0xf ? "0" : "", data[i]);

    ++i;

    if (_DBUS_ALIGN_ADDRESS (&data[i], 4) == &data[i])
    {
        if (i > 3)
            _dbus_verbose ("BE: %d LE: %d",
                _dbus_unpack_uint32 (DBUS_BIG_ENDIAN,
&data[i-4]),
                _dbus_unpack_uint32 (DBUS_LITTLE_ENDIAN,
&data[i-4]));

        if (i > 7 &&
            _DBUS_ALIGN_ADDRESS (&data[i], 8) == &data[i])
        {
#ifdef DBUS_INT64_PRINTF_MODIFIER
            _dbus_verbose (" u64: 0x%" DBUS_INT64_PRINTF_MODIFIER
"x",
                *(dbus_uint64_t*)&data[i-8]);
#endif
        }
    }
}

```



```

        _dbus_verbose (" dbl: %g",
                       *(double*)&data[i-8]);
    }

    _dbus_verbose ("\n");
}

_dbus_verbose ("\n");
}

/**
 * Dump the given part of the string to verbose log.
 *
 * @param str the string
 * @param start the start of range to dump
 * @param len length of range
 */
void
_dbus_verbose_bytes_of_string (const DBusString *str,
                              int start,
                              int len)
{
    const char *d;
    int real_len;

    real_len = _dbus_string_get_length (str);

    _dbus_assert (start >= 0);

    if (start > real_len)
    {
        _dbus_verbose (" [%d,%d] is not inside string of length %d\n",
                       start, len, real_len);
        return;
    }

    if ((start + len) > real_len)
    {
        _dbus_verbose (" [%d,%d] extends outside string of length
%d\n",
                       start, len, real_len);
        len = real_len - start;
    }

    d = _dbus_string_get_const_data_len (str, start, len);

    _dbus_verbose_bytes (d, len, start);
}

static int
map_type_char_to_type (int t)

```

```

{
    if (t == DBUS_STRUCT_BEGIN_CHAR)
        return DBUS_TYPE_STRUCT;
    else if (t == DBUS_DICT_ENTRY_BEGIN_CHAR)
        return DBUS_TYPE_DICT_ENTRY;
    else
        {
            _dbus_assert (t != DBUS_STRUCT_END_CHAR);
            _dbus_assert (t != DBUS_DICT_ENTRY_END_CHAR);
            return t;
        }
}

/**
 * Get the first type in the signature. The difference between this
 * and just getting the first byte of the signature is that you won't
 * get DBUS_STRUCT_BEGIN_CHAR, you'll get DBUS_TYPE_STRUCT
 * instead.
 *
 * @param str string containing signature
 * @param pos where the signature starts
 * @returns the first type in the signature
 */
int
_dbus_first_type_in_signature (const DBusString *str,
                              int pos)
{
    return map_type_char_to_type (_dbus_string_get_byte (str, pos));
}

/**
 * Similar to #_dbus_first_type_in_signature, but operates
 * on a C string buffer.
 *
 * @param str a C string buffer
 * @param pos where the signature starts
 * @returns the first type in the signature
 */
int
_dbus_first_type_in_signature_c_str (const char *str,
                                     int pos)
{
    return map_type_char_to_type (str[pos]);
}

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

/**

```

```

* Reads a block of fixed-length basic values, as an optimization
* vs. reading each one individually into a new buffer.
*
* This function returns the data in-place; it does not make a copy,
* and it does not swap the bytes.
*
* If you ask for #DBUS_TYPE_DOUBLE you will get a "const double*"
back
* and the "value" argument should be a "const double**" and so on.
*
* @param str the string to read from
* @param pos position to read from
* @param element_type type of array elements
* @param value place to return the array
* @param n_elements number of array elements to read
* @param byte_order the byte order, used to read the array length
* @param new_pos #NULL or location to store a position after the
elements
*/
void
_dbus_marshal_read_fixed_multi (const DBusString *str,
                                int                pos,
                                int                element_type,
                                void              *value,
                                int                n_elements,
                                int                byte_order,
                                int                new_pos)
{
    int array_len;
    int alignment;

    _dbus_assert (dbus_type_is_fixed (element_type));
    _dbus_assert (dbus_type_is_basic (element_type));

#ifdef 0
    _dbus_verbose ("reading %d elements of %s\n",
                  n_elements, _dbus_type_to_string (element_type));
#endif

    alignment = _dbus_type_get_alignment (element_type);

    pos = _DBUS_ALIGN_VALUE (pos, alignment);

    array_len = n_elements * alignment;

    *(const DBusBasicValue**) value = (void*)
_dbus_string_get_const_data_len (str, pos, array_len);
    if (new_pos)
        *new_pos = pos + array_len;
}

static void

```



```

                                _dbus_string_get_length (&str) -
dump_pos); \
    _dbus_assert_not_reached ("demarshaled wrong value");
\
    }
\
} while (0)

#define MARSHAL_TEST(typename, byte_order, literal) \
do { \
    MARSHAL_BASIC (typename, byte_order, literal); \
    dump_pos = pos; \
    DEMARSHAL_BASIC_AND_CHECK (typename, byte_order, literal); \
} while (0)

#define MARSHAL_TEST_STRCMP(typename, byte_order, literal)
\
do {
\
    MARSHAL_BASIC (typename, byte_order, literal);
\
    dump_pos = pos;
\
    DEMARSHAL_BASIC (typename, byte_order);
\
    if (strcmp (literal, v_##typename) != 0)
\
        {
\
            _dbus_verbose_bytes_of_string (&str, dump_pos,
\
                                _dbus_string_get_length (&str)
- dump_pos); \
            _dbus_warn ("literal '%s'\nvalue  '%s'\n", literal,
v_##typename); \
            _dbus_assert_not_reached ("demarshaled wrong value");
\
        }
\
} while (0)

#define MARSHAL_FIXED_ARRAY(typename, byte_order, literal)
\
do {
\
    int next;
\
    v_UINT32 = sizeof(literal);
\
    if (!_dbus_marshal_write_basic (&str, pos, DBUS_TYPE_UINT32,
&v_UINT32, \

```

```

        byte_order, &next))
\
    _dbus_assert_not_reached ("no memory");
\
    v_ARRAY_##typename = literal;
\
    if (!_dbus_marshal_write_fixed_multi (&str, next,
DBUS_TYPE_##typename, \
        &v_ARRAY_##typename,
_DBUS_N_ELEMENTS(literal), \
        byte_order, NULL))
\
    _dbus_assert_not_reached ("no memory");
\
} while (0)

#define DEMARSHAL_FIXED_ARRAY(typename, byte_order)
\
do {
\
    int next;
\
    alignment = _dbus_type_get_alignment (DBUS_TYPE_##typename);
\
    v_UINT32 = _dbus_marshal_read_uint32 (&str, dump_pos, byte_order,
&next);
\
    _dbus_marshal_read_fixed_multi (&str, next, DBUS_TYPE_##typename,
&v_ARRAY_##typename, \
        v_UINT32/alignment,
\
        byte_order, NULL);
\
    swap_test_array (v_ARRAY_##typename, v_UINT32,
\
        byte_order, alignment);
\
} while (0)

#define DEMARSHAL_FIXED_ARRAY_AND_CHECK(typename, byte_order, literal)
\
do {
\
    DEMARSHAL_FIXED_ARRAY (typename, byte_order);
\
    if (memcmp (literal, v_ARRAY_##typename, sizeof (literal) != 0))
\
    {
\
        _dbus_verbose ("MARSHALED DATA\n");
\
        _dbus_verbose_bytes_of_string (&str, dump_pos,
\

```

```

                                _dbus_string_get_length (&str) -
dump_pos); \
    _dbus_verbose ("LITERAL DATA\n");
\
    _dbus_verbose_bytes ((char*)literal, sizeof (literal), 0);
\
    _dbus_verbose ("READ DATA\n");
\
    _dbus_verbose_bytes ((char*)v_ARRAY_##typename, sizeof
(literal), 0); \
    _dbus_assert_not_reached ("demarshaled wrong fixed array
value"); \
    }
\
} while (0)

#define MARSHAL_TEST_FIXED_ARRAY(typename, byte_order, literal)
\
do {
\
    MARSHAL_FIXED_ARRAY (typename, byte_order, literal);
\
    dump_pos = pos;
\
    DEMARSHAL_FIXED_ARRAY_AND_CHECK (typename, byte_order, literal);
\
} while (0)

dbus_bool_t
_dbus_marshal_test (void)
{
    int alignment;
    DBusString str;
    int pos, dump_pos;
    unsigned char array1[5] = { 3, 4, 0, 1, 9 };
    dbus_int16_t array2[3] = { 124, 457, 780 };
    dbus_int32_t array4[3] = { 123, 456, 789 };
#ifdef DBUS_HAVE_INT64
    dbus_int64_t array8[3] = { DBUS_INT64_CONSTANT (0x123ffffffff),
                                DBUS_INT64_CONSTANT (0x456ffffffff),
                                DBUS_INT64_CONSTANT (0x789ffffffff) };
    dbus_int64_t *v_ARRAY_INT64;
#endif
    unsigned char *v_ARRAY_BYTE;
    dbus_int16_t *v_ARRAY_INT16;
    dbus_uint16_t *v_ARRAY_UINT16;
    dbus_int32_t *v_ARRAY_INT32;
    dbus_uint32_t *v_ARRAY_UINT32;
    DBusString t;
    double v_DOUBLE;
    double t_DOUBLE;
    dbus_int16_t v_INT16;

```

```

dbus_uint16_t v_UINT16;
dbus_int32_t v_INT32;
dbus_uint32_t v_UINT32;
dbus_int64_t v_INT64;
dbus_uint64_t v_UINT64;
unsigned char v_BYTE;
dbus_bool_t v_BOOLEAN;
const char *v_STRING;
const char *v_SIGNATURE;
const char *v_OBJECT_PATH;
int byte_order;

if (!_dbus_string_init (&str))
    _dbus_assert_not_reached ("failed to init string");

pos = 0;

/* Marshal doubles */
MARSHAL_BASIC (DOUBLE, DBUS_BIG_ENDIAN, 3.14);
DEMARSHAL_BASIC (DOUBLE, DBUS_BIG_ENDIAN);
t_DOUBLE = 3.14;
if (!_DBUS_DOUBLES_BITWISE_EQUAL (t_DOUBLE, v_DOUBLE))
    _dbus_assert_not_reached ("got wrong double value");

MARSHAL_BASIC (DOUBLE, DBUS_LITTLE_ENDIAN, 3.14);
DEMARSHAL_BASIC (DOUBLE, DBUS_LITTLE_ENDIAN);
t_DOUBLE = 3.14;
if (!_DBUS_DOUBLES_BITWISE_EQUAL (t_DOUBLE, v_DOUBLE))
    _dbus_assert_not_reached ("got wrong double value");

/* Marshal signed 16 integers */
MARSHAL_TEST (INT16, DBUS_BIG_ENDIAN, -12345);
MARSHAL_TEST (INT16, DBUS_LITTLE_ENDIAN, -12345);

/* Marshal unsigned 16 integers */
MARSHAL_TEST (UINT16, DBUS_BIG_ENDIAN, 0x1234);
MARSHAL_TEST (UINT16, DBUS_LITTLE_ENDIAN, 0x1234);

/* Marshal signed integers */
MARSHAL_TEST (INT32, DBUS_BIG_ENDIAN, -12345678);
MARSHAL_TEST (INT32, DBUS_LITTLE_ENDIAN, -12345678);

/* Marshal unsigned integers */
MARSHAL_TEST (UINT32, DBUS_BIG_ENDIAN, 0x12345678);
MARSHAL_TEST (UINT32, DBUS_LITTLE_ENDIAN, 0x12345678);

#ifdef DBUS_HAVE_INT64
/* Marshal signed integers */
MARSHAL_TEST (INT64, DBUS_BIG_ENDIAN, DBUS_INT64_CONSTANT (-
0x123456789abc7));
MARSHAL_TEST (INT64, DBUS_LITTLE_ENDIAN, DBUS_INT64_CONSTANT (-
0x123456789abc7));

```



```

/* Marshal unsigned integers */
MARSHAL_TEST (UINT64, DBUS_BIG_ENDIAN, DBUS_UINT64_CONSTANT
(0x123456789abc7));
MARSHAL_TEST (UINT64, DBUS_LITTLE_ENDIAN, DBUS_UINT64_CONSTANT
(0x123456789abc7));
#endif /* DBUS_HAVE_INT64 */

/* Marshal byte */
MARSHAL_TEST (BYTE, DBUS_BIG_ENDIAN, 5);
MARSHAL_TEST (BYTE, DBUS_LITTLE_ENDIAN, 5);

/* Marshal all possible bools! */
MARSHAL_TEST (BOOLEAN, DBUS_BIG_ENDIAN, FALSE);
MARSHAL_TEST (BOOLEAN, DBUS_LITTLE_ENDIAN, FALSE);
MARSHAL_TEST (BOOLEAN, DBUS_BIG_ENDIAN, TRUE);
MARSHAL_TEST (BOOLEAN, DBUS_LITTLE_ENDIAN, TRUE);

/* Marshal strings */
MARSHAL_TEST_STRCMP (STRING, DBUS_BIG_ENDIAN, "");
MARSHAL_TEST_STRCMP (STRING, DBUS_LITTLE_ENDIAN, "");
MARSHAL_TEST_STRCMP (STRING, DBUS_BIG_ENDIAN, "This is the dbus test
string");
MARSHAL_TEST_STRCMP (STRING, DBUS_LITTLE_ENDIAN, "This is the dbus
test string");

/* object paths */
MARSHAL_TEST_STRCMP (OBJECT_PATH, DBUS_BIG_ENDIAN, "/a/b/c");
MARSHAL_TEST_STRCMP (OBJECT_PATH, DBUS_LITTLE_ENDIAN, "/a/b/c");

/* signatures */
MARSHAL_TEST_STRCMP (SIGNATURE, DBUS_BIG_ENDIAN, "");
MARSHAL_TEST_STRCMP (SIGNATURE, DBUS_LITTLE_ENDIAN, "");
MARSHAL_TEST_STRCMP (SIGNATURE, DBUS_BIG_ENDIAN, "a(ii)");
MARSHAL_TEST_STRCMP (SIGNATURE, DBUS_LITTLE_ENDIAN, "a(ii)");

/* Arrays */
MARSHAL_TEST_FIXED_ARRAY (INT16, DBUS_BIG_ENDIAN, array2);
MARSHAL_TEST_FIXED_ARRAY (INT16, DBUS_LITTLE_ENDIAN, array2);
MARSHAL_TEST_FIXED_ARRAY (UINT16, DBUS_BIG_ENDIAN, array2);
MARSHAL_TEST_FIXED_ARRAY (UINT16, DBUS_LITTLE_ENDIAN, array2);

MARSHAL_TEST_FIXED_ARRAY (INT32, DBUS_BIG_ENDIAN, array4);
MARSHAL_TEST_FIXED_ARRAY (INT32, DBUS_LITTLE_ENDIAN, array4);
MARSHAL_TEST_FIXED_ARRAY (UINT32, DBUS_BIG_ENDIAN, array4);
MARSHAL_TEST_FIXED_ARRAY (UINT32, DBUS_LITTLE_ENDIAN, array4);

MARSHAL_TEST_FIXED_ARRAY (BYTE, DBUS_BIG_ENDIAN, array1);
MARSHAL_TEST_FIXED_ARRAY (BYTE, DBUS_LITTLE_ENDIAN, array1);

#ifdef DBUS_HAVE_INT64
MARSHAL_TEST_FIXED_ARRAY (INT64, DBUS_BIG_ENDIAN, array8);

```

```

    MARSHAL_TEST_FIXED_ARRAY (INT64, DBUS_LITTLE_ENDIAN, array8);
#endif

#if 0

    /*
     * FIXME restore the set/pack tests
     */

#ifdef DBUS_HAVE_INT64
    /* set/pack 64-bit integers */
    _dbus_string_set_length (&str, 8);

    /* signed little */
    _dbus_marshal_set_int64 (&str, DBUS_LITTLE_ENDIAN,
                            0, DBUS_INT64_CONSTANT (-0x123456789abc7));

    _dbus_assert (DBUS_INT64_CONSTANT (-0x123456789abc7) ==
                  _dbus_unpack_int64 (DBUS_LITTLE_ENDIAN,
                                      _dbus_string_get_const_data
(&str)));

    /* signed big */
    _dbus_marshal_set_int64 (&str, DBUS_BIG_ENDIAN,
                            0, DBUS_INT64_CONSTANT (-0x123456789abc7));

    _dbus_assert (DBUS_INT64_CONSTANT (-0x123456789abc7) ==
                  _dbus_unpack_int64 (DBUS_BIG_ENDIAN,
                                      _dbus_string_get_const_data
(&str)));

    /* signed little pack */
    _dbus_pack_int64 (DBUS_INT64_CONSTANT (-0x123456789abc7),
                     DBUS_LITTLE_ENDIAN,
                     _dbus_string_get_data (&str));

    _dbus_assert (DBUS_INT64_CONSTANT (-0x123456789abc7) ==
                  _dbus_unpack_int64 (DBUS_LITTLE_ENDIAN,
                                      _dbus_string_get_const_data
(&str)));

    /* signed big pack */
    _dbus_pack_int64 (DBUS_INT64_CONSTANT (-0x123456789abc7),
                     DBUS_BIG_ENDIAN,
                     _dbus_string_get_data (&str));

    _dbus_assert (DBUS_INT64_CONSTANT (-0x123456789abc7) ==
                  _dbus_unpack_int64 (DBUS_BIG_ENDIAN,
                                      _dbus_string_get_const_data
(&str)));

    /* unsigned little */

```

```

    _dbus_marshal_set_uint64 (&str, DBUS_LITTLE_ENDIAN,
                              0, DBUS_UINT64_CONSTANT
(0x123456789abc7));

    _dbus_assert (DBUS_UINT64_CONSTANT (0x123456789abc7) ==
                  _dbus_unpack_uint64 (DBUS_LITTLE_ENDIAN,
                                        _dbus_string_get_const_data
(&str)));

    /* unsigned big */
    _dbus_marshal_set_uint64 (&str, DBUS_BIG_ENDIAN,
                              0, DBUS_UINT64_CONSTANT
(0x123456789abc7));

    _dbus_assert (DBUS_UINT64_CONSTANT (0x123456789abc7) ==
                  _dbus_unpack_uint64 (DBUS_BIG_ENDIAN,
                                        _dbus_string_get_const_data
(&str)));

    /* unsigned little pack */
    _dbus_pack_uint64 (DBUS_UINT64_CONSTANT (0x123456789abc7),
                      DBUS_LITTLE_ENDIAN,
                      _dbus_string_get_data (&str));

    _dbus_assert (DBUS_UINT64_CONSTANT (0x123456789abc7) ==
                  _dbus_unpack_uint64 (DBUS_LITTLE_ENDIAN,
                                        _dbus_string_get_const_data
(&str)));

    /* unsigned big pack */
    _dbus_pack_uint64 (DBUS_UINT64_CONSTANT (0x123456789abc7),
                      DBUS_BIG_ENDIAN,
                      _dbus_string_get_data (&str));

    _dbus_assert (DBUS_UINT64_CONSTANT (0x123456789abc7) ==
                  _dbus_unpack_uint64 (DBUS_BIG_ENDIAN,
                                        _dbus_string_get_const_data
(&str)));
#endif /* DBUS_HAVE_INT64 */

/* set/pack 32-bit integers */
_dbus_string_set_length (&str, 4);

/* signed little */
_dbus_marshal_set_int32 (&str, DBUS_LITTLE_ENDIAN,
                        0, -0x123456);

_dbus_assert (-0x123456 ==
              _dbus_unpack_int32 (DBUS_LITTLE_ENDIAN,
                                  _dbus_string_get_const_data
(&str)));

```

```

/* signed big */
_dbus_marshal_set_int32 (&str, DBUS_BIG_ENDIAN,
                        0, -0x123456);

_dbus_assert (-0x123456 ==
              _dbus_unpack_int32 (DBUS_BIG_ENDIAN,
                                  _dbus_string_get_const_data
(&str)));

/* signed little pack */
_dbus_pack_int32 (-0x123456,
                 DBUS_LITTLE_ENDIAN,
                 _dbus_string_get_data (&str));

_dbus_assert (-0x123456 ==
              _dbus_unpack_int32 (DBUS_LITTLE_ENDIAN,
                                  _dbus_string_get_const_data
(&str)));

/* signed big pack */
_dbus_pack_int32 (-0x123456,
                 DBUS_BIG_ENDIAN,
                 _dbus_string_get_data (&str));

_dbus_assert (-0x123456 ==
              _dbus_unpack_int32 (DBUS_BIG_ENDIAN,
                                  _dbus_string_get_const_data
(&str)));

/* unsigned little */
_dbus_marshal_set_uint32 (&str,
                          0, 0x123456,
                          DBUS_LITTLE_ENDIAN);

_dbus_assert (0x123456 ==
              _dbus_unpack_uint32 (DBUS_LITTLE_ENDIAN,
                                   _dbus_string_get_const_data
(&str)));

/* unsigned big */
_dbus_marshal_set_uint32 (&str,
                          0, 0x123456,
                          DBUS_BIG_ENDIAN);

_dbus_assert (0x123456 ==
              _dbus_unpack_uint32 (DBUS_BIG_ENDIAN,
                                   _dbus_string_get_const_data
(&str)));

/* unsigned little pack */
_dbus_pack_uint32 (0x123456,
                  DBUS_LITTLE_ENDIAN,

```

```

        _dbus_string_get_data (&str));

_dbus_assert (0x123456 ==
              _dbus_unpack_uint32 (DBUS_LITTLE_ENDIAN,
                                   _dbus_string_get_const_data
(&str)));

/* unsigned big pack */
_dbus_pack_uint32 (0x123456,
                  DBUS_BIG_ENDIAN,
                  _dbus_string_get_data (&str));

_dbus_assert (0x123456 ==
              _dbus_unpack_uint32 (DBUS_BIG_ENDIAN,
                                   _dbus_string_get_const_data
(&str)));

#endif /* set/pack tests for integers */

/* Strings in-place set */
byte_order = DBUS_LITTLE_ENDIAN;
while (TRUE)
{
    /* Init a string */
    _dbus_string_set_length (&str, 0);

    /* reset pos for the macros */
    pos = 0;

    MARSHAL_TEST_STRCMP (STRING, byte_order, "Hello world");

    /* Set it to something longer */
    _dbus_string_init_const (&t, "Hello world foo");

    v_STRING = _dbus_string_get_const_data (&t);
    _dbus_marshal_set_basic (&str, 0, DBUS_TYPE_STRING,
                            &v_STRING, byte_order, NULL, NULL);

    _dbus_marshal_read_basic (&str, 0, DBUS_TYPE_STRING,
                              &v_STRING, byte_order,
                              NULL);
    _dbus_assert (strcmp (v_STRING, "Hello world foo") == 0);

    /* Set it to something shorter */
    _dbus_string_init_const (&t, "Hello");

    v_STRING = _dbus_string_get_const_data (&t);
    _dbus_marshal_set_basic (&str, 0, DBUS_TYPE_STRING,
                            &v_STRING, byte_order, NULL, NULL);
    _dbus_marshal_read_basic (&str, 0, DBUS_TYPE_STRING,
                              &v_STRING, byte_order,
                              NULL);
}

```

```

    _dbus_assert (strcmp (v_STRING, "Hello") == 0);

    /* Do the other byte order */
    if (byte_order == DBUS_LITTLE_ENDIAN)
        byte_order = DBUS_BIG_ENDIAN;
    else
        break;
}

/* Clean up */
_dbus_string_free (&str);

return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

File = dbus-marshal-basic.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-basic.h  Marshalling routines for basic (primitive)
types
*
* Copyright (C) 2002  CodeFactory AB
* Copyright (C) 2004, 2005  Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
*
*/

#ifndef DBUS_MARSHAL_BASIC_H
#define DBUS_MARSHAL_BASIC_H

```

```

#ifdef HAVE_BYTESWAP_H
#include <byteswap.h>
#endif

#include <dbus/dbus-protocol.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-arch-deps.h>
#include <dbus/dbus-string.h>

#ifdef WORDS_BIGENDIAN
#define DBUS_COMPILER_BYTE_ORDER DBUS_BIG_ENDIAN
#else
#define DBUS_COMPILER_BYTE_ORDER DBUS_LITTLE_ENDIAN
#endif

#ifdef HAVE_BYTESWAP_H
#define DBUS_UINT16_SWAP_LE_BE_CONSTANT(val) bswap_16(val)
#define DBUS_UINT32_SWAP_LE_BE_CONSTANT(val) bswap_32(val)
#else /* HAVE_BYTESWAP_H */

#define DBUS_UINT16_SWAP_LE_BE_CONSTANT(val)      ((dbus_uint16_t) (
\
\   (dbus_uint16_t) ((dbus_uint16_t) (val) >> 8) |
\
\   (dbus_uint16_t) ((dbus_uint16_t) (val) << 8)))

#define DBUS_UINT32_SWAP_LE_BE_CONSTANT(val)      ((dbus_uint32_t) (
\
\   (((dbus_uint32_t) (val) & (dbus_uint32_t) 0x000000ffU) << 24) |
\
\   (((dbus_uint32_t) (val) & (dbus_uint32_t) 0x0000ff00U) <<  8) |
\
\   (((dbus_uint32_t) (val) & (dbus_uint32_t) 0x00ff0000U) >>  8) |
\
\   (((dbus_uint32_t) (val) & (dbus_uint32_t) 0xff000000U) >> 24)))

#endif /* HAVE_BYTESWAP_H */

#ifdef DBUS_HAVE_INT64

#ifdef HAVE_BYTESWAP_H
#define DBUS_UINT64_SWAP_LE_BE_CONSTANT(val) bswap_64(val)
#else /* HAVE_BYTESWAP_H */

#define DBUS_UINT64_SWAP_LE_BE_CONSTANT(val)      ((dbus_uint64_t) (
\
\   (((dbus_uint64_t) (val) &
\
\   (dbus_uint64_t) DBUS_UINT64_CONSTANT (0x00000000000000ff)) << 56)
\
\   (((dbus_uint64_t) (val) &

```



```

# define DBUS_UINT16_TO_LE(val) (DBUS_UINT16_SWAP_LE_BE (val))
# define DBUS_INT32_TO_BE(val) ((dbus_int32_t) (val))
# define DBUS_UINT32_TO_BE(val) ((dbus_uint32_t) (val))
# define DBUS_INT32_TO_LE(val) (DBUS_INT32_SWAP_LE_BE (val))
# define DBUS_UINT32_TO_LE(val) (DBUS_UINT32_SWAP_LE_BE (val))
# ifdef DBUS_HAVE_INT64
#   define DBUS_INT64_TO_BE(val) ((dbus_int64_t) (val))
#   define DBUS_UINT64_TO_BE(val) ((dbus_uint64_t) (val))
#   define DBUS_INT64_TO_LE(val) (DBUS_INT64_SWAP_LE_BE (val))
#   define DBUS_UINT64_TO_LE(val) (DBUS_UINT64_SWAP_LE_BE (val))
# endif /* DBUS_HAVE_INT64 */

#else /* WORDS_BIGENDIAN */

# define DBUS_INT16_TO_LE(val) ((dbus_int16_t) (val))
# define DBUS_UINT16_TO_LE(val) ((dbus_uint16_t) (val))
# define DBUS_INT16_TO_BE(val) ((dbus_int16_t) DBUS_UINT16_SWAP_LE_BE
(val))
# define DBUS_UINT16_TO_BE(val) (DBUS_UINT16_SWAP_LE_BE (val))
# define DBUS_INT32_TO_LE(val) ((dbus_int32_t) (val))
# define DBUS_UINT32_TO_LE(val) ((dbus_uint32_t) (val))
# define DBUS_INT32_TO_BE(val) ((dbus_int32_t) DBUS_UINT32_SWAP_LE_BE
(val))
# define DBUS_UINT32_TO_BE(val) (DBUS_UINT32_SWAP_LE_BE (val))
# ifdef DBUS_HAVE_INT64
#   define DBUS_INT64_TO_LE(val) ((dbus_int64_t) (val))
#   define DBUS_UINT64_TO_LE(val) ((dbus_uint64_t) (val))
#   define DBUS_INT64_TO_BE(val) ((dbus_int64_t)
DBUS_UINT64_SWAP_LE_BE (val))
#   define DBUS_UINT64_TO_BE(val) (DBUS_UINT64_SWAP_LE_BE (val))
# endif /* DBUS_HAVE_INT64 */
#endif

/* The transformation is symmetric, so the FROM just maps to the TO.
*/
#define DBUS_INT16_FROM_LE(val) (DBUS_INT16_TO_LE (val))
#define DBUS_UINT16_FROM_LE(val) (DBUS_UINT16_TO_LE (val))
#define DBUS_INT16_FROM_BE(val) (DBUS_INT16_TO_BE (val))
#define DBUS_UINT16_FROM_BE(val) (DBUS_UINT16_TO_BE (val))
#define DBUS_INT32_FROM_LE(val) (DBUS_INT32_TO_LE (val))
#define DBUS_UINT32_FROM_LE(val) (DBUS_UINT32_TO_LE (val))
#define DBUS_INT32_FROM_BE(val) (DBUS_INT32_TO_BE (val))
#define DBUS_UINT32_FROM_BE(val) (DBUS_UINT32_TO_BE (val))
#ifdef DBUS_HAVE_INT64
# define DBUS_INT64_FROM_LE(val) (DBUS_INT64_TO_LE (val))
# define DBUS_UINT64_FROM_LE(val) (DBUS_UINT64_TO_LE (val))
# define DBUS_INT64_FROM_BE(val) (DBUS_INT64_TO_BE (val))
# define DBUS_UINT64_FROM_BE(val) (DBUS_UINT64_TO_BE (val))
#endif /* DBUS_HAVE_INT64 */

#ifdef DBUS_DISABLE_ASSERT
#define _dbus_unpack_uint16(byte_order, data) \

```



```

n_elements,
byte_order,
*pos_after);
void      _dbus_marshal_read_basic      (const DBusString *str,
int      pos,
int      type,
void
*value,
byte_order,
*new_pos);
void      _dbus_marshal_read_fixed_multi (const DBusString *str,
int      pos,
int
element_type,
void
*value,
int
n_elements,
int
byte_order,
int
*new_pos);
void      _dbus_marshal_skip_basic      (const DBusString *str,
int      type,
int
byte_order,
int      *pos);
void      _dbus_marshal_skip_array      (const DBusString *str,
int
element_type,
int
byte_order,
int      *pos);
void      _dbus_marshal_set_uint32      (DBusString      *str,
int      pos,
dbus_uint32_t
value,
int
byte_order);
dbus_uint32_t _dbus_marshal_read_uint32 (const DBusString *str,
int      pos,
int
byte_order,
int
*new_pos);
int      _dbus_type_get_alignment      (int
typecode);

```

```

int          _dbus_type_get_alignment      (int
typecode);
const char*  _dbus_type_to_string        (int
typecode);

int          _dbus_first_type_in_signature (const DBusString *str,
int          pos);

int          _dbus_first_type_in_signature_c_str (const char
*str,
int          pos);

void _dbus_swap_array (unsigned char *data,
int          n_elements,
int          alignment);

#endif /* DBUS_MARSHAL_BASIC_H */

```

File = dbus-marshal-byteswap-util.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-byteswap-util.c Would be in dbus-marshal-byteswap.c
but tests/bus only

```

```

*
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

```

```

#include <config.h>

```

```

#ifdef DBUS_BUILD_TESTS

```

```

#include "dbus-marshal-byteswap.h"
#include "dbus-test.h"
#include <stdio.h>

static void
do_byteswap_test (int byte_order)
{
    int sequence;
    DBusString signature;
    DBusString body;
    int opposite_order;

    if (!_dbus_string_init (&signature) || !_dbus_string_init (&body))
        _dbus_assert_not_reached ("oom");

    opposite_order = byte_order == DBUS_LITTLE_ENDIAN ? DBUS_BIG_ENDIAN
: DBUS_LITTLE_ENDIAN;

    sequence = 0;
    while (dbus_internal_do_not_use_generate_bodies (sequence,
                                                    byte_order,
                                                    &signature, &body))
    {
        DBusString copy;
        DBusTypeReader body_reader;
        DBusTypeReader copy_reader;

        if (!_dbus_string_init (&copy))
            _dbus_assert_not_reached ("oom");

        if (!_dbus_string_copy (&body, 0, &copy, 0))
            _dbus_assert_not_reached ("oom");

        _dbus_marshal_byteswap (&signature, 0,
                                byte_order,
                                opposite_order,
                                &copy, 0);

        _dbus_type_reader_init (&body_reader, byte_order, &signature, 0,
                                &body, 0);
        _dbus_type_reader_init (&copy_reader, opposite_order,
                                &signature, 0,
                                &copy, 0);

        if (!_dbus_type_reader_equal_values (&body_reader,
                                &copy_reader))
        {
            _dbus_verbose_bytes_of_string (&signature, 0,
                                            _dbus_string_get_length
(&signature));
            _dbus_verbose_bytes_of_string (&body, 0,

```

```

        _dbus_string_get_length
(&body));
    _dbus_verbose_bytes_of_string (&copy, 0,
        _dbus_string_get_length
(&copy));

    _dbus_warn ("Byte-swapped data did not have same values as
original data\n");
    _dbus_assert_not_reached ("test failed");
}

    _dbus_string_free (&copy);

    _dbus_string_set_length (&signature, 0);
    _dbus_string_set_length (&body, 0);
    ++sequence;
}

    _dbus_string_free (&signature);
    _dbus_string_free (&body);

    printf (" %d blocks swapped from order '%c' to '%c'\n",
        sequence, byte_order, opposite_order);
}

dbus_bool_t
_dbus_marshal_byteswap_test (void)
{
    do_byteswap_test (DBUS_LITTLE_ENDIAN);
    do_byteswap_test (DBUS_BIG_ENDIAN);

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-marshal-byteswap.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-byteswap.c Swap a block of marshaled data
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or

```

```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

```

```

#include <config.h>
#include "dbus-marshal-byteswap.h"
#include "dbus-marshal-basic.h"
#include "dbus-signature.h"

```

```
/**
```

```

* @addtogroup DBusMarshal
* @{
*/

```

```

static void
byteswap_body_helper (DBusTypeReader      *reader,
                     dbus_bool_t         walk_reader_to_end,
                     int                 old_byte_order,
                     int                 new_byte_order,
                     unsigned char       *p,
                     unsigned char       **new_p)
{
    int current_type;

    while ((current_type = _dbus_type_reader_get_current_type (reader))
!= DBUS_TYPE_INVALID)
    {
        switch (current_type)
        {
            case DBUS_TYPE_BYTE:
                ++p;
                break;

            case DBUS_TYPE_INT16:
            case DBUS_TYPE_UINT16:
                {
                    p = _DBUS_ALIGN_ADDRESS (p, 2);
                    *((dbus_uint16_t*)p) = DBUS_UINT16_SWAP_LE_BE
(*((dbus_uint16_t*)p));
                    p += 2;
                }
                break;
        }
    }
}

```

```

    case DBUS_TYPE_BOOLEAN:
    case DBUS_TYPE_INT32:
    case DBUS_TYPE_UINT32:
        {
            p = _DBUS_ALIGN_ADDRESS (p, 4);
            *((dbus_uint32_t*)p) = DBUS_UINT32_SWAP_LE_BE
(*((dbus_uint32_t*)p));
            p += 4;
        }
        break;

    case DBUS_TYPE_INT64:
    case DBUS_TYPE_UINT64:
    case DBUS_TYPE_DOUBLE:
        {
            p = _DBUS_ALIGN_ADDRESS (p, 8);
#ifdef DBUS_HAVE_INT64
            *((dbus_uint64_t*)p) = DBUS_UINT64_SWAP_LE_BE
(*((dbus_uint64_t*)p));
#else
            _dbus_swap_array (p, 1, 8);
#endif
            p += 8;
        }
        break;

    case DBUS_TYPE_ARRAY:
    case DBUS_TYPE_STRING:
    case DBUS_TYPE_OBJECT_PATH:
        {
            dbus_uint32_t array_len;

            p = _DBUS_ALIGN_ADDRESS (p, 4);

            array_len = _dbus_unpack_uint32 (old_byte_order, p);

            *((dbus_uint32_t*)p) = DBUS_UINT32_SWAP_LE_BE
(*((dbus_uint32_t*)p));
            p += 4;

            if (current_type == DBUS_TYPE_ARRAY)
                {
                    int elem_type;
                    int alignment;

                    elem_type = _dbus_type_reader_get_element_type
(reader);
                    alignment = _dbus_type_get_alignment (elem_type);

                    _dbus_assert ((array_len / alignment) <
DBUS_MAXIMUM_ARRAY_LENGTH);
                }
        }

```



```

        p = _DBUS_ALIGN_ADDRESS (p, alignment);

        if (dbus_type_is_fixed (elem_type))
            {
                if (alignment > 1)
                    _dbus_swap_array (p, array_len / alignment,
alignment);
                p += array_len;
            }
        else
            {
                DBusTypeReader sub;
                const unsigned char *array_end;

                array_end = p + array_len;

                _dbus_type_reader_recurse (reader, &sub);

                while (p < array_end)
                    {
                        byteswap_body_helper (&sub,
                                                FALSE,
                                                old_byte_order,
                                                new_byte_order,
                                                p, &p);
                    }
            }
        else
            {
                _dbus_assert (current_type == DBUS_TYPE_STRING ||
                             current_type == DBUS_TYPE_OBJECT_PATH);

                p += (array_len + 1); /* + 1 for nul */
            }
        break;

case DBUS_TYPE_SIGNATURE:
    {
        dbus_uint32_t sig_len;

        sig_len = *p;

        p += (sig_len + 2); /* +2 for len and nul */
    }
    break;

case DBUS_TYPE_VARIANT:
    {
        /* 1 byte sig len, sig typecodes, align to

```

```

        * contained-type-boundary, values.
        */
        dbus_uint32_t sig_len;
        DBusString sig;
        DBusTypeReader sub;
        int contained_alignment;

        sig_len = *p;
        ++p;

        _dbus_string_init_const_len (&sig, p, sig_len);

        p += (sig_len + 1); /* 1 for nul */

        contained_alignment = _dbus_type_get_alignment
(_dbus_first_type_in_signature (&sig, 0));

        p = _DBUS_ALIGN_ADDRESS (p, contained_alignment);

        _dbus_type_reader_init_types_only (&sub, &sig, 0);

        byteswap_body_helper (&sub, FALSE, old_byte_order,
new_byte_order, p, &p);
    }
    break;

    case DBUS_TYPE_STRUCT:
    case DBUS_TYPE_DICT_ENTRY:
    {
        DBusTypeReader sub;

        p = _DBUS_ALIGN_ADDRESS (p, 8);

        _dbus_type_reader_recurse (reader, &sub);

        byteswap_body_helper (&sub, TRUE, old_byte_order,
new_byte_order, p, &p);
    }
    break;

    case DBUS_TYPE_UNIX_FD:
        /* fds can only be passed on a local machine, so byte order
must always match */
        _dbus_assert_not_reached("attempted to byteswap unix fds
which makes no sense");
        break;

    default:
        _dbus_assert_not_reached ("invalid typecode in supposedly-
validated signature");
        break;
}

```

```

        if (walk_reader_to_end)
            _dbus_type_reader_next (reader);
        else
            break;
    }

    if (new_p)
        *new_p = p;
}

/**
 * Byteswaps the marshaled data in the given value_str.
 *
 * @param signature the types in the value_str
 * @param signature_start where in signature is the signature
 * @param old_byte_order the old byte order
 * @param new_byte_order the new byte order
 * @param value_str the string containing the body
 * @param value_pos where the values start
 */
void
_dbus_marshal_byteswap (const DBusString *signature,
                        int signature_start,
                        int old_byte_order,
                        int new_byte_order,
                        DBusString *value_str,
                        int value_pos)
{
    DBusTypeReader reader;

    _dbus_assert (value_pos >= 0);
    _dbus_assert (value_pos <= _dbus_string_get_length (value_str));

    if (old_byte_order == new_byte_order)
        return;

    _dbus_type_reader_init_types_only (&reader,
                                       signature, signature_start);

    byteswap_body_helper (&reader, TRUE,
                          old_byte_order, new_byte_order,
                          _dbus_string_get_data_len (value_str,
value_pos, 0),
                          NULL);
}

/** @} */

/* Tests in dbus-marshal-byteswap-util.c */

```

File = dbus-marshal-byteswap.h

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-byteswap.h  Swap a block of marshaled data
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 *
 */
```

```
#ifndef DBUS_MARSHAL_BYTESWAP_H
#define DBUS_MARSHAL_BYTESWAP_H

#include <dbus/dbus-protocol.h>
#include <dbus/dbus-marshal-recursive.h>
```

```
void _dbus_marshal_byteswap (const DBusString *signature,
                             int signature_start,
                             int old_byte_order,
                             int new_byte_order,
                             DBusString *value_str,
                             int value_pos);
```

```
#endif /* DBUS_MARSHAL_BYTESWAP_H */
```

File = dbus-marshal-header.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-header.c  Managing marshaling/demarshaling of message
headers
 *
 */
```

```
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus/dbus-shared.h"
#include "dbus-marshall-header.h"
#include "dbus-marshall-recursive.h"
#include "dbus-marshall-byteswap.h"
```

```
/**
 * @addtogroup DBusMarshal
 *
 * @{
 */
```

```
/* Not thread locked, but strictly const/read-only so should be OK
 */
```

```
/** Static #DBusString containing the signature of a message header */
_DBUS_STRING_DEFINE_STATIC(_dbus_header_signature_str,
DBus_HEADER_SIGNATURE);
/** Static #DBusString containing the local interface */
_DBUS_STRING_DEFINE_STATIC(_dbus_local_interface_str,
DBus_INTERFACE_LOCAL);
/** Static #DBusString containing the local path */
_DBUS_STRING_DEFINE_STATIC(_dbus_local_path_str,
DBus_PATH_LOCAL);
```

```
/** Offset from start of _dbus_header_signature_str to the signature
of the fields array */
#define FIELDS_ARRAY_SIGNATURE_OFFSET 6
```

```

/** Offset from start of _dbus_header_signature_str to the signature
of an element of the fields array */
#define FIELDS_ARRAY_ELEMENT_SIGNATURE_OFFSET 7

/** Offset to byte order from start of header */
#define BYTE_ORDER_OFFSET 0
/** Offset to type from start of header */
#define TYPE_OFFSET 1
/** Offset to flags from start of header */
#define FLAGS_OFFSET 2
/** Offset to version from start of header */
#define VERSION_OFFSET 3
/** Offset to body length from start of header */
#define BODY_LENGTH_OFFSET 4
/** Offset to client serial from start of header */
#define SERIAL_OFFSET 8
/** Offset to fields array length from start of header */
#define FIELDS_ARRAY_LENGTH_OFFSET 12
/** Offset to first field in header */
#define FIRST_FIELD_OFFSET 16

typedef struct
{
    unsigned char code; /**< the field code */
    unsigned char type; /**< the value type */
} HeaderFieldType;

static const HeaderFieldType
_dbus_header_field_types[DBUS_HEADER_FIELD_LAST+1] = {
    { DBUS_HEADER_FIELD_INVALID, DBUS_TYPE_INVALID },
    { DBUS_HEADER_FIELD_PATH, DBUS_TYPE_OBJECT_PATH },
    { DBUS_HEADER_FIELD_INTERFACE, DBUS_TYPE_STRING },
    { DBUS_HEADER_FIELD_MEMBER, DBUS_TYPE_STRING },
    { DBUS_HEADER_FIELD_ERROR_NAME, DBUS_TYPE_STRING },
    { DBUS_HEADER_FIELD_REPLY_SERIAL, DBUS_TYPE_UINT32 },
    { DBUS_HEADER_FIELD_DESTINATION, DBUS_TYPE_STRING },
    { DBUS_HEADER_FIELD_SENDER, DBUS_TYPE_STRING },
    { DBUS_HEADER_FIELD_SIGNATURE, DBUS_TYPE_SIGNATURE },
    { DBUS_HEADER_FIELD_UNIX_FDS, DBUS_TYPE_UINT32 }
};

/** Macro to look up the correct type for a field */
#define EXPECTED_TYPE_OF_FIELD(field)
    (_dbus_header_field_types[field].type)

/** The most padding we could ever need for a header */
#define MAX_POSSIBLE_HEADER_PADDING 7
static dbus_bool_t
reserve_header_padding (DBusHeader *header)
{
    _dbus_assert (header->padding <= MAX_POSSIBLE_HEADER_PADDING);

```

```

    if (!_dbus_string_lengthen (&header->data,
                                MAX_POSSIBLE_HEADER_PADDING - header->padding))
        return FALSE;
    header->padding = MAX_POSSIBLE_HEADER_PADDING;
    return TRUE;
}

static void
correct_header_padding (DBusHeader *header)
{
    int unpadded_len;

    _dbus_assert (header->padding == 7);

    _dbus_string_shorten (&header->data, header->padding);
    unpadded_len = _dbus_string_get_length (&header->data);

    if (!_dbus_string_align_length (&header->data, 8))
        _dbus_assert_not_reached ("couldn't pad header though enough padding was preallocated");

    header->padding = _dbus_string_get_length (&header->data) - unpadded_len;
}

/** Compute the end of the header, ignoring padding */
#define HEADER_END_BEFORE_PADDING(header) \
    (_dbus_string_get_length (&(header)->data) - (header)->padding)

/**
 * Invalidates all fields in the cache. This may be used when the
 * cache is totally uninitialized (contains junk) so should not
 * look at what's in there now.
 *
 * @param header the header
 */
static void
_dbus_header_cache_invalidate_all (DBusHeader *header)
{
    int i;

    i = 0;
    while (i <= DBUS_HEADER_FIELD_LAST)
    {
        header->fields[i].value_pos = _DBUS_HEADER_FIELD_VALUE_UNKNOWN;
        ++i;
    }
}

/**

```

```

* Caches one field
*
* @param header the header
* @param field_code the field
* @param variant_reader the reader for the variant in the field
*/
static void
_dbus_header_cache_one (DBusHeader      *header,
                        int              field_code,
                        DBusTypeReader *variant_reader)
{
    header->fields[field_code].value_pos =
        _dbus_type_reader_get_value_pos (variant_reader);

#ifdef 0
    _dbus_verbose ("cached value_pos %d for field %d\n",
                  header->fields[field_code].value_pos, field_code)
#endif
}

/**
 * Returns the header's byte order.
 *
 * @param header the header
 * @returns the byte order
 */
char
_dbus_header_get_byte_order (const DBusHeader *header)
{
    _dbus_assert (_dbus_string_get_length (&header->data) >
                 BYTE_ORDER_OFFSET);

    return (char) _dbus_string_get_byte (&header->data,
                                       BYTE_ORDER_OFFSET);
}

/**
 * Revalidates the fields cache
 *
 * @param header the header
 */
static void
_dbus_header_cache_revalidate (DBusHeader *header)
{
    DBusTypeReader array;
    DBusTypeReader reader;
    int i;

    i = 0;
    while (i <= DBUS_HEADER_FIELD_LAST)
    {

```



```

        header->fields[i].value_pos =
_DBUS_HEADER_FIELD_VALUE_NONEXISTENT;
        ++i;
    }

    _dbus_type_reader_init (&reader,
                            _dbus_header_get_byte_order (header),
                            &_dbus_header_signature_str,
                            FIELDS_ARRAY_SIGNATURE_OFFSET,
                            &header->data,
                            FIELDS_ARRAY_LENGTH_OFFSET);

    _dbus_type_reader_recurse (&reader, &array);

    while (_dbus_type_reader_get_current_type (&array) !=
_DBUS_TYPE_INVALID)
    {
        DBusTypeReader sub;
        DBusTypeReader variant;
        unsigned char field_code;

        _dbus_type_reader_recurse (&array, &sub);

        _dbus_assert (_dbus_type_reader_get_current_type (&sub) ==
_DBUS_TYPE_BYTE);
        _dbus_type_reader_read_basic (&sub, &field_code);

        /* Unknown fields should be ignored */
        if (field_code > DBUS_HEADER_FIELD_LAST)
            goto next_field;

        _dbus_type_reader_next (&sub);

        _dbus_assert (_dbus_type_reader_get_current_type (&sub) ==
_DBUS_TYPE_VARIANT);
        _dbus_type_reader_recurse (&sub, &variant);

        _dbus_header_cache_one (header, field_code, &variant);

    next_field:
        _dbus_type_reader_next (&array);
    }
}

/**
 * Checks for a field, updating the cache if required.
 *
 * @param header the header
 * @param field the field to check
 * @returns #FALSE if the field doesn't exist
 */
static dbus_bool_t

```

```

_dbus_header_cache_check (DBusHeader      *header,
                          int              field)
{
    _dbus_assert (field <= DBUS_HEADER_FIELD_LAST);

    if (header->fields[field].value_pos ==
        _DBUS_HEADER_FIELD_VALUE_UNKNOWN)
        _dbus_header_cache_revalidate (header);

    if (header->fields[field].value_pos ==
        _DBUS_HEADER_FIELD_VALUE_NONEXISTENT)
        return FALSE;

    return TRUE;
}

/**
 * Checks whether a field is known not to exist. It may exist
 * even if it's not known to exist.
 *
 * @param header the header
 * @param field the field to check
 * @returns #FALSE if the field definitely doesn't exist
 */
static dbus_bool_t
_dbus_header_cache_known_nonexistent (DBusHeader      *header,
                                      int              field)
{
    _dbus_assert (field <= DBUS_HEADER_FIELD_LAST);

    return (header->fields[field].value_pos ==
        _DBUS_HEADER_FIELD_VALUE_NONEXISTENT);
}

/**
 * Writes a struct of { byte, variant } with the given basic type.
 *
 * @param writer the writer (should be ready to write a struct)
 * @param type the type of the value
 * @param value the value as for _dbus_marshal_set_basic()
 * @returns #FALSE if no memory
 */
static dbus_bool_t
write_basic_field (DBusTypeWriter *writer,
                  int              field,
                  int              type,
                  const void      *value)
{
    DBusTypeWriter sub;
    DBusTypeWriter variant;
    int start;
    int padding;

```

```

unsigned char field_byte;
DBusString contained_type;
char buf[2];

start = writer->value_pos;
padding = _dbus_string_get_length (writer->value_str) - start;

if (!_dbus_type_writer_recurse (writer, DBUS_TYPE_STRUCT,
                                NULL, 0, &sub))
    goto append_failed;

field_byte = field;
if (!_dbus_type_writer_write_basic (&sub, DBUS_TYPE_BYTE,
                                    &field_byte))
    goto append_failed;

buf[0] = type;
buf[1] = '\\0';
_dbus_string_init_const_len (&contained_type, buf, 1);

if (!_dbus_type_writer_recurse (&sub, DBUS_TYPE_VARIANT,
                                &contained_type, 0, &variant))
    goto append_failed;

if (!_dbus_type_writer_write_basic (&variant, type, value))
    goto append_failed;

if (!_dbus_type_writer_unrecurse (&sub, &variant))
    goto append_failed;

if (!_dbus_type_writer_unrecurse (writer, &sub))
    goto append_failed;

return TRUE;

append_failed:
_dbus_string_delete (writer->value_str,
                    start,
                    _dbus_string_get_length (writer->value_str) -
start - padding);
return FALSE;
}

/**
 * Sets a struct of { byte, variant } with the given basic type.
 *
 * @param reader the reader (should be iterating over the array
pointing at the field to set)
 * @param type the type of the value
 * @param value the value as for _dbus_marshal_set_basic()
 * @param realign_root where to realign from
 * @returns #FALSE if no memory

```

```

*/
static dbus_bool_t
set_basic_field (DBusTypeReader      *reader,
                int                   field,
                int                   type,
                const void            *value,
                const DBusTypeReader *realign_root)
{
    DBusTypeReader sub;
    DBusTypeReader variant;

    _dbus_type_reader_recurse (reader, &sub);

    _dbus_assert (_dbus_type_reader_get_current_type (&sub) ==
DBUS_TYPE_BYTE);
#ifdef DBUS_DISABLE_ASSERT
    {
        unsigned char v_BYTE;
        _dbus_type_reader_read_basic (&sub, &v_BYTE);
        _dbus_assert (((int) v_BYTE) == field);
    }
#endif

    if (!_dbus_type_reader_next (&sub))
        _dbus_assert_not_reached ("no variant field?");

    _dbus_type_reader_recurse (&sub, &variant);
    _dbus_assert (_dbus_type_reader_get_current_type (&variant) ==
type);

    if (!_dbus_type_reader_set_basic (&variant, value, realign_root))
        return FALSE;

    return TRUE;
}

/**
 * Gets the type of the message.
 *
 * @param header the header
 * @returns the type
 */
int
_dbus_header_get_message_type (DBusHeader *header)
{
    int type;

    type = _dbus_string_get_byte (&header->data, TYPE_OFFSET);
    _dbus_assert (type != DBUS_MESSAGE_TYPE_INVALID);

    return type;
}

```

```

/**
 * Sets the serial number of a header.  This can only be done once on
 * a header.
 *
 * @param header the header
 * @param serial the serial
 */
void
_dbus_header_set_serial (DBusHeader *header,
                        dbus_uint32_t serial)
{
    /* we use this function to set the serial on outgoing
     * messages, and to reset the serial in dbus_message_copy;
     * this assertion should catch a double-set on outgoing.
     */
    _dbus_assert (_dbus_header_get_serial (header) == 0 ||
                  serial == 0);

    _dbus_marshal_set_uint32 (&header->data,
                              SERIAL_OFFSET,
                              serial,
                              _dbus_header_get_byte_order (header));
}

/**
 * See dbus_message_get_serial()
 *
 * @param header the header
 * @returns the client serial
 */
dbus_uint32_t
_dbus_header_get_serial (DBusHeader *header)
{
    return _dbus_marshal_read_uint32 (&header->data,
                                      SERIAL_OFFSET,
                                      _dbus_header_get_byte_order
                                      (header),
                                      NULL);
}

/**
 * Re-initializes a header that was previously initialized and never
 * freed.  After this, to make the header valid you have to call
 * _dbus_header_create().
 *
 * @param header header to re-initialize
 */
void
_dbus_header_reinit (DBusHeader *header)
{
    _dbus_string_set_length (&header->data, 0);
}

```

```

    header->padding = 0;

    _dbus_header_cache_invalidate_all (header);
}

/**
 * Initializes a header, but doesn't prepare it for use;
 * to make the header valid, you have to call _dbus_header_create().
 *
 * @param header header to initialize
 * @param byte_order byte order of the header
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
_dbus_header_init (DBusHeader *header)
{
    if (!_dbus_string_init_preallocated (&header->data, 32))
        return FALSE;

    _dbus_header_reinit (header);

    return TRUE;
}

/**
 * Frees a header.
 *
 * @param header the header
 */
void
_dbus_header_free (DBusHeader *header)
{
    _dbus_string_free (&header->data);
}

/**
 * Initializes dest with a copy of the given header.
 * Resets the message serial to 0 on the copy.
 *
 * @param header header to copy
 * @param dest destination for copy
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
_dbus_header_copy (const DBusHeader *header,
                  DBusHeader *dest)
{
    *dest = *header;

    if (!_dbus_string_init_preallocated (&dest->data,

```

```

                                                                    _dbus_string_get_length
(&header->data))
    return FALSE;

    if (!_dbus_string_copy (&header->data, 0, &dest->data, 0))
    {
        _dbus_string_free (&dest->data);
        return FALSE;
    }

    /* Reset the serial */
    _dbus_header_set_serial (dest, 0);

    return TRUE;
}

/**
 * Fills in the primary fields of the header, so the header is ready
 * for use. #NULL may be specified for some or all of the fields to
 * avoid adding those fields. Some combinations of fields don't make
 * sense, and passing them in will trigger an assertion failure.
 *
 * @param header the header
 * @param message_type the message type
 * @param destination destination field or #NULL
 * @param path path field or #NULL
 * @param interface interface field or #NULL
 * @param member member field or #NULL
 * @param error_name error name or #NULL
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
_dbus_header_create (DBusHeader *header,
                    int          byte_order,
                    int          message_type,
                    const char *destination,
                    const char *path,
                    const char *interface,
                    const char *member,
                    const char *error_name)
{
    unsigned char v_BYTE;
    dbus_uint32_t v_UINT32;
    DBusTypeWriter writer;
    DBusTypeWriter array;

    _dbus_assert (byte_order == DBUS_LITTLE_ENDIAN ||
                 byte_order == DBUS_BIG_ENDIAN);
    _dbus_assert (((interface || message_type !=
DBUS_MESSAGE_TYPE_SIGNAL) && member) ||
                 (error_name) ||
                 !(interface || member || error_name));

```

```

_dbus_assert (_dbus_string_get_length (&header->data) == 0);

if (!reserve_header_padding (header))
    return FALSE;

_dbus_type_writer_init_values_only (&writer, byte_order,
                                     &_dbus_header_signature_str, 0,
                                     &header->data,
                                     HEADER_END_BEFORE_PADDING
(header));

v_BYTE = byte_order;
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_BYTE,
                                     &v_BYTE))
    goto oom;

v_BYTE = message_type;
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_BYTE,
                                     &v_BYTE))
    goto oom;

v_BYTE = 0; /* flags */
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_BYTE,
                                     &v_BYTE))
    goto oom;

v_BYTE = DBUS_MAJOR_PROTOCOL_VERSION;
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_BYTE,
                                     &v_BYTE))
    goto oom;

v_UINT32 = 0; /* body length */
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_UINT32,
                                     &v_UINT32))
    goto oom;

v_UINT32 = 0; /* serial */
if (!_dbus_type_writer_write_basic (&writer, DBUS_TYPE_UINT32,
                                     &v_UINT32))
    goto oom;

if (!_dbus_type_writer_recurse (&writer, DBUS_TYPE_ARRAY,
                                &_dbus_header_signature_str,
                                FIELDS_ARRAY_SIGNATURE_OFFSET,
                                &array))
    goto oom;

/* Marshal all the fields (Marshall Fields?) */

if (path != NULL)
{
    if (!write_basic_field (&array,

```



```

        DBUS_HEADER_FIELD_PATH,
        DBUS_TYPE_OBJECT_PATH,
        &path))
    goto oom;
}

if (destination != NULL)
{
    if (!write_basic_field (&array,
        DBUS_HEADER_FIELD_DESTINATION,
        DBUS_TYPE_STRING,
        &destination))

        goto oom;
}

if (interface != NULL)
{
    if (!write_basic_field (&array,
        DBUS_HEADER_FIELD_INTERFACE,
        DBUS_TYPE_STRING,
        &interface))

        goto oom;
}

if (member != NULL)
{
    if (!write_basic_field (&array,
        DBUS_HEADER_FIELD_MEMBER,
        DBUS_TYPE_STRING,
        &member))

        goto oom;
}

if (error_name != NULL)
{
    if (!write_basic_field (&array,
        DBUS_HEADER_FIELD_ERROR_NAME,
        DBUS_TYPE_STRING,
        &error_name))

        goto oom;
}

if (!_dbus_type_writer_unrecurse (&writer, &array))
    goto oom;

correct_header_padding (header);

return TRUE;

oom:
_dbus_string_delete (&header->data, 0,

```

```

        _dbus_string_get_length (&header->data) -
header->padding);
    correct_header_padding (header);

    return FALSE;
}

/**
 * Given data long enough to contain the length of the message body
 * and the fields array, check whether the data is long enough to
 * contain the entire message (assuming the claimed lengths are
 * accurate). Also checks that the lengths are in sanity parameters.
 *
 * @param max_message_length maximum length of a valid message
 * @param validity return location for why the data is invalid if it
is
 * @param byte_order return location for byte order
 * @param fields_array_len return location for claimed fields array
length
 * @param header_len return location for claimed header length
 * @param body_len return location for claimed body length
 * @param str the data
 * @param start start of data, 8-aligned
 * @param len length of data
 * @returns #TRUE if the data is long enough for the claimed length,
and the lengths were valid
 */
dbus_bool_t
_dbus_header_have_message_untrusted (int
max_message_length,
                                     DBusValidity    *validity,
                                     int               *byte_order,
                                     int
*fields_array_len,
                                     int               *header_len,
                                     int               *body_len,
                                     const DBusString *str,
                                     int               start,
                                     int               len)
{
    dbus_uint32_t header_len_unsigned;
    dbus_uint32_t fields_array_len_unsigned;
    dbus_uint32_t body_len_unsigned;

    _dbus_assert (start >= 0);
    _dbus_assert (start < _DBUS_INT32_MAX / 2);
    _dbus_assert (len >= 0);

    _dbus_assert (start == (int) _DBUS_ALIGN_VALUE (start, 8));

```

```

    *byte_order = _dbus_string_get_byte (str, start +
    BYTE_ORDER_OFFSET);

    if (*byte_order != DBUS_LITTLE_ENDIAN && *byte_order !=
    DBUS_BIG_ENDIAN)
    {
        *validity = DBUS_INVALID_BAD_BYTE_ORDER;
        return FALSE;
    }

    _dbus_assert (FIELDS_ARRAY_LENGTH_OFFSET + 4 <= len);
    fields_array_len_unsigned = _dbus_marshal_read_uint32 (str, start +
    FIELDS_ARRAY_LENGTH_OFFSET,
                                                                *byte_order,
    NULL);

    if (fields_array_len_unsigned > (unsigned) max_message_length)
    {
        *validity = DBUS_INVALID_INSANE_FIELDS_ARRAY_LENGTH;
        return FALSE;
    }

    _dbus_assert (BODY_LENGTH_OFFSET + 4 < len);
    body_len_unsigned = _dbus_marshal_read_uint32 (str, start +
    BODY_LENGTH_OFFSET,
                                                                *byte_order, NULL);

    if (body_len_unsigned > (unsigned) max_message_length)
    {
        *validity = DBUS_INVALID_INSANE_BODY_LENGTH;
        return FALSE;
    }

    header_len_unsigned = FIRST_FIELD_OFFSET +
    fields_array_len_unsigned;
    header_len_unsigned = _DBUS_ALIGN_VALUE (header_len_unsigned, 8);

    /* overflow should be impossible since the lengths aren't allowed to
    * be huge.
    */
    _dbus_assert (max_message_length < _DBUS_INT32_MAX / 2);
    if (body_len_unsigned + header_len_unsigned > (unsigned)
    max_message_length)
    {
        *validity = DBUS_INVALID_MESSAGE_TOO_LONG;
        return FALSE;
    }

    _dbus_assert (body_len_unsigned < (unsigned) _DBUS_INT32_MAX);
    _dbus_assert (fields_array_len_unsigned < (unsigned)
    _DBUS_INT32_MAX);
    _dbus_assert (header_len_unsigned < (unsigned) _DBUS_INT32_MAX);

```

```

*body_len = body_len_unsigned;
*fields_array_len = fields_array_len_unsigned;
*header_len = header_len_unsigned;

*validity = DBUS_VALID;

_dbus_verbose ("have %d bytes, need body %u + header %u = %u\n",
               len, body_len_unsigned, header_len_unsigned,
               body_len_unsigned + header_len_unsigned);

return (body_len_unsigned + header_len_unsigned) <= (unsigned) len;
}

static DBusValidity
check_mandatory_fields (DBusHeader *header)
{
#define REQUIRE_FIELD(name) do { if (header->fields[DBUS_HEADER_FIELD_##name].value_pos < 0) return DBUS_INVALID_MISSING_##name; } while (0)

switch (_dbus_header_get_message_type (header))
{
case DBUS_MESSAGE_TYPE_SIGNAL:
    REQUIRE_FIELD (INTERFACE);
    /* FALL THRU - signals also require the path and member */
case DBUS_MESSAGE_TYPE_METHOD_CALL:
    REQUIRE_FIELD (PATH);
    REQUIRE_FIELD (MEMBER);
    break;
case DBUS_MESSAGE_TYPE_ERROR:
    REQUIRE_FIELD (ERROR_NAME);
    REQUIRE_FIELD (REPLY_SERIAL);
    break;
case DBUS_MESSAGE_TYPE_METHOD_RETURN:
    REQUIRE_FIELD (REPLY_SERIAL);
    break;
default:
    /* other message types allowed but ignored */
    break;
}

return DBUS_VALID;
}

static DBusValidity
load_and_validate_field (DBusHeader *header,
                        int field,
                        DBusTypeReader *variant_reader)
{
int type;
int expected_type;

```

```

const DBusString *value_str;
int value_pos;
int str_data_pos;
dbus_uint32_t v_UINT32;
int bad_string_code;
dbus_bool_t (* string_validation_func) (const DBusString *str,
                                        int start, int len);

/* Supposed to have been checked already */
_dbus_assert (field <= DBUS_HEADER_FIELD_LAST);
_dbus_assert (field != DBUS_HEADER_FIELD_INVALID);

/* Before we can cache a field, we need to know it has the right
type */
type = _dbus_type_reader_get_current_type (variant_reader);

_dbus_assert (_dbus_header_field_types[field].code == field);

expected_type = EXPECTED_TYPE_OF_FIELD (field);
if (type != expected_type)
{
    _dbus_verbose ("Field %d should have type %d but has %d\n",
                  field, expected_type, type);
    return DBUS_INVALID_HEADER_FIELD_HAS_WRONG_TYPE;
}

/* If the field was provided twice, we aren't happy */
if (header->fields[field].value_pos >= 0)
{
    _dbus_verbose ("Header field %d seen a second time\n", field);
    return DBUS_INVALID_HEADER_FIELD_APPEARS_TWICE;
}

/* Now we can cache and look at the field content */
_dbus_verbose ("initially caching field %d\n", field);
_dbus_header_cache_one (header, field, variant_reader);

string_validation_func = NULL;

/* make compiler happy that all this is initialized */
v_UINT32 = 0;
value_str = NULL;
value_pos = -1;
str_data_pos = -1;
bad_string_code = DBUS_VALID;

if (expected_type == DBUS_TYPE_UINT32)
{
    _dbus_header_get_field_basic (header, field, expected_type,
                                &v_UINT32);
}
else if (expected_type == DBUS_TYPE_STRING ||

```

```

        expected_type == DBUS_TYPE_OBJECT_PATH ||
        expected_type == DBUS_TYPE_SIGNATURE)
    {
        _dbus_header_get_field_raw (header, field,
                                   &value_str, &value_pos);
        str_data_pos = _DBUS_ALIGN_VALUE (value_pos, 4) + 4;
    }
else
    {
        _dbus_assert_not_reached ("none of the known fields should have
this type");
    }

switch (field)
    {
case DBUS_HEADER_FIELD_DESTINATION:
    string_validation_func = _dbus_validate_bus_name;
    bad_string_code = DBUS_INVALID_BAD_DESTINATION;
    break;
case DBUS_HEADER_FIELD_INTERFACE:
    string_validation_func = _dbus_validate_interface;
    bad_string_code = DBUS_INVALID_BAD_INTERFACE;

    if (_dbus_string_equal_substring (&_dbus_local_interface_str,
                                      0,
                                      _dbus_string_get_length
(&_dbus_local_interface_str),
                                      value_str, str_data_pos))
        {
            _dbus_verbose ("Message is on the local interface\n");
            return DBUS_INVALID_USES_LOCAL_INTERFACE;
        }
    break;

case DBUS_HEADER_FIELD_MEMBER:
    string_validation_func = _dbus_validate_member;
    bad_string_code = DBUS_INVALID_BAD_MEMBER;
    break;

case DBUS_HEADER_FIELD_ERROR_NAME:
    string_validation_func = _dbus_validate_error_name;
    bad_string_code = DBUS_INVALID_BAD_ERROR_NAME;
    break;

case DBUS_HEADER_FIELD_SENDER:
    string_validation_func = _dbus_validate_bus_name;
    bad_string_code = DBUS_INVALID_BAD_SENDER;
    break;

case DBUS_HEADER_FIELD_PATH:
    /* OBJECT_PATH was validated generically due to its type */
    string_validation_func = NULL;

```

```

        if (_dbus_string_equal_substring (&_dbus_local_path_str,
                                         0,
                                         _dbus_string_get_length
(&_dbus_local_path_str),
                                         value_str, str_data_pos))
        {
            _dbus_verbose ("Message is from the local path\n");
            return DBUS_INVALID_USES_LOCAL_PATH;
        }
        break;

case DBUS_HEADER_FIELD_REPLY_SERIAL:
    /* Can't be 0 */
    if (v_UINT32 == 0)
    {
        return DBUS_INVALID_BAD_SERIAL;
    }
    break;

case DBUS_HEADER_FIELD_UNIX_FDS:
    /* Every value makes sense */
    break;

case DBUS_HEADER_FIELD_SIGNATURE:
    /* SIGNATURE validated generically due to its type */
    string_validation_func = NULL;
    break;

default:
    _dbus_assert_not_reached ("unknown field shouldn't be seen
here");
    break;
}

if (string_validation_func)
{
    dbus_uint32_t len;

    _dbus_assert (bad_string_code != DBUS_VALID);

    len = _dbus_marshal_read_uint32 (value_str, value_pos,
                                     _dbus_header_get_byte_order
(header),
                                     NULL);

#ifdef 0
    _dbus_verbose ("Validating string header field; code %d if
fails\n",
                  bad_string_code);
#endif
    if (!(*string_validation_func) (value_str, str_data_pos, len))

```

```

        return bad_string_code;
    }

    return DBUS_VALID;
}

/**
 * Creates a message header from potentially-untrusted data. The
 * return value is #TRUE if there was enough memory and the data was
 * valid. If it returns #TRUE, the header will be created. If it
 * returns #FALSE and *validity == #DBUS_VALIDITY_UNKNOWN_OOM_ERROR,
 * then there wasn't enough memory. If it returns #FALSE
 * and *validity != #DBUS_VALIDITY_UNKNOWN_OOM_ERROR then the data was
 * invalid.
 *
 * The byte_order, fields_array_len, and body_len args should be from
 * _dbus_header_have_message_untrusted(). Validation performed in
 * _dbus_header_have_message_untrusted() is assumed to have been
 * already done.
 *
 * @param header the header (must be initialized)
 * @param mode whether to do validation
 * @param validity return location for invalidity reason
 * @param byte_order byte order from header
 * @param fields_array_len claimed length of fields array
 * @param body_len claimed length of body
 * @param header_len claimed length of header
 * @param str a string
 * @param start start of header, 8-aligned
 * @param len length of string to look at
 * @returns #FALSE if no memory or data was invalid, #TRUE otherwise
 */
dbus_bool_t
_dbus_header_load (DBusHeader      *header,
                  DBusValidationMode mode,
                  DBusValidity     *validity,
                  int               byte_order,
                  int               fields_array_len,
                  int               header_len,
                  int               body_len,
                  const DBusString *str,
                  int               start,
                  int               len)
{
    int leftover;
    DBusValidity v;
    DBusTypeReader reader;
    DBusTypeReader array_reader;
    unsigned char v_byte;
    dbus_uint32_t v_uint32;
    dbus_uint32_t serial;
    int padding_start;

```



```

int padding_len;
int i;

_dbus_assert (start == (int) _DBUS_ALIGN_VALUE (start, 8));
_dbus_assert (header_len <= len);
_dbus_assert (_dbus_string_get_length (&header->data) == 0);

if (!_dbus_string_copy_len (str, start, header_len, &header->data,
0))
{
    _dbus_verbose ("Failed to copy buffer into new header\n");
    *validity = DBUS_VALIDITY_UNKNOWN_OOM_ERROR;
    return FALSE;
}

if (mode == DBUS_VALIDATION_MODE_WE_TRUST_THIS_DATA_ABSOLUTELY)
{
    leftover = len - header_len - body_len - start;
}
else
{
    v = _dbus_validate_body_with_reason
(&_dbus_header_signature_str, 0,
                                     byte_order,
                                     &leftover,
                                     str, start, len);

    if (v != DBUS_VALID)
    {
        *validity = v;
        goto invalid;
    }
}

_dbus_assert (leftover < len);

padding_len = header_len - (FIRST_FIELD_OFFSET + fields_array_len);
padding_start = start + FIRST_FIELD_OFFSET + fields_array_len;
_dbus_assert (start + header_len == (int) _DBUS_ALIGN_VALUE
(padding_start, 8));
_dbus_assert (start + header_len == padding_start + padding_len);

if (mode != DBUS_VALIDATION_MODE_WE_TRUST_THIS_DATA_ABSOLUTELY)
{
    if (!_dbus_string_validate_nul (str, padding_start,
padding_len))
    {
        *validity = DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
        goto invalid;
    }
}

```

```

header->padding = padding_len;

if (mode == DBUS_VALIDATION_MODE_WE_TRUST_THIS_DATA_ABSOLUTELY)
{
    *validity = DBUS_VALID;
    return TRUE;
}

/* We now know the data is well-formed, but we have to check that
 * it's valid.
 */

_dbus_type_reader_init (&reader,
                        byte_order,
                        &_dbus_header_signature_str, 0,
                        str, start);

/* BYTE ORDER */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_BYTE);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
BYTE_ORDER_OFFSET);
_dbus_type_reader_read_basic (&reader, &v_byte);
_dbus_type_reader_next (&reader);

_dbus_assert (v_byte == byte_order);

/* MESSAGE TYPE */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_BYTE);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
TYPE_OFFSET);
_dbus_type_reader_read_basic (&reader, &v_byte);
_dbus_type_reader_next (&reader);

/* unknown message types are supposed to be ignored, so only
validation here is
 * that it isn't invalid
 */
if (v_byte == DBUS_MESSAGE_TYPE_INVALID)
{
    *validity = DBUS_INVALID_BAD_MESSAGE_TYPE;
    goto invalid;
}

/* FLAGS */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_BYTE);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
FLAGS_OFFSET);
_dbus_type_reader_read_basic (&reader, &v_byte);
_dbus_type_reader_next (&reader);

```

```

/* unknown flags should be ignored */

/* PROTOCOL VERSION */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_BYTE);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
VERSION_OFFSET);
_dbus_type_reader_read_basic (&reader, &v_byte);
_dbus_type_reader_next (&reader);

if (v_byte != DBUS_MAJOR_PROTOCOL_VERSION)
{
    *validity = DBUS_INVALID_BAD_PROTOCOL_VERSION;
    goto invalid;
}

/* BODY LENGTH */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_UINT32);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
BODY_LENGTH_OFFSET);
_dbus_type_reader_read_basic (&reader, &v_uint32);
_dbus_type_reader_next (&reader);

_dbus_assert (body_len == (signed) v_uint32);

/* SERIAL */
_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_UINT32);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
SERIAL_OFFSET);
_dbus_type_reader_read_basic (&reader, &serial);
_dbus_type_reader_next (&reader);

if (serial == 0)
{
    *validity = DBUS_INVALID_BAD_SERIAL;
    goto invalid;
}

_dbus_assert (_dbus_type_reader_get_current_type (&reader) ==
DBUS_TYPE_ARRAY);
_dbus_assert (_dbus_type_reader_get_value_pos (&reader) ==
FIELDS_ARRAY_LENGTH_OFFSET);

_dbus_type_reader_recurse (&reader, &array_reader);
while (_dbus_type_reader_get_current_type (&array_reader) !=
DBUS_TYPE_INVALID)
{
    DBusTypeReader struct_reader;
    DBusTypeReader variant_reader;

```

```

    unsigned char field_code;

    _dbus_assert (_dbus_type_reader_get_current_type (&array_reader)
== DBUS_TYPE_STRUCT);

    _dbus_type_reader_recurse (&array_reader, &struct_reader);

    _dbus_assert (_dbus_type_reader_get_current_type
(&struct_reader) == DBUS_TYPE_BYTE);
    _dbus_type_reader_read_basic (&struct_reader, &field_code);
    _dbus_type_reader_next (&struct_reader);

    if (field_code == DBUS_HEADER_FIELD_INVALID)
    {
        _dbus_verbose ("invalid header field code\n");
        *validity = DBUS_INVALID_HEADER_FIELD_CODE;
        goto invalid;
    }

    if (field_code > DBUS_HEADER_FIELD_LAST)
    {
        _dbus_verbose ("unknown header field code %d, skipping\n",
            field_code);
        goto next_field;
    }

    _dbus_assert (_dbus_type_reader_get_current_type
(&struct_reader) == DBUS_TYPE_VARIANT);
    _dbus_type_reader_recurse (&struct_reader, &variant_reader);

    v = load_and_validate_field (header, field_code,
&variant_reader);
    if (v != DBUS_VALID)
    {
        _dbus_verbose ("Field %d was invalid\n", field_code);
        *validity = v;
        goto invalid;
    }

next_field:
    _dbus_type_reader_next (&array_reader);
}

/* Anything we didn't fill in is now known not to exist */
i = 0;
while (i <= DBUS_HEADER_FIELD_LAST)
{
    if (header->fields[i].value_pos ==
_DBUS_HEADER_FIELD_VALUE_UNKNOWN)
        header->fields[i].value_pos =
_DBUS_HEADER_FIELD_VALUE_NONEXISTENT;
    ++i;
}

```

```

    }

    v = check_mandatory_fields (header);
    if (v != DBUS_VALID)
    {
        _dbus_verbose ("Mandatory fields were missing, code %d\n", v);
        *validity = v;
        goto invalid;
    }

    *validity = DBUS_VALID;
    return TRUE;

invalid:
    _dbus_string_set_length (&header->data, 0);
    return FALSE;
}

/**
 * Fills in the correct body length.
 *
 * @param header the header
 * @param body_len the length of the body
 */
void
_dbus_header_update_lengths (DBusHeader *header,
                             int          body_len)
{
    _dbus_marshal_set_uint32 (&header->data,
                              BODY_LENGTH_OFFSET,
                              body_len,
                              _dbus_header_get_byte_order (header));
}

/**
 * Try to find the given field.
 *
 * @param header the header
 * @param field the field code
 * @param reader a type reader; on success this is left pointing at
the struct
 * (uv) for the field, while on failure it is left pointing into
empty space
 * at the end of the header fields
 * @param realign_root another type reader; on success or failure it
is left
 * pointing to the beginning of the array of fields (i.e. the thing
that might
 * need realigning)
 * @returns #TRUE on success
 */
static dbus_bool_t

```

```

find_field_for_modification (DBusHeader      *header,
                             int            field,
                             DBusTypeReader *reader,
                             DBusTypeReader *realign_root)
{
    dbus_bool_t retval;

    retval = FALSE;

    _dbus_type_reader_init (realign_root,
                            _dbus_header_get_byte_order (header),
                            &_dbus_header_signature_str,
                            FIELDS_ARRAY_SIGNATURE_OFFSET,
                            &header->data,
                            FIELDS_ARRAY_LENGTH_OFFSET);

    _dbus_type_reader_recurse (realign_root, reader);

    while (_dbus_type_reader_get_current_type (reader) !=
          DBUS_TYPE_INVALID)
    {
        DBusTypeReader sub;
        unsigned char field_code;

        _dbus_type_reader_recurse (reader, &sub);

        _dbus_assert (_dbus_type_reader_get_current_type (&sub) ==
          DBUS_TYPE_BYTE);
        _dbus_type_reader_read_basic (&sub, &field_code);

        if (field_code == (unsigned) field)
        {
            _dbus_assert (_dbus_type_reader_get_current_type (reader) ==
          DBUS_TYPE_STRUCT);
            retval = TRUE;
            goto done;
        }

        _dbus_type_reader_next (reader);
    }

done:
    return retval;
}

/**
 * Sets the value of a field with basic type. If the value is a string
 * value, it isn't allowed to be #NULL. If the field doesn't exist,
 * it will be created.
 *
 * @param header the header
 * @param field the field to set

```

```

* @param type the type of the value
* @param value the value as for _dbus_marshal_set_basic()
* @returns #FALSE if no memory
*/
dbus_bool_t
_dbus_header_set_field_basic (DBusHeader      *header,
                             int             field,
                             int             type,
                             const void     *value)
{
    _dbus_assert (field <= DBUS_HEADER_FIELD_LAST);

    if (!reserve_header_padding (header))
        return FALSE;

    /* If the field exists we set, otherwise we append */
    if (_dbus_header_cache_check (header, field))
    {
        DBusTypeReader reader;
        DBusTypeReader realign_root;

        if (!find_field_for_modification (header, field,
                                          &reader, &realign_root))
            _dbus_assert_not_reached ("field was marked present in cache
but wasn't found");

        if (!set_basic_field (&reader, field, type, value,
&realign_root))
            return FALSE;
    }
    else
    {
        DBusTypeWriter writer;
        DBusTypeWriter array;

        _dbus_type_writer_init_values_only (&writer,
                                          _dbus_header_get_byte_order
(header),
                                          &_dbus_header_signature_str,
FIELD_ARRAY_SIGNATURE_OFFSET,
                                          &header->data,
FIELD_ARRAY_LENGTH_OFFSET);

        /* recurse into array without creating a new length, and jump to
* end of array.
*/
        if (!_dbus_type_writer_append_array (&writer,
&_dbus_header_signature_str,
FIELD_ARRAY_ELEMENT_SIGNATURE_OFFSET,

```

```

                                &array))
    _dbus_assert_not_reached ("recurse into ARRAY should not have
used memory");

    _dbus_assert (array.u.array.len_pos ==
FIELDS_ARRAY_LENGTH_OFFSET);
    _dbus_assert (array.u.array.start_pos == FIRST_FIELD_OFFSET);
    _dbus_assert (array.value_pos == HEADER_END_BEFORE_PADDING
(header));

    if (!write_basic_field (&array,
                            field, type, value))
        return FALSE;

    if (!_dbus_type_writer_unrecurse (&writer, &array))
        _dbus_assert_not_reached ("unrecurse from ARRAY should not
have used memory");
}

correct_header_padding (header);

/* We could be smarter about this (only invalidate fields after the
* one we modified, or even only if the one we modified changed
* length). But this hack is a start.
*/
_dbus_header_cache_invalidate_all (header);

return TRUE;
}

/**
 * Gets the value of a field with basic type. If the field
 * doesn't exist, returns #FALSE, otherwise returns #TRUE.
 *
 * @param header the header
 * @param field the field to get
 * @param type the type of the value
 * @param value the value as for _dbus_marshal_read_basic()
 * @returns #FALSE if the field doesn't exist
 */
dbus_bool_t
_dbus_header_get_field_basic (DBusHeader *header,
                             int field,
                             int type,
                             void *value)
{
    _dbus_assert (field != DBUS_HEADER_FIELD_INVALID);
    _dbus_assert (field <= DBUS_HEADER_FIELD_LAST);
    _dbus_assert (_dbus_header_field_types[field].code == field);
    /* in light of this you might ask why the type is passed in;
    * the only rationale I can think of is so the caller has
    * to specify its expectation and breaks if we change it

```



```

    */
    _dbus_assert (type == EXPECTED_TYPE_OF_FIELD (field));

    if (!_dbus_header_cache_check (header, field))
        return FALSE;

    _dbus_assert (header->fields[field].value_pos >= 0);

    _dbus_marshal_read_basic (&header->data,
                              header->fields[field].value_pos,
                              type, value, _dbus_header_get_byte_order
(header),
                              NULL);

    return TRUE;
}

/**
 * Gets the raw marshaled data for a field. If the field doesn't
 * exist, returns #FALSE, otherwise returns #TRUE. Returns the start
 * of the marshaled data, i.e. usually the byte where the length
 * starts (for strings and arrays) or for basic types just the value
 * itself.
 *
 * @param header the header
 * @param field the field to get
 * @param str return location for the data string
 * @param pos return location for start of field value
 * @returns #FALSE if the field doesn't exist
 */
dbus_bool_t
_dbus_header_get_field_raw (DBusHeader      *header,
                            int             field,
                            const DBusString **str,
                            int            *pos)
{
    if (!_dbus_header_cache_check (header, field))
        return FALSE;

    if (str)
        *str = &header->data;
    if (pos)
        *pos = header->fields[field].value_pos;

    return TRUE;
}

/**
 * Deletes a field, if it exists.
 *
 * @param header the header
 * @param field the field to delete

```

```

* @returns #FALSE if no memory
*/
dbus_bool_t
_dbus_header_delete_field (DBusHeader *header,
                           int         field)
{
    DBusTypeReader reader;
    DBusTypeReader realign_root;

    if (_dbus_header_cache_known_nonexistent (header, field))
        return TRUE; /* nothing to do */

    /* Scan to the field we want, delete and realign, reappend
     * padding. Field may turn out not to exist.
     */
    if (!find_field_for_modification (header, field,
                                      &reader, &realign_root))
        return TRUE; /* nothing to do */

    if (!reserve_header_padding (header))
        return FALSE;

    if (!_dbus_type_reader_delete (&reader,
                                   &realign_root))
        return FALSE;

    correct_header_padding (header);

    _dbus_header_cache_invalidate_all (header);

    _dbus_assert (!_dbus_header_cache_check (header, field)); /*
Expensive assertion ... */

    return TRUE;
}

/**
 * Toggles a message flag bit, turning on the bit if value = TRUE and
 * flipping it off if value = FALSE.
 *
 * @param header the header
 * @param flag the message flag to toggle
 * @param value toggle on or off
 */
void
_dbus_header_toggle_flag (DBusHeader *header,
                          dbus_uint32_t flag,
                          dbus_bool_t value)
{
    unsigned char *flags_p;

```

```

    flags_p = _dbus_string_get_data_len (&header->data, FLAGS_OFFSET,
1);

    if (value)
        *flags_p |= flag;
    else
        *flags_p &= ~flag;
}

/**
 * Gets a message flag bit, returning TRUE if the bit is set.
 *
 * @param header the header
 * @param flag the message flag to get
 * @returns #TRUE if the flag is set
 */
dbus_bool_t
_dbus_header_get_flag (DBusHeader *header,
                      dbus_uint32_t flag)
{
    const unsigned char *flags_p;

    flags_p = _dbus_string_get_const_data_len (&header->data,
FLAGS_OFFSET, 1);

    return (*flags_p & flag) != 0;
}

/**
 * Swaps the header into the given order if required.
 *
 * @param header the header
 * @param new_order the new byte order
 */
void
_dbus_header_byteswap (DBusHeader *header,
                      int new_order)
{
    char byte_order;

    byte_order = _dbus_header_get_byte_order (header);

    if (byte_order == new_order)
        return;

    _dbus_marshal_byteswap (&_dbus_header_signature_str,
                          0, byte_order,
                          new_order,
                          &header->data, 0);

    _dbus_string_set_byte (&header->data, BYTE_ORDER_OFFSET, new_order);
}

```

```

/** @} */

File = dbus-marshal-header.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-header.h Managing marshaling/demarshaling of message
headers
*
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef DBUS_MARSHAL_HEADER_H
#define DBUS_MARSHAL_HEADER_H

#include <dbus/dbus-marshal-basic.h>
#include <dbus/dbus-marshal-validate.h>

typedef struct DBusHeader DBusHeader;
typedef struct DBusHeaderField DBusHeaderField;

#define _DBUS_HEADER_FIELD_VALUE_UNKNOWN -1
#define _DBUS_HEADER_FIELD_VALUE_NONEXISTENT -2

/**
* Cached information about a header field in the message
*/
struct DBusHeaderField
{
    int value_pos; /**< Position of field value, or -1/-2 */

```

```

};

/**
 * Message header data and some cached details of it.
 */
struct DBusHeader
{
    DBusString data; /**< Header network data, stored
                    * separately from body so we can
                    * independently realloc it.
                    */

    DBusHeaderField fields[DBUS_HEADER_FIELD_LAST + 1]; /**< Track the
location
                                                    * of each field
in header
                                                    */

    dbus_uint32_t padding : 3;          /**< bytes of alignment in header
*/
    dbus_uint32_t byte_order : 8;      /**< byte order of header */
};

dbus_bool_t    _dbus_header_init          (DBusHeader
*header);
void          _dbus_header_free          (DBusHeader
*header);
void          _dbus_header_reinit        (DBusHeader
*header);
dbus_bool_t    _dbus_header_create        (DBusHeader
*header,
int
byte_order,
int
type,
const char
*destination,
const char
*path,
const char
*interface,
const char
*member,
const char
*error_name);
dbus_bool_t    _dbus_header_copy          (const DBusHeader
*header,
DBusHeader
*dest);
int           _dbus_header_get_message_type (DBusHeader
*header);

```

```

void          _dbus_header_set_serial      (DBusHeader
*header,
serial);
dbus_uint32_t _dbus_header_get_serial      (DBusHeader
*header);
void          _dbus_header_update_lengths  (DBusHeader
*header,
int
body_len);
dbus_bool_t   _dbus_header_set_field_basic (DBusHeader
*header,
int
field,
int
type,
const void
*value);
dbus_bool_t   _dbus_header_get_field_basic (DBusHeader
*header,
int
field,
int
type,
void
*value);
dbus_bool_t   _dbus_header_get_field_raw   (DBusHeader
*header,
int
field,
const DBusString
**str,
int
*pos);
dbus_bool_t   _dbus_header_delete_field    (DBusHeader
*header,
int
field);
void          _dbus_header_toggle_flag     (DBusHeader
*header,
dbus_uint32_t
flag,
dbus_bool_t
value);
dbus_bool_t   _dbus_header_get_flag        (DBusHeader
*header,
dbus_uint32_t
flag);
dbus_bool_t   _dbus_header_ensure_signature (DBusHeader
*header,
DBusString
**type_str,

```

```

int
*type_pos);
dbus_bool_t _dbus_header_have_message_untrusted (int
max_message_length,
DBusValidity
*validity,
int
*byte_order,
int
*fields_array_len,
int
*header_len,
int
*body_len,
const DBusString
*str,
int
start,
int
len);
dbus_bool_t _dbus_header_load (DBusHeader
*header,
DBusValidationMode
mode,
DBusValidity
*validity,
int
byte_order,
int
fields_array_len,
int
header_len,
int
body_len,
const DBusString
*str,
int
start,
int
len);
void _dbus_header_byteswap (DBusHeader
*header,
int
new_order);
char _dbus_header_get_byte_order (const DBusHeader
*header);

#endif /* DBUS_MARSHAL_HEADER_H */

```

```

File = dbus-marshal-recursive-util.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-recursive-util.c Would be in dbus-marshal-
recursive.c, but only used in bus/tests
*
* Copyright (C) 2004, 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>

#ifdef DBUS_BUILD_TESTS

#include "dbus-marshal-recursive.h"
#include "dbus-marshal-basic.h"
#include "dbus-signature.h"
#include "dbus-internals.h"
#include <string.h>

static void
basic_value_zero (DBusBasicValue *value)
{
#ifdef DBUS_HAVE_INT64
    value->u64 = 0;
#else
    value->eight.first32 = 0;
    value->eight.second32 = 0;
#endif
}

```



```

static dbus_bool_t
basic_value_equal (int type,
                  DBusBasicValue *lhs,
                  DBusBasicValue *rhs)
{
    if (type == DBUS_TYPE_STRING ||
        type == DBUS_TYPE_SIGNATURE ||
        type == DBUS_TYPE_OBJECT_PATH)
    {
        return strcmp (lhs->str, rhs->str) == 0;
    }
    else
    {
#ifdef DBUS_HAVE_INT64
        return lhs->u64 == rhs->u64;
#else
        return lhs->eight.first32 == rhs->eight.first32 &&
            lhs->eight.second32 == rhs->eight.second32;
#endif
    }
}

static dbus_bool_t
equal_values_helper (DBusTypeReader *lhs,
                    DBusTypeReader *rhs)
{
    int lhs_type;
    int rhs_type;

    lhs_type = _dbus_type_reader_get_current_type (lhs);
    rhs_type = _dbus_type_reader_get_current_type (rhs);

    if (lhs_type != rhs_type)
        return FALSE;

    if (lhs_type == DBUS_TYPE_INVALID)
        return TRUE;

    if (dbus_type_is_basic (lhs_type))
    {
        DBusBasicValue lhs_value;
        DBusBasicValue rhs_value;

        basic_value_zero (&lhs_value);
        basic_value_zero (&rhs_value);

        _dbus_type_reader_read_basic (lhs, &lhs_value);
        _dbus_type_reader_read_basic (rhs, &rhs_value);

        return basic_value_equal (lhs_type, &lhs_value, &rhs_value);
    }
    else

```

```

    {
        DBusTypeReader lhs_sub;
        DBusTypeReader rhs_sub;

        _dbus_type_reader_recurse (lhs, &lhs_sub);
        _dbus_type_reader_recurse (rhs, &rhs_sub);

        return equal_values_helper (&lhs_sub, &rhs_sub);
    }
}

/**
 * See whether the two readers point to identical data blocks.
 *
 * @param lhs reader 1
 * @param rhs reader 2
 * @returns #TRUE if the data blocks have the same values
 */
dbus_bool_t
_dbus_type_reader_equal_values (const DBusTypeReader *lhs,
                                const DBusTypeReader *rhs)
{
    DBusTypeReader copy_lhs = *lhs;
    DBusTypeReader copy_rhs = *rhs;

    return equal_values_helper (&copy_lhs, &copy_rhs);
}

/* TESTS */

#ifdef DOXYGEN_SHOULD_SKIP_THIS

#include "dbus-test.h"
#include "dbus-list.h"
#include <stdio.h>
#include <stdlib.h>

/* Whether to do the OOM stuff (only with other expensive tests) */
#define TEST_OOM_HANDLING 0
/* We do start offset 0 through 9, to get various alignment cases.
Still this
 * obviously makes the test suite run 10x as slow.
 */
#define MAX_INITIAL_OFFSET 9

/* Largest iteration count to test copying, realignment,
 * etc. with. i.e. we only test this stuff with some of the smaller
 * data sets.
 */
#define MAX_ITERATIONS_FOR_EXPENSIVE_TESTS 1000

typedef struct

```

```

{
    int byte_order;
    int initial_offset;
    DBusString signature;
    DBusString body;
} DataBlock;

typedef struct
{
    int saved_sig_len;
    int saved_body_len;
} DataBlockState;

#define N_FENCE_BYTES 5
#define FENCE_BYTES_STR "abcde"
#define INITIAL_PADDING_BYTE '\0'

static dbus_bool_t
data_block_init (DataBlock *block,
                int         byte_order,
                int         initial_offset)
{
    if (!_dbus_string_init (&block->signature))
        return FALSE;

    if (!_dbus_string_init (&block->body))
    {
        _dbus_string_free (&block->signature);
        return FALSE;
    }

    if (!_dbus_string_insert_bytes (&block->signature, 0,
initial_offset,
                                INITIAL_PADDING_BYTE) ||
        !_dbus_string_insert_bytes (&block->body, 0, initial_offset,
                                INITIAL_PADDING_BYTE) ||
        !_dbus_string_append (&block->signature, FENCE_BYTES_STR) ||
        !_dbus_string_append (&block->body, FENCE_BYTES_STR))
    {
        _dbus_string_free (&block->signature);
        _dbus_string_free (&block->body);
        return FALSE;
    }

    block->byte_order = byte_order;
    block->initial_offset = initial_offset;

    return TRUE;
}

static void
data_block_save (DataBlock      *block,

```

```

        DataBlockState *state)
{
    state->saved_sig_len = _dbus_string_get_length (&block->signature) -
N_FENCE_BYTES;
    state->saved_body_len = _dbus_string_get_length (&block->body) -
N_FENCE_BYTES;
}

static void
data_block_restore (DataBlock      *block,
                   DataBlockState *state)
{
    _dbus_string_delete (&block->signature,
                        state->saved_sig_len,
                        _dbus_string_get_length (&block->signature) -
state->saved_sig_len - N_FENCE_BYTES);
    _dbus_string_delete (&block->body,
                        state->saved_body_len,
                        _dbus_string_get_length (&block->body) - state-
>saved_body_len - N_FENCE_BYTES);
}

static void
data_block_verify (DataBlock *block)
{
    if (!_dbus_string_ends_with_c_str (&block->signature,
                                      FENCE_BYTES_STR))
    {
        int offset;

        offset = _dbus_string_get_length (&block->signature) -
N_FENCE_BYTES - 8;
        if (offset < 0)
            offset = 0;

        _dbus_verbose_bytes_of_string (&block->signature,
                                      offset,
                                      _dbus_string_get_length (&block-
>signature) - offset);
        _dbus_assert_not_reached ("block did not verify: bad bytes at
end of signature");
    }
    if (!_dbus_string_ends_with_c_str (&block->body,
                                      FENCE_BYTES_STR))
    {
        int offset;

        offset = _dbus_string_get_length (&block->body) - N_FENCE_BYTES
- 8;
        if (offset < 0)
            offset = 0;
    }
}

```



```

    if (writer)
        _dbus_type_writer_init (writer,
                                block->byte_order,
                                &block->signature,
                                _dbus_string_get_length (&block-
>signature) - N_FENCE_BYTES,
                                &block->body,
                                _dbus_string_get_length (&block->body) -
N_FENCE_BYTES);
    }

static void
real_check_expected_type (DBusTypeReader *reader,
                          int expected,
                          const char *funcname,
                          int line)
{
    int t;

    t = _dbus_type_reader_get_current_type (reader);

    if (t != expected)
    {
        _dbus_warn ("Read type %s while expecting %s at %s line %d\n",
                    _dbus_type_to_string (t),
                    _dbus_type_to_string (expected),
                    funcname, line);

        _dbus_assert_not_reached ("read wrong type");
    }
}

#define check_expected_type(reader, expected) real_check_expected_type
(reader, expected, _DBUS_FUNCTION_NAME, __LINE__)

#define NEXT_EXPECTING_TRUE(reader) do { if (!_dbus_type_reader_next
(reader)) \
{ \
    _dbus_warn ("_dbus_type_reader_next() should have returned TRUE at
%s %d\n", \
                _DBUS_FUNCTION_NAME, __LINE__); \
    _dbus_assert_not_reached ("test failed"); \
} \
} while (0)

#define NEXT_EXPECTING_FALSE(reader) do { if (_dbus_type_reader_next
(reader)) \

```



```

static dbus_bool_t byte_read_value      (DBusTypeWriter *writer,
int seed);
static dbus_bool_t byte_set_value      (TestTypeNode *node,
DBusTypeReader *reader,
int seed);
*realign_root,
static dbus_bool_t double_write_value (TestTypeNode *node,
DataBlock *block,
DBusTypeWriter *writer,
int seed);
static dbus_bool_t double_read_value (TestTypeNode *node,
DBusTypeReader *reader,
int seed);
static dbus_bool_t double_set_value (TestTypeNode *node,
DBusTypeReader *reader,
DBusTypeReader
int seed);
*realign_root,
static dbus_bool_t object_path_write_value (TestTypeNode *node,
DataBlock *block,
DBusTypeWriter *writer,
int seed);
static dbus_bool_t object_path_read_value (TestTypeNode *node,
DBusTypeReader *reader,
int seed);
static dbus_bool_t object_path_set_value (TestTypeNode *node,
DBusTypeReader *reader,
DBusTypeReader
int seed);
*realign_root,
static dbus_bool_t signature_write_value (TestTypeNode *node,
DataBlock *block,
DBusTypeWriter *writer,
int seed);
static dbus_bool_t signature_read_value (TestTypeNode *node,
DBusTypeReader *reader,
int seed);
static dbus_bool_t signature_set_value (TestTypeNode *node,
DBusTypeReader *reader,
DBusTypeReader
int seed);
*realign_root,
static dbus_bool_t struct_write_value (TestTypeNode *node,
DataBlock *block,
DBusTypeWriter *writer,
int seed);
static dbus_bool_t struct_read_value (TestTypeNode *node,
DBusTypeReader *reader,

```



```

static const TestTypeNodeClass int16_class = {
    DBUS_TYPE_INT16,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int16_write_value,
    int16_read_value,
    int16_set_value,
    NULL,
    int16_write_multi,
    int16_read_multi
};

```

```

static const TestTypeNodeClass uint16_class = {
    DBUS_TYPE_UINT16,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int16_write_value, /* recycle from int16 */
    int16_read_value, /* recycle from int16 */
    int16_set_value, /* recycle from int16 */
    NULL,
    int16_write_multi, /* recycle from int16 */
    int16_read_multi /* recycle from int16 */
};

```

```

static const TestTypeNodeClass int32_class = {
    DBUS_TYPE_INT32,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int32_write_value,
    int32_read_value,
    int32_set_value,
    NULL,
    int32_write_multi,
    int32_read_multi
};

```

```

static const TestTypeNodeClass uint32_class = {
    DBUS_TYPE_UINT32,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int32_write_value, /* recycle from int32 */
    int32_read_value, /* recycle from int32 */
    int32_set_value, /* recycle from int32 */
    NULL,
};

```

```

    int32_write_multi, /* recycle from int32 */
    int32_read_multi   /* recycle from int32 */
};

static const TestTypeNodeClass int64_class = {
    DBUS_TYPE_INT64,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int64_write_value,
    int64_read_value,
    int64_set_value,
    NULL,
    NULL, /* FIXME */
    NULL /* FIXME */
};

static const TestTypeNodeClass uint64_class = {
    DBUS_TYPE_UINT64,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    int64_write_value, /* recycle from int64 */
    int64_read_value,  /* recycle from int64 */
    int64_set_value,   /* recycle from int64 */
    NULL,
    NULL, /* FIXME */
    NULL /* FIXME */
};

static const TestTypeNodeClass string_0_class = {
    DBUS_TYPE_STRING,
    sizeof (TestTypeNode),
    0, /* string length */
    NULL,
    NULL,
    string_write_value,
    string_read_value,
    string_set_value,
    NULL,
    NULL,
    NULL
};

static const TestTypeNodeClass string_1_class = {
    DBUS_TYPE_STRING,
    sizeof (TestTypeNode),
    1, /* string length */
    NULL,
    NULL,

```

```

    string_write_value,
    string_read_value,
    string_set_value,
    NULL,
    NULL,
    NULL
};

/* with nul, a len 3 string should fill 4 bytes and thus is "special"
*/
static const TestTypeNodeClass string_3_class = {
    DBUS_TYPE_STRING,
    sizeof (TestTypeNode),
    3, /* string length */
    NULL,
    NULL,
    string_write_value,
    string_read_value,
    string_set_value,
    NULL,
    NULL,
    NULL
};

/* with nul, a len 8 string should fill 9 bytes and thus is "special"
(far-fetched I suppose) */
static const TestTypeNodeClass string_8_class = {
    DBUS_TYPE_STRING,
    sizeof (TestTypeNode),
    8, /* string length */
    NULL,
    NULL,
    string_write_value,
    string_read_value,
    string_set_value,
    NULL,
    NULL,
    NULL
};

static const TestTypeNodeClass bool_class = {
    DBUS_TYPE_BOOLEAN,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    bool_write_value,
    bool_read_value,
    bool_set_value,
    NULL,
    NULL, /* FIXME */
    NULL /* FIXME */
};

```

```

};

static const TestTypeNodeClass byte_class = {
    DBUS_TYPE_BYTE,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    byte_write_value,
    byte_read_value,
    byte_set_value,
    NULL,
    NULL, /* FIXME */
    NULL /* FIXME */
};

static const TestTypeNodeClass double_class = {
    DBUS_TYPE_DOUBLE,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    double_write_value,
    double_read_value,
    double_set_value,
    NULL,
    NULL, /* FIXME */
    NULL /* FIXME */
};

static const TestTypeNodeClass object_path_class = {
    DBUS_TYPE_OBJECT_PATH,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    object_path_write_value,
    object_path_read_value,
    object_path_set_value,
    NULL,
    NULL,
    NULL
};

static const TestTypeNodeClass signature_class = {
    DBUS_TYPE_SIGNATURE,
    sizeof (TestTypeNode),
    0,
    NULL,
    NULL,
    signature_write_value,
    signature_read_value,

```

```

    signature_set_value,
    NULL,
    NULL,
    NULL
};

static const TestTypeNodeClass struct_1_class = {
    DBUS_TYPE_STRUCT,
    sizeof (TestTypeNodeContainer),
    1, /* number of times children appear as fields */
    NULL,
    container_destroy,
    struct_write_value,
    struct_read_value,
    struct_set_value,
    struct_build_signature,
    NULL,
    NULL
};

static const TestTypeNodeClass struct_2_class = {
    DBUS_TYPE_STRUCT,
    sizeof (TestTypeNodeContainer),
    2, /* number of times children appear as fields */
    NULL,
    container_destroy,
    struct_write_value,
    struct_read_value,
    struct_set_value,
    struct_build_signature,
    NULL,
    NULL
};

static const TestTypeNodeClass dict_1_class = {
    DBUS_TYPE_ARRAY, /* this is correct, a dict is an array of dict
entry */
    sizeof (TestTypeNodeContainer),
    1, /* number of entries */
    NULL,
    container_destroy,
    dict_write_value,
    dict_read_value,
    dict_set_value,
    dict_build_signature,
    NULL,
    NULL
};

static dbus_bool_t arrays_write_fixed_in_blocks = FALSE;

static const TestTypeNodeClass array_0_class = {

```



```

    DBUS_TYPE_ARRAY,
    sizeof (TestTypeNodeContainer),
    0, /* number of array elements */
    NULL,
    container_destroy,
    array_write_value,
    array_read_value,
    array_set_value,
    array_build_signature,
    NULL,
    NULL
};

static const TestTypeNodeClass array_1_class = {
    DBUS_TYPE_ARRAY,
    sizeof (TestTypeNodeContainer),
    1, /* number of array elements */
    NULL,
    container_destroy,
    array_write_value,
    array_read_value,
    array_set_value,
    array_build_signature,
    NULL,
    NULL
};

static const TestTypeNodeClass array_2_class = {
    DBUS_TYPE_ARRAY,
    sizeof (TestTypeNodeContainer),
    2, /* number of array elements */
    NULL,
    container_destroy,
    array_write_value,
    array_read_value,
    array_set_value,
    array_build_signature,
    NULL,
    NULL
};

static const TestTypeNodeClass array_9_class = {
    DBUS_TYPE_ARRAY,
    sizeof (TestTypeNodeContainer),
    9, /* number of array elements */
    NULL,
    container_destroy,
    array_write_value,
    array_read_value,
    array_set_value,
    array_build_signature,
    NULL,

```

```

    NULL
};

static const TestTypeNodeClass variant_class = {
    DBUS_TYPE_VARIANT,
    sizeof (TestTypeNodeContainer),
    0,
    NULL,
    container_destroy,
    variant_write_value,
    variant_read_value,
    variant_set_value,
    NULL,
    NULL,
    NULL
};

static const TestTypeNodeClass* const
basic_nodes[] = {
    &int16_class,
    &uint16_class,
    &int32_class,
    &uint32_class,
    &int64_class,
    &uint64_class,
    &bool_class,
    &byte_class,
    &double_class,
    &string_0_class,
    &string_1_class,
    &string_3_class,
    &string_8_class,
    &object_path_class,
    &signature_class
};
#define N_BASICS (_DBUS_N_ELEMENTS (basic_nodes))

static const TestTypeNodeClass* const
container_nodes[] = {
    &struct_1_class,
    &array_1_class,
    &struct_2_class,
    &array_0_class,
    &array_2_class,
    &variant_class,
    &dict_1_class /* last since we want struct and array before it */
    /* array_9_class is omitted on purpose, it's too slow;
    * we only use it in one hardcoded test below
    */
};
#define N_CONTAINERS (_DBUS_N_ELEMENTS (container_nodes))

```

```

static TestTypeNode*
node_new (const TestTypeNodeClass *klass)
{
    TestTypeNode *node;

    node = dbus_malloc0 (klass->instance_size);
    if (node == NULL)
        return NULL;

    node->klass = klass;

    if (klass->construct)
    {
        if (!(* klass->construct) (node))
        {
            dbus_free (node);
            return NULL;
        }
    }

    return node;
}

static void
node_destroy (TestTypeNode *node)
{
    if (node->klass->destroy)
        (* node->klass->destroy) (node);
    dbus_free (node);
}

static dbus_bool_t
node_write_value (TestTypeNode *node,
                  DataBlock *block,
                  DBusTypeWriter *writer,
                  int seed)
{
    dbus_bool_t retval;

    retval = (* node->klass->write_value) (node, block, writer, seed);

#ifdef 0
    /* Handy to see where things break, but too expensive to do all the
       time */
    data_block_verify (block);
#endif

    return retval;
}

static dbus_bool_t
node_read_value (TestTypeNode *node,

```

```

                DBusTypeReader *reader,
                int                seed)
{
    /* DBusTypeReader restored; */

    if (!(* node->klass->read_value) (node, reader, seed))
        return FALSE;

    return TRUE;
}

/* Warning: if this one fails due to OOM, it has side effects (can
 * modify only some of the sub-values). OK in a test suite, but we
 * never do this in real code.
 */
static dbus_bool_t
node_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int                seed)
{
    if (!(* node->klass->set_value) (node, reader, realign_root, seed))
        return FALSE;

    return TRUE;
}

static dbus_bool_t
node_build_signature (TestTypeNode *node,
                     DBusString *str)
{
    if (node->klass->build_signature)
        return (* node->klass->build_signature) (node, str);
    else
        return _dbus_string_append_byte (str, node->klass->typecode);
}

static dbus_bool_t
node_append_child (TestTypeNode *node,
                  TestTypeNode *child)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;

    _dbus_assert (node->klass->instance_size >= (int) sizeof
                  (TestTypeNodeContainer));

    if (!_dbus_list_append (&container->children, child))
        _dbus_assert_not_reached ("no memory"); /* we never check the
return value on node_append_child anyhow - it's run from outside the
malloc-failure test code */

    return TRUE;
}

```

```

}

static dbus_bool_t
node_write_multi (TestTypeNode *node,
                 DataBlock *block,
                 DBusTypeWriter *writer,
                 int seed,
                 int n_copies)
{
    dbus_bool_t retval;

    _dbus_assert (node->klass->write_multi != NULL);
    retval = (* node->klass->write_multi) (node, block, writer, seed,
n_copies);

#ifdef 0
    /* Handy to see where things break, but too expensive to do all the
time */
    data_block_verify (block);
#endif

    return retval;
}

static dbus_bool_t
node_read_multi (TestTypeNode *node,
                DBusTypeReader *reader,
                int seed,
                int n_copies)
{
    _dbus_assert (node->klass->read_multi != NULL);

    if (!(* node->klass->read_multi) (node, reader, seed, n_copies))
        return FALSE;

    return TRUE;
}

static int n_iterations_completed_total = 0;
static int n_iterations_completed_this_test = 0;
static int n_iterations_expected_this_test = 0;

typedef struct
{
    const DBusString *signature;
    DataBlock *block;
    int type_offset;
    TestTypeNode **nodes;
    int n_nodes;
} NodeIterationData;

static dbus_bool_t

```

```

run_test_copy (NodeIterationData *nid)
{
    DataBlock *src;
    DataBlock dest;
    dbus_bool_t retval;
    DBusTypeReader reader;
    DBusTypeWriter writer;

    _dbus_verbose ("\n");

    src = nid->block;

    retval = FALSE;

    if (!data_block_init (&dest, src->byte_order, src->initial_offset))
        return FALSE;

    data_block_init_reader_writer (src, &reader, NULL);
    data_block_init_reader_writer (&dest, NULL, &writer);

    /* DBusTypeWriter assumes it's writing into an existing signature,
     * so doesn't add nul on its own. We have to do that.
     */
    if (!_dbus_string_insert_byte (&dest.signature,
                                   dest.initial_offset, '\0'))
        goto out;

    if (!_dbus_type_writer_write_reader (&writer, &reader))
        goto out;

    /* Data blocks should now be identical */
    if (!_dbus_string_equal (&src->signature, &dest.signature))
    {
        _dbus_verbose ("SOURCE\n");
        _dbus_verbose_bytes_of_string (&src->signature, 0,
                                       _dbus_string_get_length (&src->signature));
        _dbus_verbose ("DEST\n");
        _dbus_verbose_bytes_of_string (&dest.signature, 0,
                                       _dbus_string_get_length
(&dest.signature));
        _dbus_assert_not_reached ("signatures did not match");
    }

    if (!_dbus_string_equal (&src->body, &dest.body))
    {
        _dbus_verbose ("SOURCE\n");
        _dbus_verbose_bytes_of_string (&src->body, 0,
                                       _dbus_string_get_length (&src->body));
        _dbus_verbose ("DEST\n");
        _dbus_verbose_bytes_of_string (&dest.body, 0,

```

```

                                _dbus_string_get_length
(&dest.body));
    _dbus_assert_not_reached ("bodies did not match");
}

retval = TRUE;

out:

data_block_free (&dest);

return retval;
}

static dbus_bool_t
run_test_values_only_write (NodeIterationData *nid)
{
    DBusTypeReader reader;
    DBusTypeWriter writer;
    int i;
    dbus_bool_t retval;
    int sig_len;

    _dbus_verbose ("\n");

    retval = FALSE;

    data_block_reset (nid->block);

    sig_len = _dbus_string_get_length (nid->signature);

    _dbus_type_writer_init_values_only (&writer,
                                        nid->block->byte_order,
                                        nid->signature, 0,
                                        &nid->block->body,
                                        _dbus_string_get_length (&nid-
>block->body) - N_FENCE_BYTES);
    _dbus_type_reader_init (&reader,
                            nid->block->byte_order,
                            nid->signature, 0,
                            &nid->block->body,
                            nid->block->initial_offset);

    i = 0;
    while (i < nid->n_nodes)
    {
        if (!node_write_value (nid->nodes[i], nid->block, &writer, i))
            goto out;

        ++i;
    }
}

```

```

/* if we wrote any typecodes then this would fail */
_dbus_assert (sig_len == _dbus_string_get_length (nid->signature));

/* But be sure we wrote out the values correctly */
i = 0;
while (i < nid->n_nodes)
{
    if (!node_read_value (nid->nodes[i], &reader, i))
        goto out;

    if (i + 1 == nid->n_nodes)
        NEXT_EXPECTING_FALSE (&reader);
    else
        NEXT_EXPECTING_TRUE (&reader);

    ++i;
}

retval = TRUE;

out:
data_block_reset (nid->block);
return retval;
}

/* offset the seed for setting, so we set different numbers than
 * we originally wrote. Don't offset by a huge number since in
 * some cases it's value = possibilities[seed % n_possibilities]
 * and we don't want to wrap around. bool_from_seed
 * is just seed % 2 even.
 */
#define SET_SEED 1
static dbus_bool_t
run_test_set_values (NodeIterationData *nid)
{
    DBusTypeReader reader;
    DBusTypeReader realign_root;
    dbus_bool_t retval;
    int i;

    _dbus_verbose ("\n");

    retval = FALSE;

    data_block_init_reader_writer (nid->block,
                                   &reader, NULL);

    realign_root = reader;

    i = 0;
    while (i < nid->n_nodes)
        {

```



```

while ((t = _dbus_type_reader_get_current_type (&reader)) !=
DBUS_TYPE_INVALID)
{
    /* Right now, deleting only works on array elements.  We delete
    * all array elements, and then verify that there aren't any
    * left.
    */
    if (t == DBUS_TYPE_ARRAY)
    {
        DBusTypeReader array;
        int n_elements;
        int elem_type;

        _dbus_type_reader_recurse (&reader, &array);
        n_elements = 0;
        while (_dbus_type_reader_get_current_type (&array) !=
DBUS_TYPE_INVALID)
        {
            n_elements += 1;
            _dbus_type_reader_next (&array);
        }

        /* reset to start of array */
        _dbus_type_reader_recurse (&reader, &array);
        _dbus_verbose ("recurring into deletion loop
reader.value_pos = %d array.value_pos = %d array.u.start_pos = %d\n",
            reader.value_pos, array.value_pos,
array.u.array.start_pos);
        while ((elem_type = _dbus_type_reader_get_current_type
(&array)) != DBUS_TYPE_INVALID)
        {
            /* We don't want to always delete from the same part of
the array. */
            static int cycle = 0;
            int elem;

            _dbus_assert (n_elements > 0);

            elem = cycle;
            if (elem == 3 || elem >= n_elements) /* end of array */
                elem = n_elements - 1;

            _dbus_verbose ("deleting array element %d of %d type %s
cycle %d reader pos %d elem pos %d\n",
                elem, n_elements, _dbus_type_to_string
(elem_type),
                cycle, reader.value_pos,
array.value_pos);
            while (elem > 0)
            {
                if (!_dbus_type_reader_next (&array))

```

```

        _dbus_assert_not_reached ("should have had another
element\n");
        --elem;
    }

    if (!_dbus_type_reader_delete (&array, &reader))
        goto out;

    n_elements -= 1;

    /* reset */
    _dbus_type_reader_recurse (&reader, &array);

    if (cycle > 2)
        cycle = 0;
    else
        cycle += 1;
    }
}
_dbus_type_reader_next (&reader);
}

/* Check that there are no array elements left */
data_block_init_reader_writer (nid->block,
                               &reader, NULL);

while ((t = _dbus_type_reader_get_current_type (&reader)) !=
DBUS_TYPE_INVALID)
{
    _dbus_type_reader_next (&reader);
}

retval = TRUE;

out:
return retval;
}

static dbus_bool_t
run_test_nodes_iteration (void *data)
{
    NodeIterationData *nid = data;
    DBusTypeReader reader;
    DBusTypeWriter writer;
    int i;
    dbus_bool_t retval;

    /* Stuff to do:
    * 1. write the value
    * 2. strcmp-compare with the signature we built
    * 3. read the value

```

```

    * 4. type-iterate the signature and the value and see if they are
the same type-wise
    */
    retval = FALSE;

    data_block_init_reader_writer (nid->block,
                                   &reader, &writer);

    /* DBusTypeWriter assumes it's writing into an existing signature,
    * so doesn't add nul on its own. We have to do that.
    */
    if (!_dbus_string_insert_byte (&nid->block->signature,
                                   nid->type_offset, '\0'))

        goto out;

    i = 0;
    while (i < nid->n_nodes)
    {
        if (!node_write_value (nid->nodes[i], nid->block, &writer, i))
            goto out;

        ++i;
    }

    if (!_dbus_string_equal_substring (nid->signature, 0,
        _dbus_string_get_length (nid->signature),
                                   &nid->block->signature, nid-
>type_offset))
    {
        _dbus_warn ("Expected signature '%s' and got '%s' with initial
offset %d\n",
                   _dbus_string_get_const_data (nid->signature),
                   _dbus_string_get_const_data_len (&nid->block-
>signature, nid->type_offset, 0),
                   nid->type_offset);
        _dbus_assert_not_reached ("wrong signature");
    }

    i = 0;
    while (i < nid->n_nodes)
    {
        if (!node_read_value (nid->nodes[i], &reader, i))
            goto out;

        if (i + 1 == nid->n_nodes)
            NEXT_EXPECTING_FALSE (&reader);
        else
            NEXT_EXPECTING_TRUE (&reader);

        ++i;
    }

```

```

    if (n_iterations_expected_this_test <=
MAX_ITERATIONS_FOR_EXPENSIVE_TESTS)
    {
        /* this set values test uses code from copy and
        * values_only_write so would ideally be last so you get a
        * simpler test case for problems with copying or values_only
        * writing; but it also needs an already-written DataBlock so it
        * has to go first. Comment it out if it breaks, and see if the
        * later tests also break - debug them first if so.
        */
        if (!run_test_set_values (nid))
            goto out;

        if (!run_test_delete_values (nid))
            goto out;

        if (!run_test_copy (nid))
            goto out;

        if (!run_test_values_only_write (nid))
            goto out;
    }

    /* FIXME type-iterate both signature and value and compare the
resulting
    * tree to the node tree perhaps
    */

    retval = TRUE;

out:

    data_block_reset (nid->block);

    return retval;
}

static void
run_test_nodes_in_one_configuration (TestTypeNode    **nodes,
                                     int              n_nodes,
                                     const DBusString *signature,
                                     int              byte_order,
                                     int              initial_offset)
{
    DataBlock block;
    NodeIterationData nid;

    if (!data_block_init (&block, byte_order, initial_offset))
        _dbus_assert_not_reached ("no memory");

    nid.signature = signature;
    nid.block = &block;
}

```

```

nid.type_offset = initial_offset;
nid.nodes = nodes;
nid.n_nodes = n_nodes;

if (TEST_OOM_HANDLING &&
    n_iterations_expected_this_test <=
MAX_ITERATIONS_FOR_EXPENSIVE_TESTS)
{
    _dbus_test_oom_handling ("running test node",
                            run_test_nodes_iteration,
                            &nid);
}
else
{
    if (!run_test_nodes_iteration (&nid))
        _dbus_assert_not_reached ("no memory");
}

data_block_free (&block);
}

static void
run_test_nodes (TestTypeNode **nodes,
                int n_nodes)
{
    int i;
    DBusString signature;

    if (!_dbus_string_init (&signature))
        _dbus_assert_not_reached ("no memory");

    i = 0;
    while (i < n_nodes)
    {
        if (!node_build_signature (nodes[i], &signature))
            _dbus_assert_not_reached ("no memory");

        ++i;
    }

    _dbus_verbose (">>> test nodes with signature '%s'\n",
                  _dbus_string_get_const_data (&signature));

    i = 0;
    while (i <= MAX_INITIAL_OFFSET)
    {
        run_test_nodes_in_one_configuration (nodes, n_nodes, &signature,
                                            DBUS_LITTLE_ENDIAN, i);
        run_test_nodes_in_one_configuration (nodes, n_nodes, &signature,
                                            DBUS_BIG_ENDIAN, i);

        ++i;
    }
}

```

```

    }

    n_iterations_completed_this_test += 1;
    n_iterations_completed_total += 1;

    if (n_iterations_completed_this_test ==
        n_iterations_expected_this_test)
    {
        fprintf (stderr, " 100%% %d this test (%d cumulative)\n",
                 n_iterations_completed_this_test,
                 n_iterations_completed_total);
    }
    /* this happens to turn out well with mod == 1 */
    else if ((n_iterations_completed_this_test %
              (int)(n_iterations_expected_this_test / 10.0)) == 1)
    {
        fprintf (stderr, " %d%% ", (int)
                 (n_iterations_completed_this_test / (double)
                  n_iterations_expected_this_test * 100));
    }

    _dbus_string_free (&signature);
}

#define N_VALUES (N_BASICS * N_CONTAINERS + N_BASICS)

static TestTypeNode*
value_generator (int *ip)
{
    int i = *ip;
    const TestTypeNodeClass *child_class;
    const TestTypeNodeClass *container_class;
    TestTypeNode *child;
    TestTypeNode *node;

    _dbus_assert (i <= N_VALUES);

    if (i == N_VALUES)
    {
        return NULL;
    }
    else if (i < N_BASICS)
    {
        node = node_new (basic_nodes[i]);
    }
    else
    {
        /* imagine an array:
         * container 0 of basic 0
         * container 0 of basic 1
         * container 0 of basic 2
         * container 1 of basic 0

```

```

        * container 1 of basic 1
        * container 1 of basic 2
        */
    i -= N_BASICS;

    container_klass = container_nodes[i / N_BASICS];
    child_klass = basic_nodes[i % N_BASICS];

    node = node_new (container_klass);
    child = node_new (child_klass);

    node_append_child (node, child);
}

*ip += 1; /* increment the generator */

return node;
}

static void
build_body (TestTypeNode **nodes,
            int          n_nodes,
            int          byte_order,
            DBusString  *signature,
            DBusString  *body)
{
    int i;
    DataBlock block;
    DBusTypeReader reader;
    DBusTypeWriter writer;

    i = 0;
    while (i < n_nodes)
    {
        if (! node_build_signature (nodes[i], signature))
            _dbus_assert_not_reached ("no memory");

        ++i;
    }

    if (!data_block_init (&block, byte_order, 0))
        _dbus_assert_not_reached ("no memory");

    data_block_init_reader_writer (&block,
                                   &reader, &writer);

    /* DBusTypeWriter assumes it's writing into an existing signature,
     * so doesn't add nul on its own. We have to do that.
     */
    if (!_dbus_string_insert_byte (&block.signature,
                                   0, '\0'))
        _dbus_assert_not_reached ("no memory");
}

```



```

i = 0;
while (i < n_nodes)
{
    if (!node_write_value (nodes[i], &block, &writer, i))
        _dbus_assert_not_reached ("no memory");

    ++i;
}

if (!_dbus_string_copy_len (&block.body, 0,
                            _dbus_string_get_length (&block.body) -
N_FENCE_BYTES,
                            body, 0))
    _dbus_assert_not_reached ("oom");

data_block_free (&block);
}

dbus_bool_t
dbus_internal_do_not_use_generate_bodies (int          sequence,
                                           int          byte_order,
                                           DBusString *signature,
                                           DBusString *body)
{
    TestTypeNode *nodes[1];
    int i;
    int n_nodes;

    nodes[0] = value_generator (&sequence);

    if (nodes[0] == NULL)
        return FALSE;

    n_nodes = 1;

    build_body (nodes, n_nodes, byte_order, signature, body);

    i = 0;
    while (i < n_nodes)
    {
        node_destroy (nodes[i]);
        ++i;
    }

    return TRUE;
}

static void
make_and_run_values_inside_container (const TestTypeNodeClass
*container_klass,

```

```

                                                    int
n_nested)
{
    TestTypeNode *root;
    TestTypeNode *container;
    TestTypeNode *child;
    int i;

    root = node_new (container_class);
    container = root;
    for (i = 1; i < n_nested; i++)
        {
            child = node_new (container_class);
            node_append_child (container, child);
            container = child;
        }

    /* container should now be the most-nested container */

    i = 0;
    while ((child = value_generator (&i))
        {
            node_append_child (container, child);

            run_test_nodes (&root, 1);

            _dbus_list_clear (&((TestTypeNodeContainer*)container)-
>children);
            node_destroy (child);
        }

    node_destroy (root);
}

static void
start_next_test (const char *format,
                 int         expected)
{
    n_iterations_completed_this_test = 0;
    n_iterations_expected_this_test = expected;

    fprintf (stderr, ">>> >>> ");
    fprintf (stderr, format,
            n_iterations_expected_this_test);
}

static void
make_and_run_test_nodes (void)
{
    int i, j, k, m;

    /* We try to do this in order of "complicatedness" so that test

```

```

* failures tend to show up in the simplest test case that
* demonstrates the failure.  There are also some tests that run
* more than once for this reason, first while going through simple
* cases, second while going through a broader range of complex
* cases.
*/
/* Each basic node.  The basic nodes should include:
*
* - each fixed-size type (in such a way that it has different
values each time,
*                               so we can tell if we mix two of them up)
* - strings of various lengths
* - object path
* - signature
*/
/* Each container node.  The container nodes should include:
*
* struct with 1 and 2 copies of the contained item
* array with 0, 1, 2 copies of the contained item
* variant
*/
/* Let a "value" be a basic node, or a container containing a
single basic node.
* Let n_values be the number of such values i.e. (n_container *
n_basic + n_basic)
* When iterating through all values to make combinations, do the
basic types
* first and the containers second.
*/
/* Each item is shown with its number of iterations to complete so
* we can keep a handle on this unit test
*/

/* FIXME test just an empty body, no types at all */

start_next_test ("Each value by itself %d iterations\n", N_VALUES);
{
    TestTypeNode *node;
    i = 0;
    while ((node = value_generator (&i))
        {
            run_test_nodes (&node, 1);

            node_destroy (node);
        }
}

start_next_test ("Each value by itself with arrays as blocks %d
iterations\n", N_VALUES);
arrays_write_fixed_in_blocks = TRUE;
{
    TestTypeNode *node;

```

```

    i = 0;
    while ((node = value_generator (&i))
        {
            run_test_nodes (&node, 1);

            node_destroy (node);
        }
    }
arrays_write_fixed_in_blocks = FALSE;

start_next_test ("All values in one big toplevel %d iteration\n",
1);
{
    TestTypeNode *nodes[N_VALUES];

    i = 0;
    while ((nodes[i] = value_generator (&i))
        ;

        run_test_nodes (nodes, N_VALUES);

        for (i = 0; i < N_VALUES; i++)
            node_destroy (nodes[i]);
    }

    start_next_test ("Each value,value pair combination as toplevel, in
both orders %d iterations\n",
                    N_VALUES * N_VALUES);
    {
        TestTypeNode *nodes[2];

        i = 0;
        while ((nodes[0] = value_generator (&i))
            {
                j = 0;
                while ((nodes[1] = value_generator (&j))
                    {
                        run_test_nodes (nodes, 2);

                        node_destroy (nodes[1]);
                    }

                node_destroy (nodes[0]);
            }
        }

        start_next_test ("Each container containing each value %d
iterations\n",
                        N_CONTAINERS * N_VALUES);
        for (i = 0; i < N_CONTAINERS; i++)
            {
                const TestTypeNodeClass *container_klass = container_nodes[i];

```

```

    make_and_run_values_inside_container (container_klass, 1);
}

start_next_test ("Each container containing each value with arrays
as blocks %d iterations\n",
                N_CONTAINERS * N_VALUES);
arrays_write_fixed_in_blocks = TRUE;
for (i = 0; i < N_CONTAINERS; i++)
{
    const TestTypeNodeClass *container_klass = container_nodes[i];

    make_and_run_values_inside_container (container_klass, 1);
}
arrays_write_fixed_in_blocks = FALSE;

start_next_test ("Each container of same container of each value %d
iterations\n",
                N_CONTAINERS * N_VALUES);
for (i = 0; i < N_CONTAINERS; i++)
{
    const TestTypeNodeClass *container_klass = container_nodes[i];

    make_and_run_values_inside_container (container_klass, 2);
}

start_next_test ("Each container of same container of same container
of each value %d iterations\n",
                N_CONTAINERS * N_VALUES);
for (i = 0; i < N_CONTAINERS; i++)
{
    const TestTypeNodeClass *container_klass = container_nodes[i];

    make_and_run_values_inside_container (container_klass, 3);
}

start_next_test ("Each value,value pair inside a struct %d
iterations\n",
                N_VALUES * N_VALUES);
{
    TestTypeNode *val1, *val2;
    TestTypeNode *node;

    node = node_new (&struct_1_class);

    i = 0;
    while ((val1 = value_generator (&i))
           {
                j = 0;
                while ((val2 = value_generator (&j))
                       {

```

```

        TestTypeNodeContainer *container =
(TestTypeNodeContainer*) node;

        node_append_child (node, val1);
        node_append_child (node, val2);

        run_test_nodes (&node, 1);

        _dbus_list_clear (&container->children);
        node_destroy (val2);
    }
    node_destroy (val1);
}
node_destroy (node);
}

start_next_test ("All values in one big struct %d iteration\n",
                1);
{
    TestTypeNode *node;
    TestTypeNode *child;

    node = node_new (&struct_1_class);

    i = 0;
    while ((child = value_generator (&i))
           node_append_child (node, child);

    run_test_nodes (&node, 1);

    node_destroy (node);
}

start_next_test ("Each value in a large array %d iterations\n",
                N_VALUES);
{
    TestTypeNode *val;
    TestTypeNode *node;

    node = node_new (&array_9_class);

    i = 0;
    while ((val = value_generator (&i))
           {
                TestTypeNodeContainer *container = (TestTypeNodeContainer*)
node;

                node_append_child (node, val);

                run_test_nodes (&node, 1);

                _dbus_list_clear (&container->children);

```

```

        node_destroy (val);
    }

    node_destroy (node);
}

if (_dbus_getenv ("DBUS_TEST_SLOW") == NULL ||
    atoi (_dbus_getenv ("DBUS_TEST_SLOW")) < 1)
{
    fprintf (stderr, "skipping remaining marshal-recursive tests, "
            "run with DBUS_TEST_SLOW=1 (or more) to enable\n");
    goto out;
}

start_next_test ("Each container of each container of each value %d
iterations\n",
                N_CONTAINERS * N_CONTAINERS * N_VALUES);
for (i = 0; i < N_CONTAINERS; i++)
{
    const TestTypeNodeClass *outer_container_class =
container_nodes[i];
    TestTypeNode *outer_container = node_new
(outer_container_class);

    for (j = 0; j < N_CONTAINERS; j++)
    {
        TestTypeNode *child;
        const TestTypeNodeClass *inner_container_class =
container_nodes[j];
        TestTypeNode *inner_container = node_new
(inner_container_class);

        node_append_child (outer_container, inner_container);

        m = 0;
        while ((child = value_generator (&m))
            {
                node_append_child (inner_container, child);

                run_test_nodes (&outer_container, 1);

                _dbus_list_clear
(&((TestTypeNodeContainer*)inner_container)->children);
                node_destroy (child);
            }
            _dbus_list_clear
(&((TestTypeNodeContainer*)outer_container)->children);
            node_destroy (inner_container);
        }
        node_destroy (outer_container);
    }
}

```

```

    start_next_test ("Each container of each container of each container
of each value %d iterations\n",
                    N_CONTAINERS * N_CONTAINERS * N_CONTAINERS *
N_VALUES);
    for (i = 0; i < N_CONTAINERS; i++)
    {
        const TestTypeNodeClass *outer_container_klass =
container_nodes[i];
        TestTypeNode *outer_container = node_new
(outer_container_klass);

        for (j = 0; j < N_CONTAINERS; j++)
        {
            const TestTypeNodeClass *inner_container_klass =
container_nodes[j];
            TestTypeNode *inner_container = node_new
(inner_container_klass);

            node_append_child (outer_container, inner_container);

            for (k = 0; k < N_CONTAINERS; k++)
            {
                TestTypeNode *child;
                const TestTypeNodeClass *center_container_klass =
container_nodes[k];
                TestTypeNode *center_container = node_new
(center_container_klass);

                node_append_child (inner_container, center_container);

                m = 0;
                while ((child = value_generator (&m)))
                {
                    node_append_child (center_container, child);

                    run_test_nodes (&outer_container, 1);

                    _dbus_list_clear
(&((TestTypeNodeContainer*)center_container)->children);
                    node_destroy (child);
                }
                _dbus_list_clear
(&((TestTypeNodeContainer*)inner_container)->children);
                node_destroy (center_container);
            }
            _dbus_list_clear
(&((TestTypeNodeContainer*)outer_container)->children);
            node_destroy (inner_container);
        }
        node_destroy (outer_container);
    }

```



```

/* This one takes a really long time (10 minutes on a Core2), so
only enable
* it if you're really sure */
if (atoi (_dbus_getenv ("DBUS_TEST_SLOW")) < 2)
{
    fprintf (stderr, "skipping really slow marshal-recursive test, "
            "run with DBUS_TEST_SLOW=2 (or more) to enable\n");
    goto out;
}

start_next_test ("Each value,value,value triplet combination as
toplevel, in all orders %d iterations\n",
                N_VALUES * N_VALUES * N_VALUES);
{
    TestTypeNode *nodes[3];

    i = 0;
    while ((nodes[0] = value_generator (&i))
        {
            j = 0;
            while ((nodes[1] = value_generator (&j))
                {
                    k = 0;
                    while ((nodes[2] = value_generator (&k))
                        {
                            run_test_nodes (nodes, 3);

                            node_destroy (nodes[2]);
                        }
                    node_destroy (nodes[1]);
                }
            node_destroy (nodes[0]);
        }
    }

out:
    fprintf (stderr, "%d total iterations of recursive marshaling
tests\n",
            n_iterations_completed_total);
    fprintf (stderr, "each iteration ran at initial offsets 0 through %d
in both big and little endian\n",
            MAX_INITIAL_OFFSET);
    fprintf (stderr, "out of memory handling %s tested\n",
            TEST_OOM_HANDLING ? "was" : "was not");
}

dbus_bool_t
_dbus_marshal_recursive_test (void)
{
    make_and_run_test_nodes ();

    return TRUE;
}

```

```

}

/*
 *
 *
 *       Implementations of each type node class
 *
 *
 *
 */
#define MAX_MULTI_COUNT 5

#define SAMPLE_INT16          1234
#define SAMPLE_INT16_ALTERNATE 6785
static dbus_int16_t
int16_from_seed (int seed)
{
    /* Generate an integer value that's predictable from seed.  We could
     * just use seed itself, but that would only ever touch one byte of
     * the int so would miss some kinds of bug.
     */
    dbus_int16_t v;

    v = 42; /* just to quiet compiler afaik */
    switch (seed % 5)
    {
        case 0:
            v = SAMPLE_INT16;
            break;
        case 1:
            v = SAMPLE_INT16_ALTERNATE;
            break;
        case 2:
            v = -1;
            break;
        case 3:
            v = _DBUS_INT16_MAX;
            break;
        case 4:
            v = 1;
            break;
    }

    if (seed > 1)
        v *= seed; /* wraps around eventually, which is fine */

    return v;
}

static dbus_bool_t
int16_write_value (TestTypeNode *node,
                  DataBlock *block,

```

```

        DBusTypeWriter *writer,
        int             seed)
{
    /* also used for uint16 */
    dbus_int16_t v;

    v = int16_from_seed (seed);

    return _dbus_type_writer_write_basic (writer,
                                           node->klass->typecode,
                                           &v);
}

static dbus_bool_t
int16_read_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 int             seed)
{
    /* also used for uint16 */
    dbus_int16_t v;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                   (dbus_int16_t*) &v);

    _dbus_assert (v == int16_from_seed (seed));

    return TRUE;
}

static dbus_bool_t
int16_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int             seed)
{
    /* also used for uint16 */
    dbus_int16_t v;

    v = int16_from_seed (seed);

    return _dbus_type_reader_set_basic (reader,
                                        &v,
                                        realign_root);
}

static dbus_bool_t
int16_write_multi (TestTypeNode *node,
                  DataBlock      *block,
                  DBusTypeWriter *writer,
                  int             seed,

```

```

                int                count)
{
    /* also used for uint16 */
    dbus_int16_t values[MAX_MULTI_COUNT];
    dbus_int16_t *v_ARRAY_INT16 = values;
    int i;

    for (i = 0; i < count; ++i)
        values[i] = int16_from_seed (seed + i);

    return _dbus_type_writer_write_fixed_multi (writer,
                                                node->klass->typecode,
                                                &v_ARRAY_INT16, count);
}

static dbus_bool_t
int16_read_multi (TestTypeNode *node,
                 DBusTypeReader *reader,
                 int seed,
                 int count)
{
    /* also used for uint16 */
    dbus_int16_t *values;
    int n_elements;
    int i;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_fixed_multi (reader,
                                        &values,
                                        &n_elements);

    if (n_elements != count)
        _dbus_warn ("got %d elements expected %d\n", n_elements, count);
    _dbus_assert (n_elements == count);

    for (i = 0; i < count; i++)
        _dbus_assert (((dbus_int16_t)_dbus_unpack_uint16 (reader->byte_order,
                                                         (const unsigned
char*)values + (i * 2))) ==
                    int16_from_seed (seed + i));

    return TRUE;
}

#define SAMPLE_INT32                12345678
#define SAMPLE_INT32_ALTERNATE 53781429
static dbus_int32_t
int32_from_seed (int seed)
{

```

```

/* Generate an integer value that's predictable from seed. We could
 * just use seed itself, but that would only ever touch one byte of
 * the int so would miss some kinds of bug.
 */
dbus_int32_t v;

v = 42; /* just to quiet compiler afaik */
switch (seed % 5)
{
    case 0:
        v = SAMPLE_INT32;
        break;
    case 1:
        v = SAMPLE_INT32_ALTERNATE;
        break;
    case 2:
        v = -1;
        break;
    case 3:
        v = _DBUS_INT_MAX;
        break;
    case 4:
        v = 1;
        break;
}

if (seed > 1)
    v *= seed; /* wraps around eventually, which is fine */

return v;
}

static dbus_bool_t
int32_write_value (TestTypeNode *node,
                  DataBlock *block,
                  DBusTypeWriter *writer,
                  int seed)
{
    /* also used for uint32 */
    dbus_int32_t v;

    v = int32_from_seed (seed);

    return _dbus_type_writer_write_basic (writer,
                                          node->klass->typecode,
                                          &v);
}

static dbus_bool_t
int32_read_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 int seed)

```

```

{
    /* also used for uint32 */
    dbus_int32_t v;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (dbus_int32_t*) &v);

    _dbus_assert (v == int32_from_seed (seed));

    return TRUE;
}

static dbus_bool_t
int32_set_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 DBusTypeReader *realign_root,
                 int seed)
{
    /* also used for uint32 */
    dbus_int32_t v;

    v = int32_from_seed (seed);

    return _dbus_type_reader_set_basic (reader,
                                        &v,
                                        realign_root);
}

static dbus_bool_t
int32_write_multi (TestTypeNode *node,
                   DataBlock *block,
                   DBusTypeWriter *writer,
                   int seed,
                   int count)
{
    /* also used for uint32 */
    dbus_int32_t values[MAX_MULTI_COUNT];
    dbus_int32_t *v_ARRAY_INT32 = values;
    int i;

    for (i = 0; i < count; ++i)
        values[i] = int32_from_seed (seed + i);

    return _dbus_type_writer_write_fixed_multi (writer,
                                                node->klass->typecode,
                                                &v_ARRAY_INT32, count);
}

static dbus_bool_t
int32_read_multi (TestTypeNode *node,

```

```

        DBusTypeReader *reader,
        int             seed,
        int             count)
{
    /* also used for uint32 */
    dbus_int32_t *values;
    int n_elements;
    int i;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_fixed_multi (reader,
                                        &values,
                                        &n_elements);

    if (n_elements != count)
        _dbus_warn ("got %d elements expected %d\n", n_elements, count);
    _dbus_assert (n_elements == count);

    for (i = 0; i < count; i++)
        _dbus_assert (((int)_dbus_unpack_uint32 (reader->byte_order,
                                                (const unsigned
char*)values + (i * 4))) ==
                    int32_from_seed (seed + i));

    return TRUE;
}

#ifdef DBUS_HAVE_INT64
static dbus_int64_t
int64_from_seed (int seed)
{
    dbus_int32_t v32;
    dbus_int64_t v;

    v32 = int32_from_seed (seed);

    v = - (dbus_int32_t) ~ v32;
    v |= (((dbus_int64_t)v32) << 32);

    return v;
}
#endif

static dbus_bool_t
int64_write_value (TestTypeNode *node,
                  DataBlock *block,
                  DBusTypeWriter *writer,
                  int seed)
{
#ifdef DBUS_HAVE_INT64
    /* also used for uint64 */

```

```

dbus_int64_t v;

v = int64_from_seed (seed);

return _dbus_type_writer_write_basic (writer,
                                      node->klass->typecode,
                                      &v);

#else
return TRUE;
#endif
}

static dbus_bool_t
int64_read_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 int seed)
{
#ifdef DBUS_HAVE_INT64
/* also used for uint64 */
dbus_int64_t v;

check_expected_type (reader, node->klass->typecode);

_dbus_type_reader_read_basic (reader,
                              (dbus_int64_t*) &v);

_dbus_assert (v == int64_from_seed (seed));

return TRUE;
#else
return TRUE;
#endif
}

static dbus_bool_t
int64_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int seed)
{
#ifdef DBUS_HAVE_INT64
/* also used for uint64 */
dbus_int64_t v;

v = int64_from_seed (seed);

return _dbus_type_reader_set_basic (reader,
                                    &v,
                                    realign_root);
#else
return TRUE;
#endif
}

```



```

}

#define MAX_SAMPLE_STRING_LEN 10
static void
string_from_seed (char *buf,
                  int    len,
                  int    seed)
{
    int i;
    unsigned char v;

    _dbus_assert (len < MAX_SAMPLE_STRING_LEN);

    /* vary the length slightly, though we also have multiple string
     * value types for this, varying it here tests the set_value code
     */
    switch (seed % 3)
    {
        case 1:
            len += 2;
            break;
        case 2:
            len -= 2;
            break;
    }
    if (len < 0)
        len = 0;

    v = (unsigned char) ('A' + seed);

    i = 0;
    while (i < len)
    {
        if (v < 'A' || v > 'z')
            v = 'A';

        buf[i] = v;

        v += 1;
        ++i;
    }

    buf[i] = '\\0';
}

static dbus_bool_t
string_write_value (TestTypeNode *node,
                   DataBlock     *block,
                   DBusTypeWriter *writer,
                   int            seed)
{
    char buf[MAX_SAMPLE_STRING_LEN + 1] = "";

```

```

const char *v_string = buf;

string_from_seed (buf, node->klass->subclass_detail,
                 seed);

return _dbus_type_writer_write_basic (writer,
                                     node->klass->typecode,
                                     &v_string);
}

static dbus_bool_t
string_read_value (TestTypeNode *node,
                  DBusTypeReader *reader,
                  int seed)
{
    const char *v;
    char buf[MAX_SAMPLE_STRING_LEN + 1];
    v = buf;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (const char **) &v);

    string_from_seed (buf, node->klass->subclass_detail,
                     seed);

    if (strcmp (buf, v) != 0)
        {
            _dbus_warn ("read string '%s' expected '%s'\n",
                       v, buf);
            _dbus_assert_not_reached ("test failed");
        }

    return TRUE;
}

static dbus_bool_t
string_set_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 DBusTypeReader *realign_root,
                 int seed)
{
    char buf[MAX_SAMPLE_STRING_LEN + 1];
    const char *v_string = buf;

    string_from_seed (buf, node->klass->subclass_detail,
                     seed);

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    {

```

```

    const char *old;
    _dbus_type_reader_read_basic (reader, &old);
    _dbus_verbose ("SETTING new string '%s' len %d in place of '%s' len
%d\n",
                  v_string, strlen (v_string), old, strlen (old));
}
#endif

return _dbus_type_reader_set_basic (reader,
                                    &v_string,
                                    realign_root);
}

#define BOOL_FROM_SEED(seed) ((dbus_bool_t)((seed) % 2))

static dbus_bool_t
bool_write_value (TestTypeNode *node,
                 DataBlock *block,
                 DBusTypeWriter *writer,
                 int seed)
{
    dbus_bool_t v;

    v = BOOL_FROM_SEED (seed);

    return _dbus_type_writer_write_basic (writer,
                                          node->klass->typecode,
                                          &v);
}

static dbus_bool_t
bool_read_value (TestTypeNode *node,
                DBusTypeReader *reader,
                int seed)
{
    dbus_bool_t v;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (unsigned char*) &v);

    _dbus_assert (v == BOOL_FROM_SEED (seed));

    return TRUE;
}

static dbus_bool_t
bool_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int seed)

```

```

{
    dbus_bool_t v;

    v = BOOL_FROM_SEED (seed);

    return _dbus_type_reader_set_basic (reader,
                                       &v,
                                       realign_root);
}

#define BYTE_FROM_SEED(seed) ((unsigned char) int32_from_seed (seed))

static dbus_bool_t
byte_write_value (TestTypeNode *node,
                 DataBlock *block,
                 DBusTypeWriter *writer,
                 int seed)
{
    unsigned char v;

    v = BYTE_FROM_SEED (seed);

    return _dbus_type_writer_write_basic (writer,
                                         node->klass->typecode,
                                         &v);
}

static dbus_bool_t
byte_read_value (TestTypeNode *node,
                DBusTypeReader *reader,
                int seed)
{
    unsigned char v;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (unsigned char*) &v);

    _dbus_assert (v == BYTE_FROM_SEED (seed));

    return TRUE;
}

static dbus_bool_t
byte_set_value (TestTypeNode *node,
               DBusTypeReader *reader,
               DBusTypeReader *realign_root,
               int seed)
{
    unsigned char v;

```

```

v = BYTE_FROM_SEED (seed);

return _dbus_type_reader_set_basic (reader,
                                   &v,
                                   realign_root);
}

static double
double_from_seed (int seed)
{
    return SAMPLE_INT32 * (double) seed + 0.3;
}

static dbus_bool_t
double_write_value (TestTypeNode *node,
                   DataBlock *block,
                   DBusTypeWriter *writer,
                   int seed)
{
    double v;

    v = double_from_seed (seed);

    return _dbus_type_writer_write_basic (writer,
                                         node->klass->typecode,
                                         &v);
}

static dbus_bool_t
double_read_value (TestTypeNode *node,
                  DBusTypeReader *reader,
                  int seed)
{
    double v;
    double expected;

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (double*) &v);

    expected = double_from_seed (seed);

    if (!_DBUS_DOUBLES_BITWISE_EQUAL (v, expected))
    {
#ifdef DBUS_INT64_PRINTF_MODIFIER
        _dbus_warn ("Expected double %g got %g\n bits = 0x%"
                   DBUS_INT64_PRINTF_MODIFIER "x vs.\n bits = 0x%"
                   DBUS_INT64_PRINTF_MODIFIER "x)\n",
                   expected, v,
                   *(dbus_uint64_t*) (char*)&expected,

```

```

        *(dbus_uint64_t*) (char*)&v);
#endif
    _dbus_assert_not_reached ("test failed");
}

return TRUE;
}

static dbus_bool_t
double_set_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 DBusTypeReader *realign_root,
                 int seed)
{
    double v;

    v = double_from_seed (seed);

    return _dbus_type_reader_set_basic (reader,
                                       &v,
                                       realign_root);
}

#define MAX_SAMPLE_OBJECT_PATH_LEN 10
static void
object_path_from_seed (char *buf,
                      int seed)
{
    int i;
    unsigned char v;
    int len;

    len = seed % 9;
    _dbus_assert (len < MAX_SAMPLE_OBJECT_PATH_LEN);

    v = (unsigned char) ('A' + seed);

    if (len < 2)
    {
        buf[0] = '/';
        i = 1;
    }
    else
    {
        i = 0;
        while (i + 1 < len)
        {
            if (v < 'A' || v > 'z')
                v = 'A';

            buf[i] = '/';
            ++i;
        }
    }
}

```

```

        buf[i] = v;
        ++i;

        v += 1;
    }
}

buf[i] = '\\0';
}

static dbus_bool_t
object_path_write_value (TestTypeNode *node,
                        DataBlock *block,
                        DBusTypeWriter *writer,
                        int seed)
{
    char buf[MAX_SAMPLE_OBJECT_PATH_LEN + 1];
    const char *v_string = buf;

    object_path_from_seed (buf, seed);

    return _dbus_type_writer_write_basic (writer,
                                         node->klass->typecode,
                                         &v_string);
}

static dbus_bool_t
object_path_read_value (TestTypeNode *node,
                       DBusTypeReader *reader,
                       int seed)
{
    const char *v;
    char buf[MAX_SAMPLE_OBJECT_PATH_LEN + 1];

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (const char **) &v);

    object_path_from_seed (buf, seed);

    if (strcmp (buf, v) != 0)
    {
        _dbus_warn ("read object path '%s' expected '%s'\n",
                    v, buf);
        _dbus_assert_not_reached ("test failed");
    }

    return TRUE;
}

static dbus_bool_t

```

```

object_path_set_value (TestTypeNode *node,
                      DBusTypeReader *reader,
                      DBusTypeReader *realign_root,
                      int seed)
{
    char buf[MAX_SAMPLE_OBJECT_PATH_LEN + 1];
    const char *v_string = buf;

    object_path_from_seed (buf, seed);

    return _dbus_type_reader_set_basic (reader,
                                       &v_string,
                                       realign_root);
}

#define MAX_SAMPLE_SIGNATURE_LEN 10
static void
signature_from_seed (char *buf,
                    int seed)
{
    /* try to avoid ascending, descending, or alternating length to help
    find bugs */
    const char *sample_signatures[] = {
        "asax"
        "",
        "asau(xxxx)",
        "x",
        "ai",
        "a(ii)"
    };
};

    strcpy (buf, sample_signatures[seed %
_DBUS_N_ELEMENTS(sample_signatures)]);
}

static dbus_bool_t
signature_write_value (TestTypeNode *node,
                      DataBlock *block,
                      DBusTypeWriter *writer,
                      int seed)
{
    char buf[MAX_SAMPLE_SIGNATURE_LEN + 1];
    const char *v_string = buf;

    signature_from_seed (buf, seed);

    return _dbus_type_writer_write_basic (writer,
                                         node->klass->typecode,
                                         &v_string);
}

static dbus_bool_t

```



```

signature_read_value (TestTypeNode *node,
                    DBusTypeReader *reader,
                    int seed)
{
    const char *v;
    char buf[MAX_SAMPLE_SIGNATURE_LEN + 1];

    check_expected_type (reader, node->klass->typecode);

    _dbus_type_reader_read_basic (reader,
                                  (const char **) &v);

    signature_from_seed (buf, seed);

    if (strcmp (buf, v) != 0)
    {
        _dbus_warn ("read signature value '%s' expected '%s'\n",
                    v, buf);
        _dbus_assert_not_reached ("test failed");
    }

    return TRUE;
}

static dbus_bool_t
signature_set_value (TestTypeNode *node,
                   DBusTypeReader *reader,
                   DBusTypeReader *realign_root,
                   int seed)
{
    char buf[MAX_SAMPLE_SIGNATURE_LEN + 1];
    const char *v_string = buf;

    signature_from_seed (buf, seed);

    return _dbus_type_reader_set_basic (reader,
                                        &v_string,
                                        realign_root);
}

static dbus_bool_t
struct_write_value (TestTypeNode *node,
                  DataBlock *block,
                  DBusTypeWriter *writer,
                  int seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DataBlockState saved;
    DBusTypeWriter sub;
    int i;
    int n_copies;

```

```

n_copies = node->klass->subclass_detail;

_dbus_assert (container->children != NULL);

data_block_save (block, &saved);

if (!_dbus_type_writer_recurse (writer, DBUS_TYPE_STRUCT,
                                NULL, 0,
                                &sub))
    return FALSE;

i = 0;
while (i < n_copies)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&container->children);
    while (link != NULL)
    {
        TestTypeNode *child = link->data;
        DBusList *next = _dbus_list_get_next_link (&container-
>children, link);

        if (!node_write_value (child, block, &sub, seed + i))
        {
            data_block_restore (block, &saved);
            return FALSE;
        }

        link = next;
    }

    ++i;
}

if (!_dbus_type_writer_unrecurse (writer, &sub))
{
    data_block_restore (block, &saved);
    return FALSE;
}

return TRUE;
}

static dbus_bool_t
struct_read_or_set_value (TestTypeNode *node,
                          DBusTypeReader *reader,
                          DBusTypeReader *realign_root,
                          int seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;

```

```

DBusTypeReader sub;
int i;
int n_copies;

n_copies = node->klass->subclass_detail;

check_expected_type (reader, DBUS_TYPE_STRUCT);

_dbus_type_reader_recurse (reader, &sub);

i = 0;
while (i < n_copies)
{
    DBusList *link;

    link = _dbus_list_get_first_link (&container->children);
    while (link != NULL)
    {
        TestTypeNode *child = link->data;
        DBusList *next = _dbus_list_get_next_link (&container-
>children, link);

        if (realign_root == NULL)
        {
            if (!node_read_value (child, &sub, seed + i))
                return FALSE;
        }
        else
        {
            if (!node_set_value (child, &sub, realign_root, seed +
i))
                return FALSE;
        }

        if (i == (n_copies - 1) && next == NULL)
            NEXT_EXPECTING_FALSE (&sub);
        else
            NEXT_EXPECTING_TRUE (&sub);

        link = next;
    }

    ++i;
}

return TRUE;
}

static dbus_bool_t
struct_read_value (TestTypeNode *node,
                  DBusTypeReader *reader,
                  int seed)

```

```

{
    return struct_read_or_set_value (node, reader, NULL, seed);
}

static dbus_bool_t
struct_set_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 DBusTypeReader *realign_root,
                 int seed)
{
    return struct_read_or_set_value (node, reader, realign_root, seed);
}

static dbus_bool_t
struct_build_signature (TestTypeNode *node,
                      DBusString *str)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    int i;
    int orig_len;
    int n_copies;

    n_copies = node->klass->subclass_detail;

    orig_len = _dbus_string_get_length (str);

    if (!_dbus_string_append_byte (str, DBUS_STRUCT_BEGIN_CHAR))
        goto oom;

    i = 0;
    while (i < n_copies)
    {
        DBusList *link;

        link = _dbus_list_get_first_link (&container->children);
        while (link != NULL)
        {
            TestTypeNode *child = link->data;
            DBusList *next = _dbus_list_get_next_link (&container->children, link);

            if (!node_build_signature (child, str))
                goto oom;

            link = next;
        }

        ++i;
    }

    if (!_dbus_string_append_byte (str, DBUS_STRUCT_END_CHAR))
        goto oom;
}

```

```

    return TRUE;

oom:
    _dbus_string_set_length (str, orig_len);
    return FALSE;
}

static dbus_bool_t
array_write_value (TestTypeNode *node,
                  DataBlock *block,
                  DBusTypeWriter *writer,
                  int seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DataBlockState saved;
    DBusTypeWriter sub;
    DBusString element_signature;
    int i;
    int n_copies;
    int element_type;
    TestTypeNode *child;

    n_copies = node->klass->subclass_detail;

    _dbus_assert (container->children != NULL);

    data_block_save (block, &saved);

    if (!_dbus_string_init (&element_signature))
        return FALSE;

    child = _dbus_list_get_first (&container->children);

    if (!node_build_signature (child,
                              &element_signature))
        goto oom;

    element_type = _dbus_first_type_in_signature (&element_signature,
0);

    if (!_dbus_type_writer_recurse (writer, DBUS_TYPE_ARRAY,
                              &element_signature, 0,
                              &sub))
        goto oom;

    if (arrays_write_fixed_in_blocks &&
        dbus_type_is_fixed (element_type) &&
        child->klass->write_multi)
    {
        if (!node_write_multi (child, block, &sub, seed, n_copies))
            goto oom;
    }
}

```

```

    }
else
    {
        i = 0;
        while (i < n_copies)
            {
                DBusList *link;

                link = _dbus_list_get_first_link (&container->children);
                while (link != NULL)
                    {
                        TestTypeNode *child = link->data;
                        DBusList *next = _dbus_list_get_next_link (&container-
>children, link);

                        if (!node_write_value (child, block, &sub, seed + i))
                            goto oom;

                        link = next;
                    }

                ++i;
            }
    }

if (!_dbus_type_writer_unrecurse (writer, &sub))
    goto oom;

_dbus_string_free (&element_signature);
return TRUE;

oom:
data_block_restore (block, &saved);
_dbus_string_free (&element_signature);
return FALSE;
}

static dbus_bool_t
array_read_or_set_value (TestTypeNode *node,
                        DBusTypeReader *reader,
                        DBusTypeReader *realign_root,
                        int seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DBusTypeReader sub;
    int i;
    int n_copies;
    TestTypeNode *child;

    n_copies = node->klass->subclass_detail;

    check_expected_type (reader, DBUS_TYPE_ARRAY);

```

```

child = _dbus_list_get_first (&container->children);

if (n_copies > 0)
{
    _dbus_type_reader_recurse (reader, &sub);

    if (realign_root == NULL && arrays_write_fixed_in_blocks &&
        dbus_type_is_fixed (_dbus_type_reader_get_element_type
(reader)) &&
        child->klass->read_multi)
    {
        if (!node_read_multi (child, &sub, seed, n_copies))
            return FALSE;
    }
    else
    {
        i = 0;
        while (i < n_copies)
        {
            DBusList *link;

            link = _dbus_list_get_first_link (&container->children);
            while (link != NULL)
            {
                TestTypeNode *child = link->data;
                DBusList *next = _dbus_list_get_next_link
(&container->children, link);

                _dbus_assert (child->klass->typecode ==
                    _dbus_type_reader_get_element_type
(reader));

                if (realign_root == NULL)
                {
                    if (!node_read_value (child, &sub, seed + i))
                        return FALSE;
                }
                else
                {
                    if (!node_set_value (child, &sub, realign_root,
seed + i))

                        return FALSE;
                }

                if (i == (n_copies - 1) && next == NULL)
                    NEXT_EXPECTING_FALSE (&sub);
                else
                    NEXT_EXPECTING_TRUE (&sub);

                link = next;
            }
        }
    }
}

```

```

        ++i;
    }
}

return TRUE;
}

static dbus_bool_t
array_read_value (TestTypeNode *node,
                 DBusTypeReader *reader,
                 int seed)
{
    return array_read_or_set_value (node, reader, NULL, seed);
}

static dbus_bool_t
array_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int seed)
{
    return array_read_or_set_value (node, reader, realign_root, seed);
}

static dbus_bool_t
array_build_signature (TestTypeNode *node,
                     DBusString *str)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    int orig_len;

    orig_len = _dbus_string_get_length (str);

    if (!_dbus_string_append_byte (str, DBUS_TYPE_ARRAY))
        goto oom;

    if (!node_build_signature (_dbus_list_get_first (&container->children),
                             str))
        goto oom;

    return TRUE;

oom:
    _dbus_string_set_length (str, orig_len);
    return FALSE;
}

/* 10 is random just to add another seed that we use in the suite */
#define VARIANT_SEED 10

```



```

static dbus_bool_t
variant_write_value (TestTypeNode *node,
                    DataBlock      *block,
                    DBusTypeWriter *writer,
                    int             seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DataBlockState saved;
    DBusTypeWriter sub;
    DBusString content_signature;
    TestTypeNode *child;

    _dbus_assert (container->children != NULL);
    _dbus_assert (_dbus_list_length_is_one (&container->children));

    child = _dbus_list_get_first (&container->children);

    data_block_save (block, &saved);

    if (!_dbus_string_init (&content_signature))
        return FALSE;

    if (!node_build_signature (child,
                              &content_signature))
        goto oom;

    if (!_dbus_type_writer_recurse (writer, DBUS_TYPE_VARIANT,
                                   &content_signature, 0,
                                   &sub))
        goto oom;

    if (!node_write_value (child, block, &sub, seed + VARIANT_SEED))
        goto oom;

    if (!_dbus_type_writer_unrecurse (writer, &sub))
        goto oom;

    _dbus_string_free (&content_signature);
    return TRUE;

oom:
    data_block_restore (block, &saved);
    _dbus_string_free (&content_signature);
    return FALSE;
}

static dbus_bool_t
variant_read_or_set_value (TestTypeNode *node,
                          DBusTypeReader *reader,
                          DBusTypeReader *realign_root,
                          int             seed)

```

```

{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DBusTypeReader sub;
    TestTypeNode *child;

    _dbus_assert (container->children != NULL);
    _dbus_assert (_dbus_list_length_is_one (&container->children));

    child = _dbus_list_get_first (&container->children);

    check_expected_type (reader, DBUS_TYPE_VARIANT);

    _dbus_type_reader_recurse (reader, &sub);

    if (realign_root == NULL)
    {
        if (!node_read_value (child, &sub, seed + VARIANT_SEED))
            return FALSE;
    }
    else
    {
        if (!node_set_value (child, &sub, realign_root, seed +
VARIANT_SEED))
            return FALSE;
    }

    NEXT_EXPECTING_FALSE (&sub);

    return TRUE;
}

static dbus_bool_t
variant_read_value (TestTypeNode *node,
                   DBusTypeReader *reader,
                   int seed)
{
    return variant_read_or_set_value (node, reader, NULL, seed);
}

static dbus_bool_t
variant_set_value (TestTypeNode *node,
                  DBusTypeReader *reader,
                  DBusTypeReader *realign_root,
                  int seed)
{
    return variant_read_or_set_value (node, reader, realign_root, seed);
}

static dbus_bool_t
dict_write_value (TestTypeNode *node,
                 DataBlock *block,
                 DBusTypeWriter *writer,

```



```

i = 0;
while (i < n_entries)
{
    DBusTypeWriter entry_sub;
    dbus_int32_t key;

    if (!_dbus_type_writer_recurse (&sub, DBUS_TYPE_DICT_ENTRY,
                                    NULL, 0,
                                    &entry_sub))

        goto oom;

    key = int32_from_seed (seed + i);

    if (!_dbus_type_writer_write_basic (&entry_sub,
                                        DBUS_TYPE_INT32,
                                        &key))

        goto oom;

    if (!node_write_value (child, block, &entry_sub, seed + i))
        goto oom;

    if (!_dbus_type_writer_unrecurse (&sub, &entry_sub))
        goto oom;

    ++i;
}

if (!_dbus_type_writer_unrecurse (writer, &sub))
    goto oom;

_dbus_string_free (&entry_value_signature);
_dbus_string_free (&dict_entry_signature);
return TRUE;

oom:
data_block_restore (block, &saved);
_dbus_string_free (&entry_value_signature);
_dbus_string_free (&dict_entry_signature);
return FALSE;
}

static dbus_bool_t
dict_read_or_set_value (TestTypeNode *node,
                       DBusTypeReader *reader,
                       DBusTypeReader *realign_root,
                       int seed)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DBusTypeReader sub;
    int i;
    int n_entries;
    TestTypeNode *child;

```

```

n_entries = node->klass->subclass_detail;

check_expected_type (reader, DBUS_TYPE_ARRAY);

child = _dbus_list_get_first (&container->children);

if (n_entries > 0)
{
    _dbus_type_reader_recurse (reader, &sub);

    check_expected_type (&sub, DBUS_TYPE_DICT_ENTRY);

    i = 0;
    while (i < n_entries)
    {
        DBusTypeReader entry_sub;

        check_expected_type (&sub, DBUS_TYPE_DICT_ENTRY);

        _dbus_type_reader_recurse (&sub, &entry_sub);

        if (realign_root == NULL)
        {
            dbus_int32_t v;

            check_expected_type (&entry_sub, DBUS_TYPE_INT32);

            _dbus_type_reader_read_basic (&entry_sub,
                                         (dbus_int32_t*) &v);

            _dbus_assert (v == int32_from_seed (seed + i));

            NEXT_EXPECTING_TRUE (&entry_sub);

            if (!node_read_value (child, &entry_sub, seed + i))
                return FALSE;

            NEXT_EXPECTING_FALSE (&entry_sub);
        }
        else
        {
            dbus_int32_t v;

            v = int32_from_seed (seed + i);

            if (!_dbus_type_reader_set_basic (&entry_sub,
                                             &v,
                                             realign_root))
                return FALSE;

            NEXT_EXPECTING_TRUE (&entry_sub);
        }
    }
}

```

```

        if (!node_set_value (child, &entry_sub, realign_root,
seed + i))
            return FALSE;

        NEXT_EXPECTING_FALSE (&entry_sub);
    }

    if (i == (n_entries - 1))
        NEXT_EXPECTING_FALSE (&sub);
    else
        NEXT_EXPECTING_TRUE (&sub);

    ++i;
}

return TRUE;
}

static dbus_bool_t
dict_read_value (TestTypeNode *node,
                DBusTypeReader *reader,
                int seed)
{
    return dict_read_or_set_value (node, reader, NULL, seed);
}

static dbus_bool_t
dict_set_value (TestTypeNode *node,
                DBusTypeReader *reader,
                DBusTypeReader *realign_root,
                int seed)
{
    return dict_read_or_set_value (node, reader, realign_root, seed);
}

static dbus_bool_t
dict_build_signature (TestTypeNode *node,
                    DBusString *str)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    int orig_len;

    orig_len = _dbus_string_get_length (str);

    if (!_dbus_string_append_byte (str, DBUS_TYPE_ARRAY))
        goto oom;

    if (!_dbus_string_append (str, DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING
DBUS_TYPE_INT32_AS_STRING))
        goto oom;

```

```

    if (!node_build_signature (_dbus_list_get_first (&container-
>children),
                                str))
        goto oom;

    if (!_dbus_string_append_byte (str, DBUS_DICT_ENTRY_END_CHAR))
        goto oom;

    return TRUE;

oom:
    _dbus_string_set_length (str, orig_len);
    return FALSE;
}

static void
container_destroy (TestTypeNode *node)
{
    TestTypeNodeContainer *container = (TestTypeNodeContainer*) node;
    DBusList *link;

    link = _dbus_list_get_first_link (&container->children);
    while (link != NULL)
    {
        TestTypeNode *child = link->data;
        DBusList *next = _dbus_list_get_next_link (&container->children,
link);

        node_destroy (child);

        _dbus_list_free_link (link);

        link = next;
    }
}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-marshal-recursive.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-recursive.c  Marshalling routines for recursive types
 *
 * Copyright (C) 2004, 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *

```



```

/** turn this on to get deluged in TypeReader verbose spam */
#define RECURSIVE_MARSHAL_READ_TRACE 0

/** turn this on to get deluged in TypeWriter verbose spam */
#define RECURSIVE_MARSHAL_WRITE_TRACE 0

static void
free_fixups (DBusList **fixups)
{
    DBusList *link;

    link = _dbus_list_get_first_link (fixups);
    while (link != NULL)
    {
        DBusList *next;

        next = _dbus_list_get_next_link (fixups, link);

        dbus_free (link->data);
        _dbus_list_free_link (link);

        link = next;
    }

    *fixups = NULL;
}

static void
apply_and_free_fixups (DBusList **fixups,
                       DBusTypeReader *reader)
{
    DBusList *link;

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    if (*fixups)
        _dbus_verbose (" %d FIXUPS to apply\n",
                       _dbus_list_get_length (fixups));
#endif

    link = _dbus_list_get_first_link (fixups);
    while (link != NULL)
    {
        DBusList *next;

        next = _dbus_list_get_next_link (fixups, link);

        if (reader)
        {
            DBusArrayLenFixup *f;

            f = link->data;

```

```

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" applying FIXUP to reader %p at pos %d
new_len = %d old len %d\n",
                reader, f->len_pos_in_reader, f->new_len,
                _dbus_marshal_read_uint32 (reader->value_str,
                f-
>len_pos_in_reader,
                reader-
>byte_order, NULL));
#endif

        _dbus_marshal_set_uint32 ((DBusString*) reader->value_str,
                f->len_pos_in_reader,
                f->new_len,
                reader->byte_order);
    }

    dbus_free (link->data);
    _dbus_list_free_link (link);

    link = next;
}

*fixups = NULL;
}

/**
 * Virtual table for a type reader.
 */
struct DBusTypeReaderClass
{
    const char *name;          /**< name for debugging */
    int         id;           /**< index in all_reader_classes */
    dbus_bool_t types_only;  /**< only iterates over types, not values */
    void        (* recurse)   (DBusTypeReader *sub,
                              DBusTypeReader *parent);
    /**< recurse with this reader as sub */
    dbus_bool_t (* check_finished) (const DBusTypeReader *reader);
    /**< check whether reader is at the end */
    void        (* next)      (DBusTypeReader *reader,
                              int
current_type); /**< go to the next value */
};

static int
element_type_get_alignment (const DBusString *str,
                           int                pos)
{
    return _dbus_type_get_alignment (_dbus_first_type_in_signature (str,
pos));
}

```

```

static void
reader_init (DBusTypeReader *reader,
            int byte_order,
            const DBusString *type_str,
            int type_pos,
            const DBusString *value_str,
            int value_pos)
{
    reader->byte_order = byte_order;
    reader->finished = FALSE;
    reader->type_str = type_str;
    reader->type_pos = type_pos;
    reader->value_str = value_str;
    reader->value_pos = value_pos;
}

static void
base_reader_recurse (DBusTypeReader *sub,
                    DBusTypeReader *parent)
{
    /* point subreader at the same place as parent */
    reader_init (sub,
                parent->byte_order,
                parent->type_str,
                parent->type_pos,
                parent->value_str,
                parent->value_pos);
}

static void
struct_or_dict_entry_types_only_reader_recurse (DBusTypeReader *sub,
                                                DBusTypeReader
*parent)
{
    base_reader_recurse (sub, parent);

    _dbus_assert (_dbus_string_get_byte (sub->type_str,
                                        sub->type_pos) ==
DBUS_STRUCT_BEGIN_CHAR ||
                 _dbus_string_get_byte (sub->type_str,
                                        sub->type_pos) ==
DBUS_DICT_ENTRY_BEGIN_CHAR);

    sub->type_pos += 1;
}

static void
struct_or_dict_entry_reader_recurse (DBusTypeReader *sub,
                                    DBusTypeReader *parent)
{
    struct_or_dict_entry_types_only_reader_recurse (sub, parent);
}

```

```

    /* struct and dict entry have 8 byte alignment */
    sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos, 8);
}

static void
array_types_only_reader_recurse (DBusTypeReader *sub,
                                 DBusTypeReader *parent)
{
    base_reader_recurse (sub, parent);

    /* point type_pos at the array element type */
    sub->type_pos += 1;

    /* Init with values likely to crash things if misused */
    sub->u.array.start_pos = _DBUS_INT_MAX;
    sub->array_len_offset = 7;
}

/** compute position of array length given array_len_offset, which is
    the offset back from start_pos to end of the len */
#define ARRAY_READER_LEN_POS(reader) \
    ((reader)->u.array.start_pos - ((int)(reader)->array_len_offset) -
4)

static int
array_reader_get_array_len (const DBusTypeReader *reader)
{
    dbus_uint32_t array_len;
    int len_pos;

    len_pos = ARRAY_READER_LEN_POS (reader);

    _dbus_assert (_DBUS_ALIGN_VALUE (len_pos, 4) == (unsigned) len_pos);
    array_len = _dbus_unpack_uint32 (reader->byte_order,
                                     _dbus_string_get_const_data_len
(reader->value_str, len_pos, 4));

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose ("    reader %p len_pos %d array len %u len_offset
%d\n",
                  reader, len_pos, array_len, reader-
>array_len_offset);
#endif

    _dbus_assert (reader->u.array.start_pos - len_pos - 4 < 8);

    return array_len;
}

static void
array_reader_recurse (DBusTypeReader *sub,
                     DBusTypeReader *parent)

```

```

{
    int alignment;
    int len_pos;

    array_types_only_reader_recurse (sub, parent);

    sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos, 4);

    len_pos = sub->value_pos;

    sub->value_pos += 4; /* for the length */

    alignment = element_type_get_alignment (sub->type_str,
                                             sub->type_pos);

    sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos, alignment);

    sub->u.array.start_pos = sub->value_pos;
    _dbus_assert ((sub->u.array.start_pos - (len_pos + 4)) < 8); /* only
3 bits in array_len_offset */
    sub->array_len_offset = sub->u.array.start_pos - (len_pos + 4);

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose ("    type reader %p array start = %d len_offset = %d
array len = %d array element type = %s\n",
                  sub,
                  sub->u.array.start_pos,
                  sub->array_len_offset,
                  array_reader_get_array_len (sub),
                  _dbus_type_to_string (_dbus_first_type_in_signature
(sub->type_str,
                                     sub-
>type_pos)));
#endif
}

static void
variant_reader_recurse (DBusTypeReader *sub,
                        DBusTypeReader *parent)
{
    int sig_len;
    int contained_alignment;

    base_reader_recurse (sub, parent);

    /* Variant is 1 byte sig length (without nul), signature with nul,
    * padding to 8-boundary, then values
    */

    sig_len = _dbus_string_get_byte (sub->value_str, sub->value_pos);

    sub->type_str = sub->value_str;

```

```

sub->type_pos = sub->value_pos + 1;

sub->value_pos = sub->type_pos + sig_len + 1;

contained_alignment = _dbus_type_get_alignment
(_dbus_first_type_in_signature (sub->type_str,
sub->type_pos));

sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos,
contained_alignment);

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose ("    type reader %p variant containing '%s'\n",
        sub,
        _dbus_string_get_const_data_len (sub->type_str,
sub->type_pos, 0));
#endif
}

static dbus_bool_t
array_reader_check_finished (const DBusTypeReader *reader)
{
    int end_pos;

    /* return the array element type if elements remain, and
     * TYPE_INVALID otherwise
     */

    end_pos = reader->u.array.start_pos + array_reader_get_array_len
(reader);

    _dbus_assert (reader->value_pos <= end_pos);
    _dbus_assert (reader->value_pos >= reader->u.array.start_pos);

    return reader->value_pos == end_pos;
}

static void
skip_one_complete_type (const DBusString *type_str,
                        int *type_pos)
{
    _dbus_type_signature_next (_dbus_string_get_const_data (type_str),
type_pos);
}

/**
 * Skips to the next "complete" type inside a type signature.
 * The signature is read starting at type_pos, and the next
 * type position is stored in the same variable.
 *
 * @param type_str a type signature (must be valid)

```

```

* @param type_pos an integer position in the type signature (in and
out)
*/
void
_dbus_type_signature_next (const char          *type_str,
                           int                *type_pos)
{
    const unsigned char *p;
    const unsigned char *start;

    _dbus_assert (type_str != NULL);
    _dbus_assert (type_pos != NULL);

    start = type_str;
    p = start + *type_pos;

    _dbus_assert (*p != DBUS_STRUCT_END_CHAR);
    _dbus_assert (*p != DBUS_DICT_ENTRY_END_CHAR);

    while (*p == DBUS_TYPE_ARRAY)
        ++p;

    _dbus_assert (*p != DBUS_STRUCT_END_CHAR);
    _dbus_assert (*p != DBUS_DICT_ENTRY_END_CHAR);

    if (*p == DBUS_STRUCT_BEGIN_CHAR)
    {
        int depth;

        depth = 1;

        while (TRUE)
        {
            _dbus_assert (*p != DBUS_TYPE_INVALID);

            ++p;

            _dbus_assert (*p != DBUS_TYPE_INVALID);

            if (*p == DBUS_STRUCT_BEGIN_CHAR)
                depth += 1;
            else if (*p == DBUS_STRUCT_END_CHAR)
            {
                depth -= 1;
                if (depth == 0)
                {
                    ++p;
                    break;
                }
            }
        }
    }
}

```

```

else if (*p == DBUS_DICT_ENTRY_BEGIN_CHAR)
{
    int depth;

    depth = 1;

    while (TRUE)
    {
        _dbus_assert (*p != DBUS_TYPE_INVALID);

        ++p;

        _dbus_assert (*p != DBUS_TYPE_INVALID);

        if (*p == DBUS_DICT_ENTRY_BEGIN_CHAR)
            depth += 1;
        else if (*p == DBUS_DICT_ENTRY_END_CHAR)
        {
            depth -= 1;
            if (depth == 0)
            {
                ++p;
                break;
            }
        }
    }
}
else
{
    ++p;
}

*type_pos = (int) (p - start);
}

static int
find_len_of_complete_type (const DBusString *type_str,
                           int                type_pos)
{
    int end;

    end = type_pos;

    skip_one_complete_type (type_str, &end);

    return end - type_pos;
}

static void
base_reader_next (DBusTypeReader *reader,
                  int              current_type)
{

```



```

switch (current_type)
{
case DBUS_TYPE_DICT_ENTRY:
case DBUS_TYPE_STRUCT:
case DBUS_TYPE_VARIANT:
    /* Scan forward over the entire container contents */
    {
        DBusTypeReader sub;

        if (reader->klass->types_only && current_type ==
DBUS_TYPE_VARIANT)
            ;
        else
        {
            /* Recurse into the struct or variant */
            _dbus_type_reader_recurse (reader, &sub);

            /* Skip everything in this subreader */
            while (_dbus_type_reader_next (&sub))
            {
                /* nothing */;
            }
        }
        if (!reader->klass->types_only)
            reader->value_pos = sub.value_pos;

        /* Now we are at the end of this container; for variants, the
         * subreader's type_pos is totally inapplicable (it's in the
         * value string) but we know that we increment by one past the
         * DBUS_TYPE_VARIANT
         */
        if (current_type == DBUS_TYPE_VARIANT)
            reader->type_pos += 1;
        else
            reader->type_pos = sub.type_pos;
    }
    break;

case DBUS_TYPE_ARRAY:
    {
        if (!reader->klass->types_only)
            _dbus_marshal_skip_array (reader->value_str,
                                     _dbus_first_type_in_signature
(reader->type_str,
reader->type_pos + 1),
                                     reader->byte_order,
                                     &reader->value_pos);

        skip_one_complete_type (reader->type_str, &reader->type_pos);
    }
    break;

```

```

default:
    if (!reader->klass->types_only)
        _dbus_marshal_skip_basic (reader->value_str,
                                   current_type, reader->byte_order,
                                   &reader->value_pos);

    reader->type_pos += 1;
    break;
}
}

static void
struct_reader_next (DBusTypeReader *reader,
                    int current_type)
{
    int t;

    base_reader_next (reader, current_type);

    /* for STRUCT containers we return FALSE at the end of the struct,
     * for INVALID we return FALSE at the end of the signature.
     * In both cases we arrange for get_current_type() to return INVALID
     * which is defined to happen iff we're at the end (no more next())
     */
    t = _dbus_string_get_byte (reader->type_str, reader->type_pos);
    if (t == DBUS_STRUCT_END_CHAR)
    {
        reader->type_pos += 1;
        reader->finished = TRUE;
    }
}

static void
dict_entry_reader_next (DBusTypeReader *reader,
                        int current_type)
{
    int t;

    base_reader_next (reader, current_type);

    /* for STRUCT containers we return FALSE at the end of the struct,
     * for INVALID we return FALSE at the end of the signature.
     * In both cases we arrange for get_current_type() to return INVALID
     * which is defined to happen iff we're at the end (no more next())
     */
    t = _dbus_string_get_byte (reader->type_str, reader->type_pos);
    if (t == DBUS_DICT_ENTRY_END_CHAR)
    {
        reader->type_pos += 1;
        reader->finished = TRUE;
    }
}

```

```

}

static void
array_types_only_reader_next (DBusTypeReader *reader,
                             int             current_type)
{
    /* We have one "element" to be iterated over
     * in each array, which is its element type.
     * So the finished flag indicates whether we've
     * iterated over it yet or not.
     */
    reader->finished = TRUE;
}

static void
array_reader_next (DBusTypeReader *reader,
                  int             current_type)
{
    /* Skip one array element */
    int end_pos;

    end_pos = reader->u.array.start_pos + array_reader_get_array_len
(reader);

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" reader %p array next START start_pos = %d end_pos
= %d value_pos = %d current_type = %s\n",
                  reader,
                  reader->u.array.start_pos,
                  end_pos, reader->value_pos,
                  _dbus_type_to_string (current_type));
#endif

    _dbus_assert (reader->value_pos < end_pos);
    _dbus_assert (reader->value_pos >= reader->u.array.start_pos);

    switch (_dbus_first_type_in_signature (reader->type_str,
                                          reader->type_pos))
    {
        case DBUS_TYPE_DICT_ENTRY:
        case DBUS_TYPE_STRUCT:
        case DBUS_TYPE_VARIANT:
            {
                DBusTypeReader sub;

                /* Recurse into the struct or variant */
                _dbus_type_reader_recurse (reader, &sub);

                /* Skip everything in this element */
                while (_dbus_type_reader_next (&sub))
                {
                    /* nothing */;
                }
            }
    }
}

```

```

    }

    /* Now we are at the end of this element */
    reader->value_pos = sub.value_pos;
}
break;

case DBUS_TYPE_ARRAY:
{
    _dbus_marshal_skip_array (reader->value_str,
                             _dbus_first_type_in_signature
(reader->type_str,
                             reader-
>type_pos + 1),
                             reader->byte_order,
                             &reader->value_pos);
}
break;

default:
{
    _dbus_marshal_skip_basic (reader->value_str,
                              current_type, reader->byte_order,
                              &reader->value_pos);
}
break;
}

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" reader %p array next END start_pos = %d end_pos =
%d value_pos = %d current_type = %s\n",
                  reader,
                  reader->u.array.start_pos,
                  end_pos, reader->value_pos,
                  _dbus_type_to_string (current_type));
#endif

    _dbus_assert (reader->value_pos <= end_pos);

    if (reader->value_pos == end_pos)
    {
        skip_one_complete_type (reader->type_str,
                                &reader->type_pos);
    }
}

static const DBusTypeReaderClass body_reader_class = {
    "body", 0,
    FALSE,
    NULL, /* body is always toplevel, so doesn't get recursed into */
    NULL,
    base_reader_next

```

```

};

static const DBusTypeReaderClass body_types_only_reader_class = {
    "body types", 1,
    TRUE,
    NULL, /* body is always toplevel, so doesn't get recursed into */
    NULL,
    base_reader_next
};

static const DBusTypeReaderClass struct_reader_class = {
    "struct", 2,
    FALSE,
    struct_or_dict_entry_reader_recurse,
    NULL,
    struct_reader_next
};

static const DBusTypeReaderClass struct_types_only_reader_class = {
    "struct types", 3,
    TRUE,
    struct_or_dict_entry_types_only_reader_recurse,
    NULL,
    struct_reader_next
};

static const DBusTypeReaderClass dict_entry_reader_class = {
    "dict_entry", 4,
    FALSE,
    struct_or_dict_entry_reader_recurse,
    NULL,
    dict_entry_reader_next
};

static const DBusTypeReaderClass dict_entry_types_only_reader_class =
{
    "dict_entry types", 5,
    TRUE,
    struct_or_dict_entry_types_only_reader_recurse,
    NULL,
    dict_entry_reader_next
};

static const DBusTypeReaderClass array_reader_class = {
    "array", 6,
    FALSE,
    array_reader_recurse,
    array_reader_check_finished,
    array_reader_next
};

static const DBusTypeReaderClass array_types_only_reader_class = {

```

```

    "array types", 7,
    TRUE,
    array_types_only_reader_recurse,
    NULL,
    array_types_only_reader_next
};

static const DBusTypeReaderClass variant_reader_class = {
    "variant", 8,
    FALSE,
    variant_reader_recurse,
    NULL,
    base_reader_next
};

#ifdef DBUS_DISABLE_ASSERT
static const DBusTypeReaderClass * const
all_reader_classes[] = {
    &body_reader_class,
    &body_types_only_reader_class,
    &struct_reader_class,
    &struct_types_only_reader_class,
    &dict_entry_reader_class,
    &dict_entry_types_only_reader_class,
    &array_reader_class,
    &array_types_only_reader_class,
    &variant_reader_class
};
#endif

/**
 * Initializes a type reader.
 *
 * @param reader the reader
 * @param byte_order the byte order of the block to read
 * @param type_str the signature of the block to read
 * @param type_pos location of signature
 * @param value_str the string containing values block
 * @param value_pos start of values block
 */
void
_dbus_type_reader_init (DBusTypeReader *reader,
                       int byte_order,
                       const DBusString *type_str,
                       int type_pos,
                       const DBusString *value_str,
                       int value_pos)
{
    reader->klass = &body_reader_class;

    reader_init (reader, byte_order, type_str, type_pos,
                value_str, value_pos);
}

```

```

#if RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p init type_pos = %d value_pos = %d
remaining sig '%s'\n",
                  reader, reader->type_pos, reader->value_pos,
                  _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0));
#endif
}

/**
 * Like _dbus_type_reader_init() but the iteration is over the
 * signature, not over values.
 *
 * @param reader the reader
 * @param type_str the signature string
 * @param type_pos location in the signature string
 */
void
_dbus_type_reader_init_types_only (DBusTypeReader *reader,
                                   const DBusString *type_str,
                                   int type_pos)
{
    reader->klass = &body_types_only_reader_class;

    reader_init (reader, DBUS_COMPILER_BYTE_ORDER /* irrelevant */,
                type_str, type_pos, NULL, _DBUS_INT_MAX /* crashes if
we screw up */);

#if RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p init types only type_pos = %d
remaining sig '%s'\n",
                  reader, reader->type_pos,
                  _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0));
#endif
}

/**
 * Gets the type of the value the reader is currently pointing to;
 * or for a types-only reader gets the type it's currently pointing
to.
 * If the reader is at the end of a block or end of a container such
 * as an array, returns #DBUS_TYPE_INVALID.
 *
 * @param reader the reader
 */
int
_dbus_type_reader_get_current_type (const DBusTypeReader *reader)
{
    int t;

```

```

if (reader->finished ||
    (reader->klass->check_finished &&
     (* reader->klass->check_finished) (reader)))
    t = DBUS_TYPE_INVALID;
else
    t = _dbus_first_type_in_signature (reader->type_str,
                                       reader->type_pos);

    _dbus_assert (t != DBUS_STRUCT_END_CHAR);
    _dbus_assert (t != DBUS_STRUCT_BEGIN_CHAR);
    _dbus_assert (t != DBUS_DICT_ENTRY_END_CHAR);
    _dbus_assert (t != DBUS_DICT_ENTRY_BEGIN_CHAR);

#ifdef 0
    _dbus_verbose (" type reader %p current type_pos = %d type = %s\n",
                  reader, reader->type_pos,
                  _dbus_type_to_string (t));
#endif

    return t;
}

/**
 * Gets the type of an element of the array the reader is currently
 * pointing to. It's an error to call this if
 * _dbus_type_reader_get_current_type() doesn't return
 * #DBUS_TYPE_ARRAY
 * for this reader.
 *
 * @param reader the reader
 */
int
_dbus_type_reader_get_element_type (const DBusTypeReader *reader)
{
    int element_type;

    _dbus_assert (_dbus_type_reader_get_current_type (reader) ==
                  DBUS_TYPE_ARRAY);

    element_type = _dbus_first_type_in_signature (reader->type_str,
                                                  reader->type_pos + 1);

    return element_type;
}

/**
 * Gets the current position in the value block
 *
 * @param reader the reader
 */
int
_dbus_type_reader_get_value_pos (const DBusTypeReader *reader)
{

```



```

    return reader->value_pos;
}

/**
 * Get the address of the marshaled value in the data being read. The
 * address may not be aligned; you have to align it to the type of the
 * value you want to read. Most of the demarshal routines do this for
 * you.
 *
 * @param reader the reader
 * @param value_location the address of the marshaled value
 */
void
_dbus_type_reader_read_raw (const DBusTypeReader *reader,
                           const unsigned char **value_location)
{
    _dbus_assert (!reader->klass->types_only);

    *value_location = _dbus_string_get_const_data_len (reader-
>value_str,
                                                    reader-
>value_pos,
                                                    0);
}

/**
 * Reads a basic-typed value, as with _dbus_marshal_read_basic().
 *
 * @param reader the reader
 * @param value the address of the value
 */
void
_dbus_type_reader_read_basic (const DBusTypeReader *reader,
                             void *value)
{
    int t;

    _dbus_assert (!reader->klass->types_only);

    t = _dbus_type_reader_get_current_type (reader);

    _dbus_marshal_read_basic (reader->value_str,
                              reader->value_pos,
                              t, value,
                              reader->byte_order,
                              NULL);

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p read basic type_pos = %d value_pos
= %d remaining sig '%s'\n",
                  reader, reader->type_pos, reader->value_pos,

```



```

    _dbus_assert (element_type != DBUS_TYPE_INVALID); /* why we don't
use get_current_type() */
    _dbus_assert (dbus_type_is_fixed (element_type));

    alignment = _dbus_type_get_alignment (element_type);

    _dbus_assert (reader->value_pos >= reader->u.array.start_pos);

    total_len = array_reader_get_array_len (reader);
    end_pos = reader->u.array.start_pos + total_len;
    remaining_len = end_pos - reader->value_pos;

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose ("end_pos %d total_len %d remaining_len %d value_pos
%d\n",
                  end_pos, total_len, remaining_len, reader-
>value_pos);
#endif

    _dbus_assert (remaining_len <= total_len);

    if (remaining_len == 0)
        *(const DBusBasicValue**) value = NULL;
    else
        *(const DBusBasicValue**) value =
            (void*) _dbus_string_get_const_data_len (reader->value_str,
                                                    reader->value_pos,
                                                    remaining_len);

    *n_elements = remaining_len / alignment;
    _dbus_assert ((remaining_len % alignment) == 0);

#ifdef RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p read fixed array type_pos = %d
value_pos = %d remaining sig '%s'\n",
                  reader, reader->type_pos, reader->value_pos,
                  _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0));
#endif
}

/**
 * Initialize a new reader pointing to the first type and
 * corresponding value that's a child of the current container. It's
 * an error to call this if the current type is a non-container.
 *
 * Note that DBusTypeReader traverses values, not types. So if you
 * have an empty array of array of int, you can't recurse into it. You
 * can only recurse into each element.
 *
 * @param reader the reader

```

```

    * @param sub a reader to init pointing to the first child
    */
void
_dbus_type_reader_recurse (DBusTypeReader *reader,
                          DBusTypeReader *sub)
{
    int t;

    t = _dbus_first_type_in_signature (reader->type_str, reader->type_pos);

    switch (t)
    {
        case DBUS_TYPE_STRUCT:
            if (reader->klass->types_only)
                sub->klass = &struct_types_only_reader_class;
            else
                sub->klass = &struct_reader_class;
            break;
        case DBUS_TYPE_DICT_ENTRY:
            if (reader->klass->types_only)
                sub->klass = &dict_entry_types_only_reader_class;
            else
                sub->klass = &dict_entry_reader_class;
            break;
        case DBUS_TYPE_ARRAY:
            if (reader->klass->types_only)
                sub->klass = &array_types_only_reader_class;
            else
                sub->klass = &array_reader_class;
            break;
        case DBUS_TYPE_VARIANT:
            if (reader->klass->types_only)
                _dbus_assert_not_reached ("can't recurse into variant
typecode");
            else
                sub->klass = &variant_reader_class;
            break;
        default:
            _dbus_verbose ("recursing into type %s\n", _dbus_type_to_string
(t));
#ifdef DBUS_DISABLE_CHECKS
            if (t == DBUS_TYPE_INVALID)
                _dbus_warn_check_failed ("You can't recurse into an empty
array or off the end of a message body\n");
#endif /* DBUS_DISABLE_CHECKS */

            _dbus_assert_not_reached ("don't yet handle recursing into this
type");
    }

    _dbus_assert (sub->klass == all_reader_classes[sub->klass->id]);
}

```

```

    (* sub->klass->recurse) (sub, reader);

#if RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p RECURSED type_pos = %d value_pos =
%d remaining sig '%s'\n",
                  sub, sub->type_pos, sub->value_pos,
                  _dbus_string_get_const_data_len (sub->type_str, sub-
>type_pos, 0));
#endif
}

/**
 * Skip to the next value on this "level". e.g. the next field in a
 * struct, the next value in an array. Returns FALSE at the end of the
 * current container.
 *
 * @param reader the reader
 * @returns FALSE if nothing more to read at or below this level
 */
dbus_bool_t
_dbus_type_reader_next (DBusTypeReader *reader)
{
    int t;

    t = _dbus_type_reader_get_current_type (reader);

#if RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p START next() { type_pos = %d
value_pos = %d remaining sig '%s' current_type = %s\n",
                  reader, reader->type_pos, reader->value_pos,
                  _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0),
                  _dbus_type_to_string (t));
#endif

    if (t == DBUS_TYPE_INVALID)
        return FALSE;

    (* reader->klass->next) (reader, t);

#if RECURSIVE_MARSHAL_READ_TRACE
    _dbus_verbose (" type reader %p END next() type_pos = %d value_pos
= %d remaining sig '%s' current_type = %s\n",
                  reader, reader->type_pos, reader->value_pos,
                  _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0),
                  _dbus_type_to_string
(_dbus_type_reader_get_current_type (reader)));
#endif
}

```

```

    return _dbus_type_reader_get_current_type (reader) !=
    DBUS_TYPE_INVALID;
}

/**
 * Check whether there's another value on this "level". e.g. the next
 * field in a struct, the next value in an array. Returns FALSE at the
 * end of the current container.
 *
 * You probably don't want to use this; it makes for an awkward
for/while
 * loop. A nicer one is "while ((current_type = get_current_type()) !=
INVALID)"
 *
 * @param reader the reader
 * @returns FALSE if nothing more to read at or below this level
 */
dbus_bool_t
_dbus_type_reader_has_next (const DBusTypeReader *reader)
{
    /* Not efficient but works for now. */
    DBusTypeReader copy;

    copy = *reader;
    return _dbus_type_reader_next (&copy);
}

/**
 * Gets the string and range of said string containing the signature
 * of the current value. Essentially a more complete version of
 * _dbus_type_reader_get_current_type() (returns the full type
 * rather than only the outside of the onion).
 *
 * Note though that the first byte in a struct signature is
 * #DBUS_STRUCT_BEGIN_CHAR while the current type will be
 * #DBUS_TYPE_STRUCT so it isn't true that the first byte of the
 * signature is always the same as the current type. Another
 * difference is that this function will still return a signature when
 * inside an empty array; say you recurse into empty array of int32,
 * the signature is "i" but the current type will always be
 * #DBUS_TYPE_INVALID since there are no elements to be currently
 * pointing to.
 *
 * @param reader the reader
 * @param str_p place to return the string with the type in it
 * @param start_p place to return start of the type
 * @param len_p place to return the length of the type
 */
void
_dbus_type_reader_get_signature (const DBusTypeReader *reader,
                                const DBusString **str_p,
                                int *start_p,

```

```

                                int                                *len_p)
{
    *str_p = reader->type_str;
    *start_p = reader->type_pos;
    *len_p = find_len_of_complete_type (reader->type_str, reader-
>type_pos);
}

typedef struct
{
    DBusString replacement; /**< Marshaled value including alignment
padding */
    int padding;           /**< How much of the replacement block is
padding */
} ReplacementBlock;

static dbus_bool_t
replacement_block_init (ReplacementBlock *block,
                        DBusTypeReader *reader)
{
    if (!_dbus_string_init (&block->replacement))
        return FALSE;

    /* % 8 is the padding to have the same align properties in
    * our replacement string as we do at the position being replaced
    */
    block->padding = reader->value_pos % 8;

    if (!_dbus_string_lengthen (&block->replacement, block->padding))
        goto oom;

    return TRUE;

oom:
    _dbus_string_free (&block->replacement);
    return FALSE;
}

static dbus_bool_t
replacement_block_replace (ReplacementBlock *block,
                           DBusTypeReader *reader,
                           const DBusTypeReader *realign_root)
{
    DBusTypeWriter writer;
    DBusTypeReader realign_reader;
    DBusList *fixups;
    int orig_len;

    _dbus_assert (realign_root != NULL);

    orig_len = _dbus_string_get_length (&block->replacement);

```

```

    realign_reader = *realign_root;

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("INITIALIZING replacement block writer %p at
value_pos %d\n",
        &writer, _dbus_string_get_length (&block-
>replacement));
#endif
    _dbus_type_writer_init_values_only (&writer,
                                        realign_reader.byte_order,
                                        realign_reader.type_str,
                                        realign_reader.type_pos,
                                        &block->replacement,
                                        _dbus_string_get_length (&block-
>replacement));

    _dbus_assert (realign_reader.value_pos <= reader->value_pos);

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("COPYING from reader at value_pos %d to writer %p
starting after value_pos %d\n",
        realign_reader.value_pos, &writer, reader-
>value_pos);
#endif
    fixups = NULL;
    if (!_dbus_type_writer_write_reader_partial (&writer,
                                                &realign_reader,
                                                reader,
                                                block->padding,
                                                _dbus_string_get_length
(&block->replacement) - block->padding,
                                                &fixups))
        goto oom;

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("REPLACEMENT at padding %d len %d\n", block->padding,
        _dbus_string_get_length (&block->replacement) -
block->padding);
    _dbus_verbose_bytes_of_string (&block->replacement, block->padding,
        _dbus_string_get_length (&block-
>replacement) - block->padding);
    _dbus_verbose ("TO BE REPLACED at value_pos = %d (align pad %d) len
%d realign_reader.value_pos %d\n",
        reader->value_pos, reader->value_pos % 8,
        realign_reader.value_pos - reader->value_pos,
        realign_reader.value_pos);
    _dbus_verbose_bytes_of_string (reader->value_str,
        reader->value_pos,
        realign_reader.value_pos - reader-
>value_pos);
#endif

```



```

    /* Move the replacement into position
     * (realign_reader should now be at the end of the block to be
    replaced)
     */
    if (!_dbus_string_replace_len (&block->replacement, block->padding,
    _dbus_string_get_length (&block-
>replacement) - block->padding,
    (DBusString*) reader->value_str,
    reader->value_pos,
    realign_reader.value_pos - reader-
>value_pos))
        goto oom;

    /* Process our fixups now that we can't have an OOM error */
    apply_and_free_fixups (&fixups, reader);

    return TRUE;

oom:
    _dbus_string_set_length (&block->replacement, orig_len);
    free_fixups (&fixups);
    return FALSE;
}

static void
replacement_block_free (ReplacementBlock *block)
{
    _dbus_string_free (&block->replacement);
}

/* In the variable-length case, we have to fix alignment after we
insert.
* The strategy is as follows:
*
* - pad a new string to have the same alignment as the
*   start of the current basic value
* - write the new basic value
* - copy from the original reader to the new string,
*   which will fix the alignment of types following
*   the new value
* - this copy has to start at realign_root,
*   but not really write anything until it
*   passes the value being set
* - as an optimization, we can stop copying
*   when the source and dest values are both
*   on an 8-boundary, since we know all following
*   padding and alignment will be identical
* - copy the new string back to the original
*   string, replacing the relevant part of the
*   original string
* - now any arrays in the original string that
*   contained the replaced string may have the

```

```

*      wrong length; so we have to fix that
*/
static dbus_bool_t
reader_set_basic_variable_length (DBusTypeReader      *reader,
                                  int                  current_type,
                                  const void          *value,
                                  const DBusTypeReader *realign_root)
{
    dbus_bool_t retval;
    ReplacementBlock block;
    DBusTypeWriter writer;

    _dbus_assert (realign_root != NULL);

    retval = FALSE;

    if (!replacement_block_init (&block, reader))
        return FALSE;

    /* Write the new basic value */
#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("INITIALIZING writer %p to write basic value at
value_pos %d of replacement string\n",
                  &writer, _dbus_string_get_length
(&block.replacement));
#endif
    _dbus_type_writer_init_values_only (&writer,
                                       reader->byte_order,
                                       reader->type_str,
                                       reader->type_pos,
                                       &block.replacement,
                                       _dbus_string_get_length
(&block.replacement));
#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("WRITING basic value to writer %p (replacement
string)\n", &writer);
#endif
    if (!_dbus_type_writer_write_basic (&writer, current_type, value))
        goto out;

    if (!replacement_block_replace (&block,
                                    reader,
                                    realign_root))

        goto out;

    retval = TRUE;

out:
    replacement_block_free (&block);
    return retval;
}

```

```

static void
reader_set_basic_fixed_length (DBusTypeReader *reader,
                               int             current_type,
                               const void     *value)
{
    _dbus_marshal_set_basic ((DBusString*) reader->value_str,
                             reader->value_pos,
                             current_type,
                             value,
                             reader->byte_order,
                             NULL, NULL);
}

/**
 * Sets a new value for the basic type value pointed to by the reader,
 * leaving the reader valid to continue reading. Any other readers
 * will be invalidated if you set a variable-length type such as a
 * string.
 *
 * The provided realign_root is the reader to start from when
 * realigning the data that follows the newly-set value. The reader
 * parameter must point to a value below the realign_root parameter.
 * If the type being set is fixed-length, then realign_root may be
 * #NULL. Only values reachable from realign_root will be realigned,
 * so if your string contains other values you will need to deal with
 * those somehow yourself. It is OK if realign_root is the same
 * reader as the reader parameter, though if you aren't setting the
 * root it may not be such a good idea.
 *
 * @todo DBusTypeReader currently takes "const" versions of the type
 * and value strings, and this function modifies those strings by
 * casting away the const, which is of course bad if we want to get
 * picky. (To be truly clean you'd have an object which contained the
 * type and value strings and set_basic would be a method on that
 * object... this would also make DBusTypeReader the same thing as
 * DBusTypeMark. But since DBusMessage is effectively that object for
 * D-Bus it doesn't seem worth creating some random object.)
 *
 * @todo optimize this by only rewriting until the old and new values
 * are at the same alignment. Frequently this should result in only
 * replacing the value that's immediately at hand.
 *
 * @param reader reader indicating where to set a new value
 * @param value address of the value to set
 * @param realign_root realign from here
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
_dbus_type_reader_set_basic (DBusTypeReader *reader,
                             const void     *value,
                             const DBusTypeReader *realign_root)
{

```

```

int current_type;

_dbus_assert (!reader->klass->types_only);
_dbus_assert (reader->value_str == realign_root->value_str);
_dbus_assert (reader->value_pos >= realign_root->value_pos);

current_type = _dbus_type_reader_get_current_type (reader);

#if RECURSIVE_MARSHAL_WRITE_TRACE
_dbus_verbose (" SET BASIC type reader %p type_pos = %d value_pos =
%d remaining sig '%s' realign_root = %p with value_pos %d current_type
= %s\n",
                reader, reader->type_pos, reader->value_pos,
                _dbus_string_get_const_data_len (reader->type_str,
reader->type_pos, 0),
                realign_root,
                realign_root ? realign_root->value_pos : -1,
                _dbus_type_to_string (current_type));
_dbus_verbose_bytes_of_string (realign_root->value_str,
realign_root->value_pos,
                _dbus_string_get_length
(realign_root->value_str) -
                realign_root->value_pos);
#endif

_dbus_assert (dbus_type_is_basic (current_type));

if (dbus_type_is_fixed (current_type))
{
    reader_set_basic_fixed_length (reader, current_type, value);
    return TRUE;
}
else
{
    _dbus_assert (realign_root != NULL);
    return reader_set_basic_variable_length (reader, current_type,
value, realign_root);
}
}

/**
 * Recursively deletes any value pointed to by the reader, leaving the
 * reader valid to continue reading. Any other readers will be
 * invalidated.
 *
 * The provided realign_root is the reader to start from when
 * realigning the data that follows the newly-set value.
 * See _dbus_type_reader_set_basic() for more details on the
 * realign_root paramter.
 *
 * @todo for now this does not delete the typecodes associated with
 * the value, so this function should only be used for array elements.

```

```

*
* @param reader reader indicating where to delete a value
* @param realign_root realign from here
* @returns #FALSE if not enough memory
*/
dbus_bool_t
_dbus_type_reader_delete (DBusTypeReader      *reader,
                          const DBusTypeReader *realign_root)
{
    dbus_bool_t retval;
    ReplacementBlock block;

    _dbus_assert (realign_root != NULL);
    _dbus_assert (reader->klass == &array_reader_class);

    retval = FALSE;

    if (!replacement_block_init (&block, reader))
        return FALSE;

    if (!replacement_block_replace (&block,
                                    reader,
                                    realign_root))
        goto out;

    retval = TRUE;

out:
    replacement_block_free (&block);
    return retval;
}

/*
 * Compares two readers, which must be iterating over the same value
 * data.
 * Returns #TRUE if the first parameter is further along than the
 * second parameter.
 *
 * @param lhs left-hand-side (first) parameter
 * @param rhs left-hand-side (first) parameter
 * @returns whether lhs is greater than rhs
 */
static dbus_bool_t
_dbus_type_reader_greater_than (const DBusTypeReader *lhs,
                                const DBusTypeReader *rhs)
{
    _dbus_assert (lhs->value_str == rhs->value_str);

    return lhs->value_pos > rhs->value_pos;
}

/*

```

```

*
*
*       DBusTypeWriter
*
*
*/

/**
 * Initialize a write iterator, which is used to write out values in
 * serialized D-Bus format.
 *
 * The type_pos passed in is expected to be inside an already-valid,
 * though potentially empty, type signature. This means that the byte
 * after type_pos must be either #DBUS_TYPE_INVALID (aka nul) or some
 * other valid type. #DBusTypeWriter won't enforce that the signature
 * is already valid (you can append the nul byte at the end if you
 * like), but just be aware that you need the nul byte eventually and
 * #DBusTypeWriter isn't going to write it for you.
 *
 * @param writer the writer to init
 * @param byte_order the byte order to marshal into
 * @param type_str the string to write typecodes into
 * @param type_pos where to insert typecodes
 * @param value_str the string to write values into
 * @param value_pos where to insert values
 *
 */
void
_dbus_type_writer_init (DBusTypeWriter *writer,
                       int byte_order,
                       DBusString *type_str,
                       int type_pos,
                       DBusString *value_str,
                       int value_pos)
{
    writer->byte_order = byte_order;
    writer->type_str = type_str;
    writer->type_pos = type_pos;
    writer->value_str = value_str;
    writer->value_pos = value_pos;
    writer->container_type = DBUS_TYPE_INVALID;
    writer->type_pos_is_expectation = FALSE;
    writer->enabled = TRUE;

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("writer %p init remaining sig '%s'\n", writer,
                  writer->type_str ?
                    _dbus_string_get_const_data_len (writer->type_str,
writer->type_pos, 0) :
                    "unknown");
#endif
}

```

```

}

/**
 * Initialize a write iterator, with the signature to be provided
 * later.
 *
 * @param writer the writer to init
 * @param byte_order the byte order to marshal into
 * @param value_str the string to write values into
 * @param value_pos where to insert values
 *
 */
void
_dbus_type_writer_init_types_delayed (DBusTypeWriter *writer,
                                     int byte_order,
                                     DBusString *value_str,
                                     int value_pos)
{
    _dbus_type_writer_init (writer, byte_order,
                           NULL, 0, value_str, value_pos);
}

/**
 * Adds type string to the writer, if it had none.
 *
 * @param writer the writer to init
 * @param type_str type string to add
 * @param type_pos type position
 *
 */
void
_dbus_type_writer_add_types (DBusTypeWriter *writer,
                             DBusString *type_str,
                             int type_pos)
{
    if (writer->type_str == NULL) /* keeps us from using this as setter
*/
    {
        writer->type_str = type_str;
        writer->type_pos = type_pos;
    }
}

/**
 * Removes type string from the writer.
 *
 * @param writer the writer to remove from
 *
 */
void
_dbus_type_writer_remove_types (DBusTypeWriter *writer)
{
    writer->type_str = NULL;
}

```

```

    writer->type_pos = -1;
}

/**
 * Like _dbus_type_writer_init(), except the type string
 * passed in should correspond to an existing signature that
 * matches what you're going to write out. The writer will
 * check what you write vs. this existing signature.
 *
 * @param writer the writer to init
 * @param byte_order the byte order to marshal into
 * @param type_str the string with signature
 * @param type_pos start of signature
 * @param value_str the string to write values into
 * @param value_pos where to insert values
 */
void
_dbus_type_writer_init_values_only (DBusTypeWriter *writer,
                                   int byte_order,
                                   const DBusString *type_str,
                                   int type_pos,
                                   DBusString *value_str,
                                   int value_pos)
{
    _dbus_type_writer_init (writer, byte_order,
                           (DBusString*)type_str, type_pos,
                           value_str, value_pos);

    writer->type_pos_is_expectation = TRUE;
}

static dbus_bool_t
_dbus_type_writer_write_basic_no_typecode (DBusTypeWriter *writer,
                                           int type,
                                           const void *value)
{
    if (writer->enabled)
        return _dbus_marshal_write_basic (writer->value_str,
                                          writer->value_pos,
                                          type,
                                          value,
                                          writer->byte_order,
                                          &writer->value_pos);
    else
        return TRUE;
}

/* If our parent is an array, things are a little bit complicated.
 *
 * The parent must have a complete element type, such as
 * "i" or "aai" or "(ii)" or "a(ii)". There can't be

```



```

* unclosed parens, or an "a" with no following type.
*
* To recurse, the only allowed operation is to recurse into the
* first type in the element type. So for "i" you can't recurse, for
* "ai" you can recurse into the array, for "(ii)" you can recurse
* into the struct.
*
* If you recurse into the array for "ai", then you must specify
* "i" for the element type of the array you recurse into.
*
* While inside an array at any level, we need to avoid writing to
* type_str, since the type only appears once for the whole array,
* it does not appear for each array element.
*
* While inside an array type_pos points to the expected next
* typecode, rather than the next place we could write a typecode.
*/
static void
writer_recurse_init_and_check (DBusTypeWriter *writer,
                              int             container_type,
                              DBusTypeWriter *sub)
{
    _dbus_type_writer_init (sub,
                           writer->byte_order,
                           writer->type_str,
                           writer->type_pos,
                           writer->value_str,
                           writer->value_pos);

    sub->container_type = container_type;

    if (writer->type_pos_is_expectation ||
        (sub->container_type == DBUS_TYPE_ARRAY || sub->container_type
 == DBUS_TYPE_VARIANT))
        sub->type_pos_is_expectation = TRUE;
    else
        sub->type_pos_is_expectation = FALSE;

    sub->enabled = writer->enabled;

#ifdef DBUS_DISABLE_CHECKS
    if (writer->type_pos_is_expectation && writer->type_str)
    {
        int expected;

        expected = _dbus_first_type_in_signature (writer->type_str,
writer->type_pos);

        if (expected != sub->container_type)
        {
            if (expected != DBUS_TYPE_INVALID)

```

```

        _dbus_warn_check_failed ("Writing an element of type %s,
but the expected type here is %s\n"
                                "The overall signature expected
here was '%s' and we are on byte %d of that signature.\n",
                                _dbus_type_to_string (sub-
>container_type),
                                _dbus_type_to_string (expected),
                                _dbus_string_get_const_data
(writer->type_str), writer->type_pos);
        else
        _dbus_warn_check_failed ("Writing an element of type %s,
but no value is expected here\n"
                                "The overall signature expected
here was '%s' and we are on byte %d of that signature.\n",
                                _dbus_type_to_string (sub-
>container_type),
                                _dbus_string_get_const_data
(writer->type_str), writer->type_pos);

        _dbus_assert_not_reached ("bad array element or variant
content written");
    }
}
#endif /* DBUS_DISABLE_CHECKS */

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p recurse parent %s type_pos = %d
value_pos = %d is_expectation = %d remaining sig '%s' enabled = %d\n",
        writer,
        _dbus_type_to_string (writer->container_type),
        writer->type_pos, writer->value_pos, writer-
>type_pos_is_expectation,
        writer->type_str ?
        _dbus_string_get_const_data_len (writer->type_str,
writer->type_pos, 0) :
        "unknown",
        writer->enabled);
    _dbus_verbose (" type writer %p recurse sub %s type_pos = %d
value_pos = %d is_expectation = %d enabled = %d\n",
        sub,
        _dbus_type_to_string (sub->container_type),
        sub->type_pos, sub->value_pos,
        sub->type_pos_is_expectation,
        sub->enabled);
#endif
}

static dbus_bool_t
write_or_verify_typecode (DBusTypeWriter *writer,
                          int typecode)
{
    /* A subwriter inside an array or variant will have type_pos

```

```

    * pointing to the expected typecode; a writer not inside an array
    * or variant has type_pos pointing to the next place to insert a
    * typecode.
    */
#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("  type writer %p write_or_verify start type_pos = %d
remaining sig '%s' enabled = %d\n",
                  writer, writer->type_pos,
                  writer->type_str ?
                    _dbus_string_get_const_data_len (writer->type_str,
writer->type_pos, 0) :
                    "unknown",
                  writer->enabled);
#endif

    if (writer->type_str == NULL)
        return TRUE;

    if (writer->type_pos_is_expectation)
    {
#ifdef DBUS_DISABLE_CHECKS
        {
            int expected;

            expected = _dbus_string_get_byte (writer->type_str, writer-
>type_pos);

            if (expected != typecode)
            {
                if (expected != DBUS_TYPE_INVALID)
                    _dbus_warn_check_failed ("Array or variant type requires
that type %s be written, but %s was written.\n"
                                             "The overall signature expected
here was '%s' and we are on byte %d of that signature.\n",
                                             _dbus_type_to_string
(expected), _dbus_type_to_string (typecode),
                                             _dbus_string_get_const_data
(writer->type_str), writer->type_pos);
                else
                    _dbus_warn_check_failed ("Array or variant type wasn't
expecting any more values to be written into it, but a value %s was
written.\n"
                                             "The overall signature expected
here was '%s' and we are on byte %d of that signature.\n",
                                             _dbus_type_to_string
(typecode),
                                             _dbus_string_get_const_data
(writer->type_str), writer->type_pos);
                    _dbus_assert_not_reached ("bad type inserted somewhere
inside an array or variant");
            }
        }
    }
}

```

```

#endif /* DBUS_DISABLE_CHECKS */

    /* if immediately inside an array we'd always be appending an
    element,
    * so the expected type doesn't change; if inside a struct or
    something
    * below an array, we need to move through said struct or
    something.
    */
    if (writer->container_type != DBUS_TYPE_ARRAY)
        writer->type_pos += 1;
}
else
{
    if (!_dbus_string_insert_byte (writer->type_str,
                                writer->type_pos,
                                typecode))

        return FALSE;

    writer->type_pos += 1;
}

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p write_or_verify end type_pos = %d
remaining sig '%s'\n",
                writer, writer->type_pos,
                _dbus_string_get_const_data_len (writer->type_str,
writer->type_pos, 0));
#endif

    return TRUE;
}

static dbus_bool_t
writer_recurse_struct_or_dict_entry (DBusTypeWriter *writer,
int begin_char,
const DBusString *contained_type,
int
contained_type_start,
int
contained_type_len,
DBusTypeWriter *sub)
{
    /* FIXME right now contained_type is ignored; we could probably
    * almost trivially fix the code so if it's present we
    * write it out and then set type_pos_is_expectation
    */

    /* Ensure that we'll be able to add alignment padding and the
    typecode */
    if (writer->enabled)
    {

```

```

        if (!_dbus_string_alloc_space (sub->value_str, 8))
            return FALSE;
    }

    if (!write_or_verify_typecode (sub, begin_char))
        _dbus_assert_not_reached ("failed to insert struct typecode after
prealloc");

    if (writer->enabled)
    {
        if (!_dbus_string_insert_bytes (sub->value_str,
                                        sub->value_pos,
                                        _DBUS_ALIGN_VALUE (sub-
>value_pos, 8) - sub->value_pos,
                                        '\0'))
            _dbus_assert_not_reached ("should not have failed to insert
alignment padding for struct");
        sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos, 8);
    }

    return TRUE;
}

static dbus_bool_t
writer_recurse_array (DBusTypeWriter *writer,
                     const DBusString *contained_type,
                     int contained_type_start,
                     int contained_type_len,
                     DBusTypeWriter *sub,
                     dbus_bool_t is_array_append)
{
    dbus_uint32_t value = 0;
    int alignment;
    int aligned;

#ifdef DBUS_DISABLE_CHECKS
    if (writer->container_type == DBUS_TYPE_ARRAY &&
        writer->type_str)
    {
        if (!_dbus_string_equal_substring (contained_type,
                                           contained_type_start,
                                           contained_type_len,
                                           writer->type_str,
                                           writer-
>u.array.element_type_pos + 1))
        {
            _dbus_warn_check_failed ("Writing an array of '%s' but this
is incompatible with the expected type of elements in the parent
array\n",
                                     _dbus_string_get_const_data_len
(contained_type,

```

```

contained_type_start,
contained_type_len));
    _dbus_assert_not_reached ("incompatible type for child
array");
    }
}
#endif /* DBUS_DISABLE_CHECKS */

if (writer->enabled && !is_array_append)
{
    /* 3 pad + 4 bytes for the array length, and 4 bytes possible
padding
    * before array values
    */
    if (!_dbus_string_alloc_space (sub->value_str, 3 + 4 + 4))
        return FALSE;
}

if (writer->type_str != NULL)
{
    sub->type_pos += 1; /* move to point to the element type, since
type_pos
                        * should be the expected type for further
writes
                        */
    sub->u.array.element_type_pos = sub->type_pos;
}

if (!writer->type_pos_is_expectation)
{
    /* sub is a toplevel/outermost array so we need to write the
type data */

    /* alloc space for array typecode, element signature */
    if (!_dbus_string_alloc_space (writer->type_str, 1 +
contained_type_len)
        return FALSE;

    if (!_dbus_string_insert_byte (writer->type_str,
writer->type_pos,
DBUS_TYPE_ARRAY))
        _dbus_assert_not_reached ("failed to insert array typecode
after prealloc");

    if (!_dbus_string_copy_len (contained_type,
contained_type_start,
contained_type_len,
sub->type_str,
sub->u.array.element_type_pos))

```

```

        _dbus_assert_not_reached ("should not have failed to insert
array element typecodes");
    }

    if (writer->type_str != NULL)
    {
        /* If the parent is an array, we hold type_pos pointing at the
array element type;
        * otherwise advance it to reflect the array value we just
recursed into
        */
        if (writer->container_type != DBUS_TYPE_ARRAY)
            writer->type_pos += 1 + contained_type_len;
        else
            _dbus_assert (writer->type_pos_is_expectation); /* because
it's an array */
    }

    if (writer->enabled)
    {
        /* Write (or jump over, if is_array_append) the length */
        sub->u.array.len_pos = _DBUS_ALIGN_VALUE (sub->value_pos, 4);

        if (is_array_append)
        {
            sub->value_pos += 4;
        }
        else
        {
            if (!_dbus_type_writer_write_basic_no_typecode (sub,
DBUS_TYPE_UINT32,
                                                                &value))
                _dbus_assert_not_reached ("should not have failed to
insert array len");
        }

        _dbus_assert (sub->u.array.len_pos == sub->value_pos - 4);

        /* Write alignment padding for array elements
        * Note that we write the padding *even for empty arrays*
        * to avoid wonky special cases
        */
        alignment = element_type_get_alignment (contained_type,
contained_type_start);

        aligned = _DBUS_ALIGN_VALUE (sub->value_pos, alignment);
        if (aligned != sub->value_pos)
        {
            if (!is_array_append)
            {
                if (!_dbus_string_insert_bytes (sub->value_str,
sub->value_pos,

```

```

aligned - sub-
>value_pos,
        '\0'))
        _dbus_assert_not_reached ("should not have failed to
insert alignment padding");
    }

    sub->value_pos = aligned;
}

sub->u.array.start_pos = sub->value_pos;

if (is_array_append)
{
    dbus_uint32_t len;

    _dbus_assert (_DBUS_ALIGN_VALUE (sub->u.array.len_pos, 4) ==
(unsigned) sub->u.array.len_pos);
    len = _dbus_unpack_uint32 (sub->byte_order,
        _dbus_string_get_const_data_len
(sub->value_str,
sub->u.array.len_pos,
4));

    sub->value_pos += len;
}
else
{
    /* not enabled, so we won't write the len_pos; set it to -1 to
so indicate */
    sub->u.array.len_pos = -1;
    sub->u.array.start_pos = sub->value_pos;
}

    _dbus_assert (sub->u.array.len_pos < sub->u.array.start_pos);
    _dbus_assert (is_array_append || sub->u.array.start_pos == sub-
>value_pos);

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p recurse array done remaining
sig '%s' array start_pos = %d len_pos = %d value_pos = %d\n", sub,
        sub->type_str ?
        _dbus_string_get_const_data_len (sub->type_str,
sub->type_pos, 0) :
        "unknown",
        sub->u.array.start_pos, sub->u.array.len_pos,
sub->value_pos);
#endif

```



```

    return TRUE;
}

/* Variant value will normally have:
 * 1 byte signature length not including nul
 * signature typecodes (nul terminated)
 * padding to alignment of contained type
 * body according to signature
 *
 * The signature string can only have a single type
 * in it but that type may be complex/recursive.
 *
 * So a typical variant type with the integer 3 will have these
 * octets:
 * 0x1 'i' '\0' [1 byte padding to alignment boundary] 0x0 0x0 0x0
0x3
 *
 * The main world of hurt for writing out a variant is that the type
 * string is the same string as the value string. Which means
 * inserting to the type string will move the value_pos; and it means
 * that inserting to the type string could break type alignment.
 */
static dbus_bool_t
writer_recurse_variant (DBusTypeWriter *writer,
                       const DBusString *contained_type,
                       int contained_type_start,
                       int contained_type_len,
                       DBusTypeWriter *sub)
{
    int contained_alignment;

    if (writer->enabled)
    {
        /* Allocate space for the worst case, which is 1 byte sig
         * length, nul byte at end of sig, and 7 bytes padding to
         * 8-boundary.
         */
        if (!_dbus_string_alloc_space (sub->value_str,
            contained_type_len + 9))
            return FALSE;
    }

    /* write VARIANT typecode to the parent's type string */
    if (!write_or_verify_typecode (writer, DBUS_TYPE_VARIANT))
        return FALSE;

    /* If not enabled, mark that we have no type_str anymore ... */

    if (!writer->enabled)
    {
        sub->type_str = NULL;
        sub->type_pos = -1;
    }
}

```

```

    return TRUE;
}

/* If we're enabled then continue ... */

if (!_dbus_string_insert_byte (sub->value_str,
                              sub->value_pos,
                              contained_type_len))
    _dbus_assert_not_reached ("should not have failed to insert
variant type sig len");

sub->value_pos += 1;

/* Here we switch over to the expected type sig we're about to write
*/
sub->type_str = sub->value_str;
sub->type_pos = sub->value_pos;

if (!_dbus_string_copy_len (contained_type, contained_type_start,
                            contained_type_len,
                            sub->value_str, sub->value_pos))
    _dbus_assert_not_reached ("should not have failed to insert
variant type sig");

sub->value_pos += contained_type_len;

if (!_dbus_string_insert_byte (sub->value_str,
                              sub->value_pos,
                              DBUS_TYPE_INVALID))
    _dbus_assert_not_reached ("should not have failed to insert
variant type nul termination");

sub->value_pos += 1;

contained_alignment = _dbus_type_get_alignment
(_dbus_first_type_in_signature (contained_type,
contained_type_start));

if (!_dbus_string_insert_bytes (sub->value_str,
                               sub->value_pos,
                               _DBUS_ALIGN_VALUE (sub->value_pos,
contained_alignment) - sub->value_pos,
                               '\0'))
    _dbus_assert_not_reached ("should not have failed to insert
alignment padding for variant body");
sub->value_pos = _DBUS_ALIGN_VALUE (sub->value_pos,
contained_alignment);

return TRUE;
}

```

```

static dbus_bool_t
_dbus_type_writer_recurse_contained_len (DBusTypeWriter *writer,
int
container_type,
const DBusString
*contained_type,
int
contained_type_start,
int
contained_type_len,
DBusTypeWriter *sub,
dbus_bool_t
is_array_append)
{
    writer_recurse_init_and_check (writer, container_type, sub);

    switch (container_type)
    {
        case DBUS_TYPE_STRUCT:
            return writer_recurse_struct_or_dict_entry (writer,
DBUS_STRUCT_BEGIN_CHAR,
contained_type,
contained_type_start, contained_type_len,
sub);

            break;
        case DBUS_TYPE_DICT_ENTRY:
            return writer_recurse_struct_or_dict_entry (writer,
DBUS_DICT_ENTRY_BEGIN_CHAR,
contained_type,
contained_type_start, contained_type_len,
sub);

            break;
        case DBUS_TYPE_ARRAY:
            return writer_recurse_array (writer,
contained_type,
contained_type_start, contained_type_len,
sub, is_array_append);

            break;
        case DBUS_TYPE_VARIANT:
            return writer_recurse_variant (writer,
contained_type,
contained_type_start, contained_type_len,
sub);

            break;
        default:
            _dbus_assert_not_reached ("tried to recurse into type that
doesn't support that");
            return FALSE;
    }
}

```

```

        break;
    }
}

/**
 * Opens a new container and writes out the initial information for
 * that container.
 *
 * @param writer the writer
 * @param container_type the type of the container to open
 * @param contained_type the array element type or variant content
 * type
 * @param contained_type_start position to look for the type
 * @param sub the new sub-writer to write container contents
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_type_writer_recurse (DBusTypeWriter *writer,
                           int container_type,
                           const DBusString *contained_type,
                           int contained_type_start,
                           DBusTypeWriter *sub)
{
    int contained_type_len;

    if (contained_type)
        contained_type_len = find_len_of_complete_type (contained_type,
        contained_type_start);
    else
        contained_type_len = 0;

    return _dbus_type_writer_recurse_contained_len (writer,
        container_type,
        contained_type,
        contained_type_start,
        contained_type_len,
        sub,
        FALSE);
}

/**
 * Append to an existing array. Essentially, the writer will read an
 * existing length at the write location; jump over that length; and
 * write new fields. On unrecurse(), the existing length will be
 * updated.
 *
 * @param writer the writer
 * @param contained_type element type
 * @param contained_type_start position of element type
 * @param sub the subwriter to init
 * @returns #FALSE if no memory
 */

```

```

*/
dbus_bool_t
_dbus_type_writer_append_array (DBusTypeWriter *writer,
                                const DBusString *contained_type,
                                int
contained_type_start,
                                DBusTypeWriter *sub)
{
    int contained_type_len;

    if (contained_type)
        contained_type_len = find_len_of_complete_type (contained_type,
contained_type_start);
    else
        contained_type_len = 0;

    return _dbus_type_writer_recurse_contained_len (writer,
DBUS_TYPE_ARRAY,
                                                    contained_type,
contained_type_start,
                                                    contained_type_len,
                                                    sub,
                                                    TRUE);
}

static int
writer_get_array_len (DBusTypeWriter *writer)
{
    _dbus_assert (writer->container_type == DBUS_TYPE_ARRAY);
    return writer->value_pos - writer->u.array.start_pos;
}

/**
 * Closes a container created by _dbus_type_writer_recurse()
 * and writes any additional information to the values block.
 *
 * @param writer the writer
 * @param sub the sub-writer created by _dbus_type_writer_recurse()
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_type_writer_unrecurse (DBusTypeWriter *writer,
                             DBusTypeWriter *sub)
{
    /* type_pos_is_expectation never gets unset once set, or we'd get
all hosed */
    _dbus_assert (!writer->type_pos_is_expectation ||
                  (writer->type_pos_is_expectation && sub-
>type_pos_is_expectation));

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE

```

```

    _dbus_verbose (" type writer %p unrecurse type_pos = %d value_pos =
%d is_expectation = %d container_type = %s\n",
                  writer, writer->type_pos, writer->value_pos, writer-
>type_pos_is_expectation,
                  _dbus_type_to_string (writer->container_type));
    _dbus_verbose (" type writer %p unrecurse sub type_pos = %d
value_pos = %d is_expectation = %d container_type = %s\n",
                  sub, sub->type_pos, sub->value_pos,
                  sub->type_pos_is_expectation,
                  _dbus_type_to_string (sub->container_type));
#endif

    if (sub->container_type == DBUS_TYPE_STRUCT)
    {
        if (!write_or_verify_typecode (sub, DBUS_STRUCT_END_CHAR))
            return FALSE;
    }
    else if (sub->container_type == DBUS_TYPE_DICT_ENTRY)
    {
        if (!write_or_verify_typecode (sub, DBUS_DICT_ENTRY_END_CHAR))
            return FALSE;
    }
    else if (sub->container_type == DBUS_TYPE_ARRAY)
    {
        if (sub->u.array.len_pos >= 0) /* len_pos == -1 if we weren't
enabled when we passed it */
        {
            dbus_uint32_t len;

            /* Set the array length */
            len = writer_get_array_len (sub);
            _dbus_marshal_set_uint32 (sub->value_str,
                                     sub->u.array.len_pos,
                                     len,
                                     sub->byte_order);
        }
        #if RECURSIVE_MARSHAL_WRITE_TRACE
            _dbus_verbose (" filled in sub array len to %u at len_pos
%d\n",
                          len, sub->u.array.len_pos);
        #endif
    }
    #if RECURSIVE_MARSHAL_WRITE_TRACE
        else
        {
            _dbus_verbose (" not filling in sub array len because we
were disabled when we passed the len\n");
        }
    #endif
}

/* Now get type_pos right for the parent writer. Here are the cases:
*

```

```

* Cases !writer->type_pos_is_expectation:
*   (in these cases we want to update to the new insertion point)
*
* - if we recursed into a STRUCT then we didn't know in advance
*   what the types in the struct would be; so we have to fill in
*   that information now.
*     writer->type_pos = sub->type_pos
*
* - if we recursed into anything else, we knew the full array
*   type, or knew the single typecode marking VARIANT, so
*   writer->type_pos is already correct.
*     writer->type_pos should remain as-is
*
* - note that the parent is never an ARRAY or VARIANT, if it were
*   then type_pos_is_expectation would be TRUE. The parent
*   is thus known to be a toplevel or STRUCT.
*
* Cases where writer->type_pos_is_expectation:
*   (in these cases we want to update to next expected type to
write)
*
* - we recursed from STRUCT into STRUCT and we didn't increment
*   type_pos in the parent just to stay consistent with the
*   !writer->type_pos_is_expectation case (though we could
*   special-case this in recurse_struct instead if we wanted)
*     writer->type_pos = sub->type_pos
*
* - we recursed from STRUCT into ARRAY or VARIANT and type_pos
*   for parent should have been incremented already
*     writer->type_pos should remain as-is
*
* - we recursed from ARRAY into a sub-element, so type_pos in the
*   parent is the element type and should remain the element type
*   for the benefit of the next child element
*     writer->type_pos should remain as-is
*
* - we recursed from VARIANT into its value, so type_pos in the
*   parent makes no difference since there's only one value
*   and we just finished writing it and won't use type_pos again
*     writer->type_pos should remain as-is
*
* For all these, DICT_ENTRY is the same as STRUCT
*/
if (writer->type_str != NULL)
{
    if ((sub->container_type == DBUS_TYPE_STRUCT ||
        sub->container_type == DBUS_TYPE_DICT_ENTRY) &&
        (writer->container_type == DBUS_TYPE_STRUCT ||
         writer->container_type == DBUS_TYPE_DICT_ENTRY ||
         writer->container_type == DBUS_TYPE_INVALID))
    {

```

```

        /* Advance the parent to the next struct field */
        writer->type_pos = sub->type_pos;
    }
}

writer->value_pos = sub->value_pos;

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p unrecursed type_pos = %d value_pos
= %d remaining sig '%s'\n",
        writer, writer->type_pos, writer->value_pos,
        writer->type_str ?
        _dbus_string_get_const_data_len (writer->type_str,
writer->type_pos, 0) :
        "unknown");
#endif

return TRUE;
}

/**
 * Writes out a basic type.
 *
 * @param writer the writer
 * @param type the type to write
 * @param value the address of the value to write
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_type_writer_write_basic (DBusTypeWriter *writer,
                               int type,
                               const void *value)
{
    dbus_bool_t retval;

    /* First ensure that our type realloc will succeed */
    if (!writer->type_pos_is_expectation && writer->type_str != NULL)
    {
        if (!_dbus_string_alloc_space (writer->type_str, 1))
            return FALSE;
    }

    retval = FALSE;

    if (!_dbus_type_writer_write_basic_no_typecode (writer, type,
value))
        goto out;

    if (!write_or_verify_typecode (writer, type))
        _dbus_assert_not_reached ("failed to write typecode after
prealloc");
}

```



```

    retval = TRUE;

    out:
#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p basic type_pos = %d value_pos = %d
is_expectation = %d enabled = %d\n",
                  writer, writer->type_pos, writer->value_pos, writer-
>type_pos_is_expectation,
                  writer->enabled);
#endif

    return retval;
}

/**
 * Writes a block of fixed-length basic values, i.e. those that are
 * both dbus_type_is_fixed() and _dbus_type_is_basic(). The block
 * must be written inside an array.
 *
 * The value parameter should be the address of said array of values,
 * so e.g. if it's an array of double, pass in "const double**"
 *
 * @param writer the writer
 * @param element_type type of stuff in the array
 * @param value address of the array
 * @param n_elements number of elements in the array
 * @returns #FALSE if no memory
 */
dbus_bool_t
_dbus_type_writer_write_fixed_multi (DBusTypeWriter      *writer,
                                     int
element_type,
                                     const void          *value,
                                     int
n_elements)
{
    _dbus_assert (writer->container_type == DBUS_TYPE_ARRAY);
    _dbus_assert (dbus_type_is_fixed (element_type));
    _dbus_assert (writer->type_pos_is_expectation);
    _dbus_assert (n_elements >= 0);

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose (" type writer %p entering fixed multi type_pos = %d
value_pos = %d n_elements %d\n",
                  writer, writer->type_pos, writer->value_pos,
n_elements);
#endif

    if (!write_or_verify_typecode (writer, element_type))
        _dbus_assert_not_reached ("OOM should not happen if only verifying
typecode");
}

```

```

if (writer->enabled)
{
    if (!_dbus_marshal_write_fixed_multi (writer->value_str,
writer->value_pos,
element_type,
value,
n_elements,
writer->byte_order,
&writer->value_pos))

        return FALSE;
}

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
_dbus_verbose (" type writer %p fixed multi written new type_pos =
%d new value_pos = %d n_elements %d\n",
writer, writer->type_pos, writer->value_pos,
n_elements);
#endif

return TRUE;
}

static void
enable_if_after (DBusTypeWriter *writer,
DBusTypeReader *reader,
const DBusTypeReader *start_after)
{
    if (start_after)
    {
        if (!writer->enabled && _dbus_type_reader_greater_than (reader,
start_after))
        {
            _dbus_type_writer_set_enabled (writer, TRUE);
#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
_dbus_verbose ("ENABLING writer %p at %d because reader at
value_pos %d is after reader at value_pos %d\n",
writer, writer->value_pos, reader->value_pos,
start_after->value_pos);
#endif
        }

        _dbus_assert ((!writer->enabled &&
!_dbus_type_reader_greater_than (reader, start_after)) ||
(writer->enabled && _dbus_type_reader_greater_than
(reader, start_after)));
    }
}

static dbus_bool_t
append_fixup (DBusList **fixups,
const DBusArrayLenFixup *fixup)
{

```

```

DBusArrayLenFixup *f;

f = dbus_new (DBusArrayLenFixup, 1);
if (f == NULL)
    return FALSE;

*f = *fixup;

if (!_dbus_list_append (fixups, f))
{
    dbus_free (f);
    return FALSE;
}

_dbus_assert (f->len_pos_in_reader == fixup->len_pos_in_reader);
_dbus_assert (f->new_len == fixup->new_len);

return TRUE;
}

/* This loop is trivial if you ignore all the start_after nonsense,
 * so if you're trying to figure it out, start by ignoring that
 */
static dbus_bool_t
writer_write_reader_helper (DBusTypeWriter      *writer,
                            DBusTypeReader      *reader,
                            const DBusTypeReader *start_after,
                            int                  start_after_new_pos,
                            int                  start_after_new_len,
                            DBusList            **fixups,
                            dbus_bool_t         inside_start_after)
{
    int current_type;

    while ((current_type = _dbus_type_reader_get_current_type (reader))
!= DBUS_TYPE_INVALID)
    {
        if (dbus_type_is_container (current_type))
        {
            DBusTypeReader subreader;
            DBusTypeWriter subwriter;
            const DBusString *sig_str;
            int sig_start;
            int sig_len;
            dbus_bool_t enabled_at_recurse;
            dbus_bool_t past_start_after;
            int reader_array_len_pos;
            int reader_array_start_pos;
            dbus_bool_t this_is_start_after;

            /* type_pos is checked since e.g. in a struct the struct
             * and its first field have the same value_pos.

```

```

    * type_str will differ in reader/start_after for variants
    * where type_str is inside the value_str
    */
if (!inside_start_after && start_after &&
    reader->value_pos == start_after->value_pos &&
    reader->type_str == start_after->type_str &&
    reader->type_pos == start_after->type_pos)
    this_is_start_after = TRUE;
else
    this_is_start_after = FALSE;

_dbus_type_reader_recurse (reader, &subreader);

if (current_type == DBUS_TYPE_ARRAY)
    {
    reader_array_len_pos = ARRAY_READER_LEN_POS
(&subreader);
    reader_array_start_pos = subreader.u.array.start_pos;
    }
else
    {
    /* quiet gcc */
    reader_array_len_pos = -1;
    reader_array_start_pos = -1;
    }

_dbus_type_reader_get_signature (&subreader, &sig_str,
                                &sig_start, &sig_len);

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("about to recurse into %s reader at %d
subreader at %d writer at %d start_after reader at %d write target len
%d inside_start_after = %d this_is_start_after = %d\n",
        _dbus_type_to_string (current_type),
        reader->value_pos,
        subreader.value_pos,
        writer->value_pos,
        start_after ? start_after->value_pos : -1,
        _dbus_string_get_length (writer->value_str),
        inside_start_after, this_is_start_after);
#endif

if (!inside_start_after && !this_is_start_after)
    enable_if_after (writer, &subreader, start_after);
enabled_at_recurse = writer->enabled;
if (!_dbus_type_writer_recurse_contained_len (writer,
current_type,
sig_start, sig_len,
sig_str,
FALSE))
    goto oom;

```

```

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("recursed into subwriter at %d write target
len %d\n",
                  subwriter.value_pos,
                  _dbus_string_get_length
(subwriter.value_str));
#endif

        if (!writer_write_reader_helper (&subwriter, &subreader,
start_after,
start_after_new_len,
start_after_new_pos,
fixups,
inside_start_after ||
this_is_start_after))
            goto oom;

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("about to unrecurse from %s subreader at %d
writer at %d subwriter at %d write target len %d\n",
                  _dbus_type_to_string (current_type),
                  subreader.value_pos,
                  writer->value_pos,
                  subwriter.value_pos,
                  _dbus_string_get_length (writer->value_str));
#endif

    if (!inside_start_after && !this_is_start_after)
        enable_if_after (writer, &subreader, start_after);
    past_start_after = writer->enabled;
    if (!_dbus_type_writer_unrecurse (writer, &subwriter))
        goto oom;

    /* If we weren't enabled when we recursed, we didn't
    * write an array len; if we passed start_after
    * somewhere inside the array, then we need to generate
    * a fixup.
    */
    if (start_after != NULL &&
        !enabled_at_recurse && past_start_after &&
        current_type == DBUS_TYPE_ARRAY &&
        fixups != NULL)
    {
        DBusArrayLenFixup fixup;
        int bytes_written_after_start_after;
        int bytes_before_start_after;
        int old_len;

        /* this subwriter access is moderately unkosher since we
        * already unrecursed, but it works as long as unrecurse
        * doesn't break us on purpose

```

```

        */
        bytes_written_after_start_after = writer_get_array_len
(&subwriter);

        bytes_before_start_after =
            start_after->value_pos - reader_array_start_pos;

        fixup.len_pos_in_reader = reader_array_len_pos;
        fixup.new_len =
            bytes_before_start_after +
            start_after_new_len +
            bytes_written_after_start_after;

        _dbus_assert (_DBUS_ALIGN_VALUE
(fixup.len_pos_in_reader, 4) ==
            (unsigned) fixup.len_pos_in_reader);

        old_len = _dbus_unpack_uint32 (reader->byte_order,
_dbus_string_get_const_data_len (reader->value_str,
fixup.len_pos_in_reader, 4));

        if (old_len != fixup.new_len && !append_fixup (fixups,
&fixup))
            goto oom;

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
        _dbus_verbose ("Generated fixup len_pos_in_reader = %d
new_len = %d reader_array_start_pos = %d start_after->value_pos = %d
bytes_before_start_after = %d start_after_new_len = %d
bytes_written_after_start_after = %d\n",
            fixup.len_pos_in_reader,
            fixup.new_len,
            reader_array_start_pos,
            start_after->value_pos,
            bytes_before_start_after,
            start_after_new_len,
            bytes_written_after_start_after);
#endif
    }
}
else
{
    DBusBasicValue val;

    _dbus_assert (dbus_type_is_basic (current_type));

#ifdef RECURSIVE_MARSHAL_WRITE_TRACE
        _dbus_verbose ("Reading basic value %s at %d\n",
            _dbus_type_to_string (current_type),
            reader->value_pos);

```

```

#endif

    _dbus_type_reader_read_basic (reader, &val);

#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("Writing basic value %s at %d write target
len %d inside_start_after = %d\n",
        _dbus_type_to_string (current_type),
        writer->value_pos,
        _dbus_string_get_length (writer->value_str),
        inside_start_after);
#endif

    if (!inside_start_after)
        enable_if_after (writer, reader, start_after);
    if (!_dbus_type_writer_write_basic (writer, current_type,
&val))
        goto oom;
#if RECURSIVE_MARSHAL_WRITE_TRACE
    _dbus_verbose ("Wrote basic value %s, new value_pos %d write
target len %d\n",
        _dbus_type_to_string (current_type),
        writer->value_pos,
        _dbus_string_get_length (writer->value_str));
#endif
}

    _dbus_type_reader_next (reader);
}

return TRUE;

oom:
    if (fixups)
        apply_and_free_fixups (fixups, NULL); /* NULL for reader to apply
to */

    return FALSE;
}

/*
 * Iterate through all values in the given reader, writing a copy of
 * each value to the writer. The reader will be moved forward to its
 * end position.
 *
 * If a reader start_after is provided, it should be a reader for the
 * same data as the reader to be written. Only values occurring after
 * the value pointed to by start_after will be written to the writer.
 *
 * If start_after is provided, then the copy of the reader will be
 * partial. This means that array lengths will not have been copied.
 * The assumption is that you wrote a new version of the value at
 * start_after to the writer. You have to pass in the start position

```



```

    writer->enabled = enabled != FALSE;
}

/** @} */ /* end of DBusMarshal group */

/* tests in dbus-marshal-recursive-util.c */

File = dbus-marshal-recursive.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-recursive.h  Marshalling routines for recursive types
 *
 * Copyright (C) 2004, 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */

#ifndef DBUS_MARSHAL_RECURSIVE_H
#define DBUS_MARSHAL_RECURSIVE_H

#include <dbus/dbus-protocol.h>
#include <dbus/dbus-list.h>

typedef struct DBusTypeReader      DBusTypeReader;
typedef struct DBusTypeWriter      DBusTypeWriter;
typedef struct DBusTypeReaderClass DBusTypeReaderClass;
typedef struct DBusArrayLenFixup   DBusArrayLenFixup;

/**
 * The type reader is an iterator for reading values from a block of
 * values.
 */

```

```

struct DBusTypeReader
{
    dbus_uint32_t byte_order : 8; /**< byte order of the block */

    dbus_uint32_t finished : 1; /**< marks we're at end iterator for
cases
                                * where we don't have another way to
tell
                                */
    dbus_uint32_t array_len_offset : 3; /**< bytes back from start_pos
that len ends */
    const DBusString *type_str; /**< string containing signature of
block */
    int type_pos; /**< current position in signature */
    const DBusString *value_str; /**< string containing values of block
*/
    int value_pos; /**< current position in values */

    const DBusTypeReaderClass *klass; /**< the vtable for the reader */
    union
    {
        struct {
            int start_pos; /**< for array readers, the start
of the array values */
        } array;
    } u; /**< class-specific data */
};

/**
 * The type writer is an iterator for writing to a block of values.
 */
struct DBusTypeWriter
{
    dbus_uint32_t byte_order : 8; /**< byte order to write
values with */

    dbus_uint32_t container_type : 8; /**< what are we inside?
(e.g. struct, variant, array) */

    dbus_uint32_t type_pos_is_expectation : 1; /**< type_pos can be
either an insertion point for or an expected next type */

    dbus_uint32_t enabled : 1; /**< whether to write values */

    DBusString *type_str; /**< where to write typecodes (or read type
expectations) */
    int type_pos; /**< current pos in type_str */
    DBusString *value_str; /**< where to write values */
    int value_pos; /**< next position to write */

    union
    {

```

```

    struct {
        int start_pos; /**< position of first element in the array */
        int len_pos;   /**< position of length of the array */
        int element_type_pos; /**< position of array element type in
type_str */
    } array;
} u; /**< class-specific data */
};

/**
 * When modifying an existing block of values, array lengths may need
 * to be adjusted; those adjustments are described by this struct.
 */
struct DBusArrayLenFixup
{
    int len_pos_in_reader; /**< where the length was in the original
block */
    int new_len;           /**< the new value of the length in the
written-out block */
};

void          _dbus_type_reader_init
(DBusTypeReader *reader,
byte_order,
DBusString    *type_str,
type_pos,
DBusString    *value_str,
value_pos);
void          _dbus_type_reader_init_types_only
(DBusTypeReader *reader,
DBusString    *type_str,
type_pos);
int          _dbus_type_reader_get_current_type
(DBusTypeReader *reader);
int          _dbus_type_reader_get_element_type
(DBusTypeReader *reader);
int          _dbus_type_reader_get_value_pos
(DBusTypeReader *reader);
void         _dbus_type_reader_read_basic
(DBusTypeReader *reader,
*value);
int          _dbus_type_reader_get_array_length
(DBusTypeReader *reader);

```

```

void      _dbus_type_reader_read_fixed_multi      (const
DBusTypeReader  *reader,
void
*value,
int
*n_elements);
void      _dbus_type_reader_read_raw      (const
DBusTypeReader  *reader,
const
unsigned char  **value_location);
void      _dbus_type_reader_recurse
(DBusTypeReader  *reader,
DBusTypeReader  *subreader);
dbus_bool_t  _dbus_type_reader_next
(DBusTypeReader  *reader);
dbus_bool_t  _dbus_type_reader_has_next      (const
DBusTypeReader  *reader);
void      _dbus_type_reader_get_signature      (const
DBusTypeReader  *reader,
const
DBusString      **str_p,
int
*start_p,
int
*len_p);
dbus_bool_t  _dbus_type_reader_set_basic
(DBusTypeReader  *reader,
const void
*value,
const
DBusTypeReader  *realign_root);
dbus_bool_t  _dbus_type_reader_delete
(DBusTypeReader  *reader,
const
DBusTypeReader  *realign_root);
dbus_bool_t  _dbus_type_reader_equal_values      (const
DBusTypeReader  *lhs,
const
DBusTypeReader  *rhs);
void      _dbus_type_signature_next      (const char
*signature,
int
*type_pos);
void      _dbus_type_writer_init      (DBusTypeWriter
*writer,
int
byte_order,

```

```

*type_str,
type_pos,
*value_str,
value_pos);
void _dbus_type_writer_init_types_delayed (DBusTypeWriter
*writer,
byte_order,
*value_str,
value_pos);
void _dbus_type_writer_add_types (DBusTypeWriter
*writer,
DBusString
*type_str,
int
type_pos);
void _dbus_type_writer_remove_types (DBusTypeWriter
*writer);
void _dbus_type_writer_init_values_only (DBusTypeWriter
*writer,
int
byte_order,
const DBusString
*type_str,
int
type_pos,
DBusString
*value_str,
int
value_pos);
dbus_bool_t _dbus_type_writer_write_basic (DBusTypeWriter
*writer,
int
type,
const void
*value);
dbus_bool_t _dbus_type_writer_write_fixed_multi (DBusTypeWriter
*writer,
int
element_type,
const void
*value,
int
n_elements);
dbus_bool_t _dbus_type_writer_recurse (DBusTypeWriter
*writer,

```

```

int
container_type,
const DBusString
*contained_type,
int
contained_type_start,
DBusTypeWriter
*sub);
dbus_bool_t _dbus_type_writer_unrecurse (DBusTypeWriter
*writer,
DBusTypeWriter
*sub);
dbus_bool_t _dbus_type_writer_append_array (DBusTypeWriter
*writer,
const DBusString
contained_type_start,
int
DBusTypeWriter
*sub);
dbus_bool_t _dbus_type_writer_write_reader (DBusTypeWriter
*writer,
DBusTypeReader
*reader);

#endif /* DBUS_MARSHAL_RECURSIVE_H */

```

File = dbus-marshal-validate-util.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-validate-util.c Would be in dbus-marshal-validate.c,
but only used by tests/bus
*
* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
*/

```

```

* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#ifdef DBUS_BUILD_TESTS

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include "dbus-internals.h"
#include "dbus-marshal-validate.h"
#include "dbus-marshal-recursive.h"

#include "dbus-test.h"
#include <stdio.h>

typedef struct
{
    const char *data;
    DBusValidity expected;
} ValidityTest;

static void
run_validity_tests (const ValidityTest *tests,
                    int n_tests,
                    DBusValidity (* func) (const DBusString*,int,int))
{
    int i;

    for (i = 0; i < n_tests; i++)
    {
        DBusString str;
        DBusValidity v;

        _dbus_string_init_const (&str, tests[i].data);

        v = (*func) (&str, 0, _dbus_string_get_length (&str));

        if (v != tests[i].expected)
        {
            _dbus_warn ("Improper validation result %d for '%s'\n",
                        v, tests[i].data);
            _dbus_assert_not_reached ("test failed");
        }

        ++i;
    }
}

```



```

static const ValidityTest signature_tests[] = {
    { "", DBUS_VALID },
    { "i", DBUS_VALID },
    { "ai", DBUS_VALID },
    { "(i)", DBUS_VALID },
    { "w", DBUS_INVALID_UNKNOWN_TYPECODE },
    { "a", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    { "aaaaaa", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    { "ii(ii)a", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    { "ia", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    /* DBUS_INVALID_SIGNATURE_TOO_LONG, */ /* too hard to test this way
*/
    {
        "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaa",
        DBUS_INVALID_EXCEEDED_MAXIMUM_ARRAY_RECURSION },
    {
        "((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
((((((ii))))))))))))))))))))))))))))))))))))))))))))))))))))))))))",
        DBUS_INVALID_EXCEEDED_MAXIMUM_STRUCT_RECURSION },
    { ")", DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED },
    { "i)", DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED },
    { "a)", DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED },
    { "((", DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED },
    { "(i", DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED },
    { "(iiii", DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED },
    { "(ai", DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED },
    { "()", DBUS_INVALID_STRUCT_HAS_NO_FIELDS },
    { "()", DBUS_INVALID_STRUCT_HAS_NO_FIELDS },
    { "a()", DBUS_INVALID_STRUCT_HAS_NO_FIELDS },
    { "i()", DBUS_INVALID_STRUCT_HAS_NO_FIELDS },
    { "()i", DBUS_INVALID_STRUCT_HAS_NO_FIELDS },
    { "(a)", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    { "a{ia}", DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE },
    { "a{}", DBUS_INVALID_DICT_ENTRY_HAS_NO_FIELDS },
    { "a{aii}", DBUS_INVALID_DICT_KEY_MUST_BE_BASIC_TYPE },
    /* { "a{i}", DBUS_INVALID_DICT_ENTRY_HAS_ONLY_ONE_FIELD }, */
    /* { "{is}", DBUS_INVALID_DICT_ENTRY_NOT_INSIDE_ARRAY }, */
    /* { "a{isi}", DBUS_INVALID_DICT_ENTRY_HAS_TOO_MANY_FIELDS }, */
};

```

```

dbus_bool_t
_dbus_marshal_validate_test (void)
{
    DBusString str;
    int i;

    const char *valid_paths[] = {
        "/",
        "/foo/bar",
        "/foo",
    };
}

```

```

    "/foo/bar/baz"
};
const char *invalid_paths[] = {
    "bar",
    "bar/baz",
    "/foo/bar/",
    "/foo/"
    "foo/",
    "boo//blah",
    "//",
    "///",
    "foo//blah/",
    "Hello World",
    "",
    " ",
    "foo bar"
};

const char *valid_interfaces[] = {
    "org.freedesktop.Foo",
    "Bar.Baz",
    "Blah.Blah.Blah.Blah.Blah",
    "a.b",
    "a.b.c.d.e.f.g",
    "a0.b1.c2.d3.e4.f5.g6",
    "abc123.foo27"
};
const char *invalid_interfaces[] = {
    ".",
    "",
    "..",
    ".Foo.Bar",
    "..Foo.Bar",
    "Foo.Bar.",
    "Foo.Bar..",
    "Foo",
    "9foo.bar.baz",
    "foo.bar..baz",
    "foo.bar...baz",
    "foo.bar.b..blah",
    ":",
    ":0-1",
    "10",
    ":11.34324",
    "0.0.0",
    "0..0",
    "foo.Bar.%",
    "foo.Bar!!",
    "!Foo.bar.bz",
    "foo.$.blah",
    "",
    " ",

```

```

    "foo bar"
};

const char *valid_unique_names[] = {
    ":0",
    ":a",
    ":",
    ":.a",
    ":.1",
    ":0.1",
    ":000.2222",
    ":.blah",
    ":abce.freedesktop.blah"
};

const char *invalid_unique_names[] = {
    //":-",
    "!",
    //":0-10",
    ":blah.",
    ":blah.",
    ":blah..org",
    ":blah.org..",
    ":..blah.org",
    "",
    "  ",
    "foo bar"
};

const char *valid_members[] = {
    "Hello",
    "Bar",
    "foobar",
    "_foobar",
    "foo89"
};

const char *invalid_members[] = {
    "9Hello",
    "10",
    "1",
    "foo-bar",
    "blah.org",
    ".blah",
    "blah.",
    "Hello.",
    "!foo",
    "",
    "  ",
    "foo bar"
};

const char *valid_signatures[] = {

```

```

    "",
    "sss",
    "i",
    "b"
};

const char *invalid_signatures[] = {
    " ",
    "not a valid signature",
    "123",
    ".",
    "(",
    "a{(ii)i}" /* https://bugs.freedesktop.org/show_bug.cgi?id=17803
*/
};

/* Signature with reason */

run_validity_tests (signature_tests, _DBUS_N_ELEMENTS
(signature_tests),
                    _dbus_validate_signature_with_reason);

/* Path validation */
i = 0;
while (i < (int) _DBUS_N_ELEMENTS (valid_paths))
{
    _dbus_string_init_const (&str, valid_paths[i]);

    if (!_dbus_validate_path (&str, 0,
                              _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Path \"%s\" should have been valid\n",
valid_paths[i]);
        _dbus_assert_not_reached ("invalid path");
    }

    ++i;
}

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (invalid_paths))
{
    _dbus_string_init_const (&str, invalid_paths[i]);

    if (_dbus_validate_path (&str, 0,
                              _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Path \"%s\" should have been invalid\n",
invalid_paths[i]);
        _dbus_assert_not_reached ("valid path");
    }
}

```

```

        ++i;
    }

    /* Interface validation */
    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (valid_interfaces))
    {
        _dbus_string_init_const (&str, valid_interfaces[i]);

        if (!_dbus_validate_interface (&str, 0,
                                       _dbus_string_get_length (&str))
            {
                _dbus_warn ("Interface \"%s\" should have been valid\n",
valid_interfaces[i]);
                _dbus_assert_not_reached ("invalid interface");
            }

        ++i;
    }

    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (invalid_interfaces))
    {
        _dbus_string_init_const (&str, invalid_interfaces[i]);

        if (_dbus_validate_interface (&str, 0,
                                       _dbus_string_get_length (&str))
            {
                _dbus_warn ("Interface \"%s\" should have been invalid\n",
invalid_interfaces[i]);
                _dbus_assert_not_reached ("valid interface");
            }

        ++i;
    }

    /* Bus name validation (check that valid interfaces are valid bus
names,
 * and invalid interfaces are invalid services except if they start
with ':')
 */
    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (valid_interfaces))
    {
        _dbus_string_init_const (&str, valid_interfaces[i]);

        if (!_dbus_validate_bus_name (&str, 0,
                                       _dbus_string_get_length (&str))
            {
                _dbus_warn ("Bus name \"%s\" should have been valid\n",
valid_interfaces[i]);
                _dbus_assert_not_reached ("invalid bus name");
            }

```

```

        }

        ++i;
    }

    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (invalid_interfaces))
    {
        if (invalid_interfaces[i][0] != ':')
        {
            _dbus_string_init_const (&str, invalid_interfaces[i]);

            if (_dbus_validate_bus_name (&str, 0,
                _dbus_string_get_length
(&str)))
            {
                _dbus_warn ("Bus name \"%s\" should have been
invalid\n", invalid_interfaces[i]);
                _dbus_assert_not_reached ("valid bus name");
            }
        }

        ++i;
    }

    /* unique name validation */
    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (valid_unique_names))
    {
        _dbus_string_init_const (&str, valid_unique_names[i]);

        if (!_dbus_validate_bus_name (&str, 0,
            _dbus_string_get_length (&str)))
        {
            _dbus_warn ("Bus name \"%s\" should have been valid\n",
valid_unique_names[i]);
            _dbus_assert_not_reached ("invalid unique name");
        }

        ++i;
    }

    i = 0;
    while (i < (int) _DBUS_N_ELEMENTS (invalid_unique_names))
    {
        _dbus_string_init_const (&str, invalid_unique_names[i]);

        if (_dbus_validate_bus_name (&str, 0,
            _dbus_string_get_length (&str)))
        {
            _dbus_warn ("Bus name \"%s\" should have been invalid\n",
invalid_unique_names[i]);

```

```

        _dbus_assert_not_reached ("valid unique name");
    }

    ++i;
}

/* Error name validation (currently identical to interfaces)
*/
i = 0;
while (i < (int) _DBUS_N_ELEMENTS (valid_interfaces))
{
    _dbus_string_init_const (&str, valid_interfaces[i]);

    if (!_dbus_validate_error_name (&str, 0,
                                    _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Error name \"%s\" should have been valid\n",
valid_interfaces[i]);
        _dbus_assert_not_reached ("invalid error name");
    }

    ++i;
}

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (invalid_interfaces))
{
    if (invalid_interfaces[i][0] != ':')
    {
        _dbus_string_init_const (&str, invalid_interfaces[i]);

        if (_dbus_validate_error_name (&str, 0,
                                        _dbus_string_get_length
(&str)))
        {
            _dbus_warn ("Error name \"%s\" should have been
invalid\n", invalid_interfaces[i]);
            _dbus_assert_not_reached ("valid error name");
        }
    }

    ++i;
}

/* Member validation */
i = 0;
while (i < (int) _DBUS_N_ELEMENTS (valid_members))
{
    _dbus_string_init_const (&str, valid_members[i]);

    if (!_dbus_validate_member (&str, 0,

```

```

        _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Member \"%s\" should have been valid\n",
valid_members[i]);
        _dbus_assert_not_reached ("invalid member");
    }

    ++i;
}

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (invalid_members))
{
    _dbus_string_init_const (&str, invalid_members[i]);

    if (_dbus_validate_member (&str, 0,
        _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Member \"%s\" should have been invalid\n",
invalid_members[i]);
        _dbus_assert_not_reached ("valid member");
    }

    ++i;
}

/* Signature validation */
i = 0;
while (i < (int) _DBUS_N_ELEMENTS (valid_signatures))
{
    _dbus_string_init_const (&str, valid_signatures[i]);

    if (!_dbus_validate_signature (&str, 0,
        _dbus_string_get_length (&str)))
    {
        _dbus_warn ("Signature \"%s\" should have been valid\n",
valid_signatures[i]);
        _dbus_assert_not_reached ("invalid signature");
    }

    ++i;
}

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (invalid_signatures))
{
    _dbus_string_init_const (&str, invalid_signatures[i]);

    if (_dbus_validate_signature (&str, 0,
        _dbus_string_get_length (&str)))
    {

```



```

        _dbus_warn ("Signature \"%s\" should have been invalid\n",
invalid_signatures[i]);
        _dbus_assert_not_reached ("valid signature");
    }

    ++i;
}

/* Validate claimed length longer than real length */
_dbus_string_init_const (&str, "abc.efg");
if (_dbus_validate_bus_name (&str, 0, 8))
    _dbus_assert_not_reached ("validated too-long string");
if (_dbus_validate_interface (&str, 0, 8))
    _dbus_assert_not_reached ("validated too-long string");
if (_dbus_validate_error_name (&str, 0, 8))
    _dbus_assert_not_reached ("validated too-long string");

_dbus_string_init_const (&str, "abc");
if (_dbus_validate_member (&str, 0, 4))
    _dbus_assert_not_reached ("validated too-long string");

_dbus_string_init_const (&str, "sss");
if (_dbus_validate_signature (&str, 0, 4))
    _dbus_assert_not_reached ("validated too-long signature");

/* Validate string exceeding max name length */
if (!_dbus_string_init (&str))
    _dbus_assert_not_reached ("no memory");

while (_dbus_string_get_length (&str) <= DBUS_MAXIMUM_NAME_LENGTH)
    if (!_dbus_string_append (&str, "abc.def"))
        _dbus_assert_not_reached ("no memory");

if (_dbus_validate_bus_name (&str, 0, _dbus_string_get_length
(&str)))
    _dbus_assert_not_reached ("validated overmax string");
if (_dbus_validate_interface (&str, 0, _dbus_string_get_length
(&str)))
    _dbus_assert_not_reached ("validated overmax string");
if (_dbus_validate_error_name (&str, 0, _dbus_string_get_length
(&str)))
    _dbus_assert_not_reached ("validated overmax string");

/* overlong member */
_dbus_string_set_length (&str, 0);
while (_dbus_string_get_length (&str) <= DBUS_MAXIMUM_NAME_LENGTH)
    if (!_dbus_string_append (&str, "abc"))
        _dbus_assert_not_reached ("no memory");

if (_dbus_validate_member (&str, 0, _dbus_string_get_length (&str)))
    _dbus_assert_not_reached ("validated overmax string");

```

```

/* overlong unique name */
_dbus_string_set_length (&str, 0);
_dbus_string_append (&str, ":");
while (_dbus_string_get_length (&str) <= DBUS_MAXIMUM_NAME_LENGTH)
    if (!_dbus_string_append (&str, "abc"))
        _dbus_assert_not_reached ("no memory");

    if (_dbus_validate_bus_name (&str, 0, _dbus_string_get_length
(&str)))
        _dbus_assert_not_reached ("validated overmax string");

    _dbus_string_free (&str);

/* Body validation; test basic validation of valid bodies for both
endian */

{
    int sequence;
    DBusString signature;
    DBusString body;

    if (!_dbus_string_init (&signature) || !_dbus_string_init (&body))
        _dbus_assert_not_reached ("oom");

    sequence = 0;
    while (dbus_internal_do_not_use_generate_bodies (sequence,
DBUS_LITTLE_ENDIAN,
                                                    &signature,
&body))
    {
        DBusValidity validity;

        validity = _dbus_validate_body_with_reason (&signature, 0,
DBUS_LITTLE_ENDIAN,
                                                    NULL, &body, 0,
_dbus_string_get_length (&body));
        if (validity != DBUS_VALID)
        {
            _dbus_warn ("invalid code %d expected valid on sequence %d
little endian\n",
                        validity, sequence);
            _dbus_verbose_bytes_of_string (&signature, 0,
_dbus_string_get_length (&signature));
            _dbus_verbose_bytes_of_string (&body, 0,
_dbus_string_get_length (&body));
            _dbus_assert_not_reached ("test failed");
        }

        _dbus_string_set_length (&signature, 0);

```

```

        _dbus_string_set_length (&body, 0);
        ++sequence;
    }

    sequence = 0;
    while (dbus_internal_do_not_use_generate_bodies (sequence,
                                                    DBUS_BIG_ENDIAN,
                                                    &signature,
&body))
    {
        DBusValidity validity;

        validity = _dbus_validate_body_with_reason (&signature, 0,
                                                    DBUS_BIG_ENDIAN,
                                                    NULL, &body, 0,
        _dbus_string_get_length (&body));
        if (validity != DBUS_VALID)
        {
            _dbus_warn ("invalid code %d expected valid on sequence %d
big endian\n",
                        validity, sequence);
            _dbus_verbose_bytes_of_string (&signature, 0,
        _dbus_string_get_length (&signature));
            _dbus_verbose_bytes_of_string (&body, 0,
        _dbus_string_get_length (&body));
            _dbus_assert_not_reached ("test failed");
        }

        _dbus_string_set_length (&signature, 0);
        _dbus_string_set_length (&body, 0);
        ++sequence;
    }

    _dbus_string_free (&signature);
    _dbus_string_free (&body);
}

return TRUE;
}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-marshal-validate.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-validate.c Validation routines for marshaled data
*

```

```

* Copyright (C) 2005 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

```

```

#include <config.h>
#include "dbus-internals.h"
#include "dbus-marshall-validate.h"
#include "dbus-marshall-recursive.h"
#include "dbus-marshall-basic.h"
#include "dbus-signature.h"
#include "dbus-string.h"

```

```
/**
```

```

* @addtogroup DBusMarshal
*
* @{
*/

```

```
/**
```

```

* Verifies that the range of type_str from type_pos to type_end is a
* valid signature. If this function returns #TRUE, it will be safe
* to iterate over the signature with a types-only #DBusTypeReader.
* The range passed in should NOT include the terminating
* nul/DBUS_TYPE_INVALID.
*
* @param type_str the string
* @param type_pos where the typecodes start
* @param len length of typecodes
* @returns #DBUS_VALID if valid, reason why invalid otherwise
*/

```

```
DBusValidity
```

```

_dbus_validate_signature_with_reason (const DBusString *type_str,
                                     int type_pos,

```

```

int len)
{
    const unsigned char *p;
    const unsigned char *end;
    int last;
    int struct_depth;
    int array_depth;
    int dict_entry_depth;
    DBusValidity result;

    int element_count;
    DBusList *element_count_stack;

    result = DBUS_VALID;
    element_count_stack = NULL;

    if (!_dbus_list_append (&element_count_stack, _DBUS_INT_TO_POINTER
(0)))
    {
        result = DBUS_VALIDITY_UNKNOWN_OOM_ERROR;
        goto out;
    }

    _dbus_assert (type_str != NULL);
    _dbus_assert (type_pos < _DBUS_INT32_MAX - len);
    _dbus_assert (len >= 0);
    _dbus_assert (type_pos >= 0);

    if (len > DBUS_MAXIMUM_SIGNATURE_LENGTH)
    {
        result = DBUS_INVALID_SIGNATURE_TOO_LONG;
        goto out;
    }

    p = _dbus_string_get_const_data_len (type_str, type_pos, 0);

    end = _dbus_string_get_const_data_len (type_str, type_pos + len, 0);
    struct_depth = 0;
    array_depth = 0;
    dict_entry_depth = 0;
    last = DBUS_TYPE_INVALID;

    while (p != end)
    {
        switch (*p)
        {
            case DBUS_TYPE_BYTE:
            case DBUS_TYPE_BOOLEAN:
            case DBUS_TYPE_INT16:
            case DBUS_TYPE_UINT16:
            case DBUS_TYPE_INT32:
            case DBUS_TYPE_UINT32:

```

```

case DBUS_TYPE_UNIX_FD:
case DBUS_TYPE_INT64:
case DBUS_TYPE_UINT64:
case DBUS_TYPE_DOUBLE:
case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
case DBUS_TYPE_SIGNATURE:
case DBUS_TYPE_VARIANT:
    break;

case DBUS_TYPE_ARRAY:
    array_depth += 1;
    if (array_depth > DBUS_MAXIMUM_TYPE_RECURSION_DEPTH)
    {
        result = DBUS_INVALID_EXCEEDED_MAXIMUM_ARRAY_RECURSION;
        goto out;
    }
    break;

case DBUS_STRUCT_BEGIN_CHAR:
    struct_depth += 1;

    if (struct_depth > DBUS_MAXIMUM_TYPE_RECURSION_DEPTH)
    {
        result = DBUS_INVALID_EXCEEDED_MAXIMUM_STRUCT_RECURSION;
        goto out;
    }

    if (!_dbus_list_append (&element_count_stack,
                           _DBUS_INT_TO_POINTER (0)))
    {
        result = DBUS_VALIDITY_UNKNOWN_OOM_ERROR;
        goto out;
    }

    break;

case DBUS_STRUCT_END_CHAR:
    if (struct_depth == 0)
    {
        result = DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED;
        goto out;
    }

    if (last == DBUS_STRUCT_BEGIN_CHAR)
    {
        result = DBUS_INVALID_STRUCT_HAS_NO_FIELDS;
        goto out;
    }

    _dbus_list_pop_last (&element_count_stack);

```

```

    struct_depth -= 1;
    break;

case DBUS_DICT_ENTRY_BEGIN_CHAR:
    if (last != DBUS_TYPE_ARRAY)
    {
        result = DBUS_INVALID_DICT_ENTRY_NOT_INSIDE_ARRAY;
        goto out;
    }

    dict_entry_depth += 1;

    if (dict_entry_depth > DBUS_MAXIMUM_TYPE_RECURSION_DEPTH)
    {
        result =
DBUS_INVALID_EXCEEDED_MAXIMUM_DICT_ENTRY_RECURSION;
        goto out;
    }

    if (!_dbus_list_append (&element_count_stack,
                           _DBUS_INT_TO_POINTER (0)))
    {
        result = DBUS_VALIDITY_UNKNOWN_OOM_ERROR;
        goto out;
    }

    break;

case DBUS_DICT_ENTRY_END_CHAR:
    if (dict_entry_depth == 0)
    {
        result = DBUS_INVALID_DICT_ENTRY_ENDED_BUT_NOT_STARTED;
        goto out;
    }

    dict_entry_depth -= 1;

    element_count =
        _DBUS_POINTER_TO_INT (_dbus_list_pop_last
(&element_count_stack));

    if (element_count != 2)
    {
        if (element_count == 0)
            result = DBUS_INVALID_DICT_ENTRY_HAS_NO_FIELDS;
        else if (element_count == 1)
            result = DBUS_INVALID_DICT_ENTRY_HAS_ONLY_ONE_FIELD;
        else
            result = DBUS_INVALID_DICT_ENTRY_HAS_TOO_MANY_FIELDS;

        goto out;
    }
}

```

```

        break;

    case DBUS_TYPE_STRUCT:      /* doesn't appear in signatures */
    case DBUS_TYPE_DICT_ENTRY: /* ditto */
    default:
        result = DBUS_INVALID_UNKNOWN_TYPECODE;
        goto out;
    }

    if (*p != DBUS_TYPE_ARRAY &&
        *p != DBUS_DICT_ENTRY_BEGIN_CHAR &&
        *p != DBUS_STRUCT_BEGIN_CHAR)
    {
        element_count =
            _DBUS_POINTER_TO_INT (_dbus_list_pop_last
            (&element_count_stack));

        ++element_count;

        if (!_dbus_list_append (&element_count_stack,
            _DBUS_INT_TO_POINTER (element_count)))
        {
            result = DBUS_VALIDITY_UNKNOWN_OOM_ERROR;
            goto out;
        }
    }

    if (array_depth > 0)
    {
        if (*p == DBUS_TYPE_ARRAY && p != end)
        {
            const char *p1;
            p1 = p + 1;
            if (*p1 == DBUS_STRUCT_END_CHAR ||
                *p1 == DBUS_DICT_ENTRY_END_CHAR)
            {
                result = DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE;
                goto out;
            }
        }
        else
        {
            array_depth = 0;
        }
    }

    if (last == DBUS_DICT_ENTRY_BEGIN_CHAR)
    {
        if (!(dbus_type_is_valid (*p) && dbus_type_is_basic (*p)))
        {
            result = DBUS_INVALID_DICT_KEY_MUST_BE_BASIC_TYPE;
            goto out;
        }
    }

```



```

        }
    }

    last = *p;
    ++p;
}

if (array_depth > 0)
{
    result = DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE;
    goto out;
}

if (struct_depth > 0)
{
    result = DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED;
    goto out;
}

if (dict_entry_depth > 0)
{
    result = DBUS_INVALID_DICT_ENTRY_STARTED_BUT_NOT_ENDED;
    goto out;
}

_dbus_assert (last != DBUS_TYPE_ARRAY);
_dbus_assert (last != DBUS_STRUCT_BEGIN_CHAR);
_dbus_assert (last != DBUS_DICT_ENTRY_BEGIN_CHAR);

result = DBUS_VALID;

out:
    _dbus_list_clear (&element_count_stack);
    return result;
}

/* note: this function is also used to validate the header's values,
 * since the header is a valid body with a particular signature.
 */
static DBusValidity
validate_body_helper (DBusTypeReader      *reader,
                    int                   byte_order,
                    dbus_bool_t           walk_reader_to_end,
                    int                   total_depth,
                    const unsigned char *p,
                    const unsigned char *end,
                    const unsigned char **new_p)
{
    int current_type;

    /* The spec allows arrays and structs to each nest 32, for total

```

```

* nesting of 2*32. We want to impose the same limit on "dynamic"
* value nesting (not visible in the signature) which is introduced
* by DBUS_TYPE_VARIANT.
*/
if (total_depth > (DBUS_MAXIMUM_TYPE_RECURSION_DEPTH * 2))
{
    return DBUS_INVALID_NESTED_TOO_DEEPLY;
}

while ((current_type = _dbus_type_reader_get_current_type (reader))
!= DBUS_TYPE_INVALID)
{
    const unsigned char *a;
    int alignment;

#ifdef 0
        _dbus_verbose ("    validating value of type %s type reader %p
type_pos %d p %p end %p %d remain\n",
                        _dbus_type_to_string (current_type), reader,
reader->type_pos, p, end,
                        (int) (end - p));
#endif

    /* Guarantee that p has one byte to look at */
    if (p == end)
        return DBUS_INVALID_NOT_ENOUGH_DATA;

    switch (current_type)
    {
        case DBUS_TYPE_BYTE:
            ++p;
            break;

        case DBUS_TYPE_BOOLEAN:
        case DBUS_TYPE_INT16:
        case DBUS_TYPE_UINT16:
        case DBUS_TYPE_INT32:
        case DBUS_TYPE_UINT32:
        case DBUS_TYPE_UNIX_FD:
        case DBUS_TYPE_INT64:
        case DBUS_TYPE_UINT64:
        case DBUS_TYPE_DOUBLE:
            alignment = _dbus_type_get_alignment (current_type);
            a = _DBUS_ALIGN_ADDRESS (p, alignment);
            if (a >= end)
                return DBUS_INVALID_NOT_ENOUGH_DATA;
            while (p != a)
            {
                if (*p != '\\0')
                    return DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
                ++p;
            }
    }

```

```

if (current_type == DBUS_TYPE_BOOLEAN)
{
    dbus_uint32_t v = _dbus_unpack_uint32 (byte_order,
                                           p);
    if (!(v == 0 || v == 1))
        return DBUS_INVALID_BOOLEAN_NOT_ZERO_OR_ONE;
}

p += alignment;
break;

case DBUS_TYPE_ARRAY:
case DBUS_TYPE_STRING:
case DBUS_TYPE_OBJECT_PATH:
{
    dbus_uint32_t claimed_len;

    a = _DBUS_ALIGN_ADDRESS (p, 4);
    if (a + 4 > end)
        return DBUS_INVALID_NOT_ENOUGH_DATA;
    while (p != a)
    {
        if (*p != '\0')
            return DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
        ++p;
    }

    claimed_len = _dbus_unpack_uint32 (byte_order, p);
    p += 4;

    /* p may now be == end */
    _dbus_assert (p <= end);

    if (current_type == DBUS_TYPE_ARRAY)
    {
        int array_elem_type =
            _dbus_type_reader_get_element_type (reader);

        if (!dbus_type_is_valid (array_elem_type))
        {
            return DBUS_INVALID_UNKNOWN_TYPECODE;
        }

        alignment = _dbus_type_get_alignment
            (array_elem_type);

        a = _DBUS_ALIGN_ADDRESS (p, alignment);

        /* a may now be == end */
        if (a > end)
            return DBUS_INVALID_NOT_ENOUGH_DATA;
    }
}

```

```

while (p != a)
{
    if (*p != '\\0')
        return DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
    ++p;
}

if (claimed_len > (unsigned long) (end - p))
    return DBUS_INVALID_LENGTH_OUT_OF_BOUNDS;

if (current_type == DBUS_TYPE_OBJECT_PATH)
{
    DBusString str;
    _dbus_string_init_const_len (&str, p, claimed_len);
    if (!_dbus_validate_path (&str, 0,
        _dbus_string_get_length
(&str)))
        return DBUS_INVALID_BAD_PATH;

    p += claimed_len;
}
else if (current_type == DBUS_TYPE_STRING)
{
    DBusString str;
    _dbus_string_init_const_len (&str, p, claimed_len);
    if (!_dbus_string_validate_utf8 (&str, 0,
        _dbus_string_get_length (&str)))
        return DBUS_INVALID_BAD_UTF8_IN_STRING;

    p += claimed_len;
}
else if (current_type == DBUS_TYPE_ARRAY && claimed_len >
0)
{
    DBusTypeReader sub;
    DBusValidity validity;
    const unsigned char *array_end;
    int array_elem_type;

    if (claimed_len > DBUS_MAXIMUM_ARRAY_LENGTH)
        return DBUS_INVALID_ARRAY_LENGTH_EXCEEDS_MAXIMUM;

    /* Remember that the reader is types only, so we can't
    * use it to iterate over elements. It stays the same
    * for all elements.
    */
    _dbus_type_reader_recurse (reader, &sub);

    array_end = p + claimed_len;

```

```

        array_elem_type = _dbus_type_reader_get_element_type
(reader);

        /* avoid recursive call to validate_body_helper if
this is an array
        * of fixed-size elements
        */
        if (dbus_type_is_fixed (array_elem_type))
        {
            /* booleans need to be handled differently, because
they can
            * have an invalid value
            */
            if (array_elem_type == DBUS_TYPE_BOOLEAN)
            {
                dbus_uint32_t v;
                alignment = _dbus_type_get_alignment
(array_elem_type);

                while (p < array_end)
                {
                    v = _dbus_unpack_uint32 (byte_order, p);

                    if (!(v == 0 || v == 1))
                        return
DBUS_INVALID_BOOLEAN_NOT_ZERO_OR_ONE;

                    p += alignment;
                }
            }
            else
            {
                p = array_end;
            }
        }
        else
        {
            while (p < array_end)
            {
                validity = validate_body_helper (&sub,
byte_order, FALSE,
1,
total_depth +
p, end, &p);

                if (validity != DBUS_VALID)
                    return validity;
            }
        }
    }

```

```

        if (p != array_end)
            return DBUS_INVALID_ARRAY_LENGTH_INCORRECT;
    }

    /* check nul termination */
    if (current_type != DBUS_TYPE_ARRAY)
    {
        if (p == end)
            return DBUS_INVALID_NOT_ENOUGH_DATA;

        if (*p != '\0')
            return DBUS_INVALID_STRING_MISSING_NUL;
        ++p;
    }
}
break;

case DBUS_TYPE_SIGNATURE:
{
    dbus_uint32_t claimed_len;
    DBusString str;
    DBusValidity validity;

    claimed_len = *p;
    ++p;

    /* 1 is for nul termination */
    if (claimed_len + 1 > (unsigned long) (end - p))
        return DBUS_INVALID_SIGNATURE_LENGTH_OUT_OF_BOUNDS;

    _dbus_string_init_const_len (&str, p, claimed_len);
    validity =
        _dbus_validate_signature_with_reason (&str, 0,
        _dbus_string_get_length (&str));

    if (validity != DBUS_VALID)
        return validity;

    p += claimed_len;

    _dbus_assert (p < end);
    if (*p != DBUS_TYPE_INVALID)
        return DBUS_INVALID_SIGNATURE_MISSING_NUL;

    ++p;

    _dbus_verbose ("p = %p end = %p claimed_len %u\n", p, end,
    claimed_len);
}
break;

```

```

case DBUS_TYPE_VARIANT:
    {
        /* 1 byte sig len, sig typecodes, align to
         * contained-type-boundary, values.
         */

        /* In addition to normal signature validation, we need to
be sure
         * the signature contains only a single (possibly
container) type.
         */
        dbus_uint32_t claimed_len;
        DBusString sig;
        DBusTypeReader sub;
        DBusValidity validity;
        int contained_alignment;
        int contained_type;
        DBusValidity reason;

        claimed_len = *p;
        ++p;

        /* + 1 for nul */
        if (claimed_len + 1 > (unsigned long) (end - p))
            return
DBUS_INVALID_VARIANT_SIGNATURE_LENGTH_OUT_OF_BOUNDS;

        _dbus_string_init_const_len (&sig, p, claimed_len);
        reason = _dbus_validate_signature_with_reason (&sig, 0,
        _dbus_string_get_length
(&sig));
        if (!(reason == DBUS_VALID))
        {
            if (reason == DBUS_VALIDITY_UNKNOWN_OOM_ERROR)
                return reason;
            else
                return DBUS_INVALID_VARIANT_SIGNATURE_BAD;
        }

        p += claimed_len;

        if (*p != DBUS_TYPE_INVALID)
            return DBUS_INVALID_VARIANT_SIGNATURE_MISSING_NUL;
        ++p;

        contained_type = _dbus_first_type_in_signature (&sig, 0);
        if (contained_type == DBUS_TYPE_INVALID)
            return DBUS_INVALID_VARIANT_SIGNATURE_EMPTY;

        contained_alignment = _dbus_type_get_alignment
(contained_type);

```

```

a = _DBUS_ALIGN_ADDRESS (p, contained_alignment);
if (a > end)
    return DBUS_INVALID_NOT_ENOUGH_DATA;
while (p != a)
    {
        if (*p != '\0')
            return DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
        ++p;
    }

_dbus_type_reader_init_types_only (&sub, &sig, 0);

_dbus_assert (_dbus_type_reader_get_current_type (&sub) !=
DBUS_TYPE_INVALID);

    validity = validate_body_helper (&sub, byte_order, FALSE,
                                    total_depth + 1,
                                    p, end, &p);

    if (validity != DBUS_VALID)
        return validity;

    if (_dbus_type_reader_next (&sub))
        return
DBUS_INVALID_VARIANT_SIGNATURE_SPECIFIES_MULTIPLE_VALUES;

    _dbus_assert (_dbus_type_reader_get_current_type (&sub) ==
DBUS_TYPE_INVALID);
}
break;

case DBUS_TYPE_DICT_ENTRY:
case DBUS_TYPE_STRUCT:
    {
        DBusTypeReader sub;
        DBusValidity validity;

        a = _DBUS_ALIGN_ADDRESS (p, 8);
        if (a > end)
            return DBUS_INVALID_NOT_ENOUGH_DATA;
        while (p != a)
            {
                if (*p != '\0')
                    return DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL;
                ++p;
            }

        _dbus_type_reader_recurse (reader, &sub);

        validity = validate_body_helper (&sub, byte_order, TRUE,
                                        total_depth + 1,
                                        p, end, &p);

        if (validity != DBUS_VALID)

```



```

        return validity;
    }
    break;

    default:
        _dbus_assert_not_reached ("invalid typecode in supposedly-
validated signature");
        break;
    }

#ifdef 0
    _dbus_verbose (" validated value of type %s type reader %p
type_pos %d p %p end %p %d remain\n",
        _dbus_type_to_string (current_type), reader,
reader->type_pos, p, end,
        (int) (end - p));
#endif

    if (p > end)
    {
        _dbus_verbose ("not enough data!!! p = %p end = %p end-p =
%d\n",
            p, end, (int) (end - p));
        return DBUS_INVALID_NOT_ENOUGH_DATA;
    }

    if (walk_reader_to_end)
        _dbus_type_reader_next (reader);
    else
        break;
}

if (new_p)
    *new_p = p;

return DBUS_VALID;
}

/**
 * Verifies that the range of value_str from value_pos to value_end is
 * a legitimate value of type expected_signature.  If this function
 * returns #TRUE, it will be safe to iterate over the values with
 * #DBusTypeReader. The signature is assumed to be already valid.
 *
 * If bytes_remaining is not #NULL, then leftover bytes will be stored
 * there and #DBUS_VALID returned. If it is #NULL, then
 * #DBUS_INVALID_TOO_MUCH_DATA will be returned if bytes are left
 * over.
 *
 * @param expected_signature the expected types in the value_str
 * @param expected_signature_start where in expected_signature is the
signature

```

```

* @param byte_order the byte order
* @param bytes_remaining place to store leftover bytes
* @param value_str the string containing the body
* @param value_pos where the values start
* @param len length of values after value_pos
* @returns #DBUS_VALID if valid, reason why invalid otherwise
*/
DBusValidity
_dbus_validate_body_with_reason (const DBusString *expected_signature,
                                int
expected_signature_start,
                                int                byte_order,
                                int                *bytes_remaining,
                                const DBusString *value_str,
                                int                value_pos,
                                int                len)
{
    DBusTypeReader reader;
    const unsigned char *p;
    const unsigned char *end;
    DBusValidity validity;

    _dbus_assert (len >= 0);
    _dbus_assert (value_pos >= 0);
    _dbus_assert (value_pos <= _dbus_string_get_length (value_str) -
len);

    _dbus_verbose ("validating body from pos %d len %d sig '%s'\n",
                    value_pos, len, _dbus_string_get_const_data_len
(expected_signature,
expected_signature_start,
                                                                0));

    _dbus_type_reader_init_types_only (&reader,
                                        expected_signature,
expected_signature_start);

    p = _dbus_string_get_const_data_len (value_str, value_pos, len);
    end = p + len;

    validity = validate_body_helper (&reader, byte_order, TRUE, 0, p,
end, &p);
    if (validity != DBUS_VALID)
        return validity;

    if (bytes_remaining)
    {
        *bytes_remaining = end - p;
        return DBUS_VALID;
    }
    else if (p < end)

```

```

        return DBUS_INVALID_TOO_MUCH_DATA;
    else
    {
        _dbus_assert (p == end);
        return DBUS_VALID;
    }
}

/**
 * Determine whether the given character is valid as the first
 * character
 * in a name.
 */
#define VALID_INITIAL_NAME_CHARACTER(c) \
    ( ((c) >= 'A' && (c) <= 'Z') || \
      ((c) >= 'a' && (c) <= 'z') || \
      ((c) == '_' ) )

/**
 * Determine whether the given character is valid as a second or later
 * character in a name
 */
#define VALID_NAME_CHARACTER(c) \
    ( ((c) >= '0' && (c) <= '9') || \
      ((c) >= 'A' && (c) <= 'Z') || \
      ((c) >= 'a' && (c) <= 'z') || \
      ((c) == '_' ) )

/**
 * Checks that the given range of the string is a valid object path
 * name in the D-Bus protocol. Part of the validation ensures that
 * the object path contains only ASCII.
 *
 * @todo this is inconsistent with most of DBusString in that
 * it allows a start,len range that extends past the string end.
 *
 * @todo change spec to disallow more things, such as spaces in the
 * path name
 *
 * @param str the string
 * @param start first byte index to check
 * @param len number of bytes to check
 * @returns #TRUE if the byte range exists and is a valid name
 */
dbus_bool_t
_dbus_validate_path (const DBusString *str,
                    int start,
                    int len)
{
    const unsigned char *s;
    const unsigned char *end;
    const unsigned char *last_slash;

```

```

_dbus_assert (start >= 0);
_dbus_assert (len >= 0);
_dbus_assert (start <= _dbus_string_get_length (str));

if (len > _dbus_string_get_length (str) - start)
    return FALSE;

if (len == 0)
    return FALSE;

s = _dbus_string_get_const_data (str) + start;
end = s + len;

if (*s != '/')
    return FALSE;
last_slash = s;
++s;

while (s != end)
{
    if (*s == '/')
    {
        if ((s - last_slash) < 2)
            return FALSE; /* no empty path components allowed */

        last_slash = s;
    }
    else
    {
        if (_DBUS_UNLIKELY (!VALID_NAME_CHARACTER (*s)))
            return FALSE;
    }

    ++s;
}

if ((end - last_slash) < 2 &&
    len > 1)
    return FALSE; /* trailing slash not allowed unless the string is
"/" */

return TRUE;
}

const char *
_dbus_validity_to_error_message (DBusValidity validity)
{
    switch (validity)
    {
        {
            case DBUS_VALIDITY_UNKNOWN_OOM_ERROR:
return "Out of memory";

```

```
    case DBUS_INVALID_FOR_UNKNOWN_REASON:
return "Unknown reason";
    case DBUS_VALID_BUT_INCOMPLETE:
return "Valid but incomplete";
    case DBUS_VALIDITY_UNKNOWN:
return "Validity unknown";
    case DBUS_VALID:
return "Valid";
    case DBUS_INVALID_UNKNOWN_TYPECODE:
return "Unknown typecode";
    case DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE:
return "Missing array element type";
    case DBUS_INVALID_SIGNATURE_TOO_LONG:
return "Signature is too long";
    case DBUS_INVALID_EXCEEDED_MAXIMUM_ARRAY_RECURSION:
return "Exceeded maximum array recursion";
    case DBUS_INVALID_EXCEEDED_MAXIMUM_STRUCT_RECURSION:
return "Exceeded maximum struct recursion";
    case DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED:
return "Struct ended but not started";
    case DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED:
return "Struct started but not ended";
    case DBUS_INVALID_STRUCT_HAS_NO_FIELDS:
return "Struct has no fields";
    case DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL:
return "Alignment padding not null";
    case DBUS_INVALID_BOOLEAN_NOT_ZERO_OR_ONE:
return "Boolean is not zero or one";
    case DBUS_INVALID_NOT_ENOUGH_DATA:
return "Not enough data";
    case DBUS_INVALID_TOO_MUCH_DATA:
return "Too much data";
    case DBUS_INVALID_BAD_BYTE_ORDER:
return "Bad byte order";
    case DBUS_INVALID_BAD_PROTOCOL_VERSION:
return "Bad protocol version";
    case DBUS_INVALID_BAD_MESSAGE_TYPE:
return "Bad message type";
    case DBUS_INVALID_BAD_SERIAL:
return "Bad serial";
    case DBUS_INVALID_INSANE_FIELDS_ARRAY_LENGTH:
return "Insane fields array length";
    case DBUS_INVALID_INSANE_BODY_LENGTH:
return "Insane body length";
    case DBUS_INVALID_MESSAGE_TOO_LONG:
return "Message too long";
    case DBUS_INVALID_HEADER_FIELD_CODE:
return "Header field code";
    case DBUS_INVALID_HEADER_FIELD_HAS_WRONG_TYPE:
return "Header field has wrong type";
    case DBUS_INVALID_USES_LOCAL_INTERFACE:
return "Uses local interface";
```

```
    case DBUS_INVALID_USES_LOCAL_PATH:
return "Uses local path";
    case DBUS_INVALID_HEADER_FIELD_APPEARS_TWICE:
return "Header field appears twice";
    case DBUS_INVALID_BAD_DESTINATION:
return "Bad destination";
    case DBUS_INVALID_BAD_INTERFACE:
return "Bad interface";
    case DBUS_INVALID_BAD_MEMBER:
return "Bad member";
    case DBUS_INVALID_BAD_ERROR_NAME:
return "Bad error name";
    case DBUS_INVALID_BAD_SENDER:
return "Bad sender";
    case DBUS_INVALID_MISSING_PATH:
return "Missing path";
    case DBUS_INVALID_MISSING_INTERFACE:
return "Missing interface";
    case DBUS_INVALID_MISSING_MEMBER:
return "Missing member";
    case DBUS_INVALID_MISSING_ERROR_NAME:
return "Missing error name";
    case DBUS_INVALID_MISSING_REPLY_SERIAL:
return "Missing reply serial";
    case DBUS_INVALID_LENGTH_OUT_OF_BOUNDS:
return "Length out of bounds";
    case DBUS_INVALID_ARRAY_LENGTH_EXCEEDS_MAXIMUM:
return "Array length exceeds maximum";
    case DBUS_INVALID_BAD_PATH:
return "Bad path";
    case DBUS_INVALID_SIGNATURE_LENGTH_OUT_OF_BOUNDS:
return "Signature length out of bounds";
    case DBUS_INVALID_BAD_UTF8_IN_STRING:
return "Bad utf8 in string";
    case DBUS_INVALID_ARRAY_LENGTH_INCORRECT:
return "Array length incorrect";
    case DBUS_INVALID_VARIANT_SIGNATURE_LENGTH_OUT_OF_BOUNDS:
return "Variant signature length out of bounds";
    case DBUS_INVALID_VARIANT_SIGNATURE_BAD:
return "Variant signature bad";
    case DBUS_INVALID_VARIANT_SIGNATURE_EMPTY:
return "Variant signature empty";
    case DBUS_INVALID_VARIANT_SIGNATURE_SPECIFIES_MULTIPLE_VALUES:
return "Variant signature specifies multiple values";
    case DBUS_INVALID_VARIANT_SIGNATURE_MISSING_NUL:
return "Variant signature missing nul";
    case DBUS_INVALID_STRING_MISSING_NUL:
return "String missing nul";
    case DBUS_INVALID_SIGNATURE_MISSING_NUL:
return "Signature missing nul";
    case DBUS_INVALID_EXCEEDED_MAXIMUM_DICT_ENTRY_RECURSION:
return "Exceeded maximum dict entry recursion";
```

```

    case DBUS_INVALID_DICT_ENTRY_ENDED_BUT_NOT_STARTED:
return "Dict entry ended but not started";
    case DBUS_INVALID_DICT_ENTRY_STARTED_BUT_NOT_ENDED:
return "Dict entry started but not ended";
    case DBUS_INVALID_DICT_ENTRY_HAS_NO_FIELDS:
return "Dict entry has no fields";
    case DBUS_INVALID_DICT_ENTRY_HAS_ONLY_ONE_FIELD:
return "Dict entry has only one field";
    case DBUS_INVALID_DICT_ENTRY_HAS_TOO_MANY_FIELDS:
return "Dict entry has too many fields";
    case DBUS_INVALID_DICT_ENTRY_NOT_INSIDE_ARRAY:
return "Dict entry not inside array";
    case DBUS_INVALID_DICT_KEY_MUST_BE_BASIC_TYPE:
return "Dict key must be basic type";
    case DBUS_INVALID_NESTED_TOO_DEEPLY:
return "Variants cannot be used to create a hugely recursive tree of
values";
    default:
    return "Invalid";
}
}

/**
 * Checks that the given range of the string is a valid interface name
 * in the D-Bus protocol. This includes a length restriction and an
 * ASCII subset, see the specification.
 *
 * @todo this is inconsistent with most of DBusString in that
 * it allows a start,len range that extends past the string end.
 *
 * @param str the string
 * @param start first byte index to check
 * @param len number of bytes to check
 * @returns #TRUE if the byte range exists and is a valid name
 */
dbus_bool_t
_dbus_validate_interface (const DBusString *str,
                        int start,
                        int len)
{
    const unsigned char *s;
    const unsigned char *end;
    const unsigned char *iface;
    const unsigned char *last_dot;

    _dbus_assert (start >= 0);
    _dbus_assert (len >= 0);
    _dbus_assert (start <= _dbus_string_get_length (str));

    if (len > _dbus_string_get_length (str) - start)
        return FALSE;

```

```

if (len > DBUS_MAXIMUM_NAME_LENGTH)
    return FALSE;

if (len == 0)
    return FALSE;

last_dot = NULL;
iface = _dbus_string_get_const_data (str) + start;
end = iface + len;
s = iface;

/* check special cases of first char so it doesn't have to be done
 * in the loop. Note we know len > 0
 */
if (_DBUS_UNLIKELY (*s == '.')) /* disallow starting with a . */
    return FALSE;
else if (_DBUS_UNLIKELY (!VALID_INITIAL_NAME_CHARACTER (*s)))
    return FALSE;
else
    ++s;

while (s != end)
    {
        if (*s == '.')
            {
                if (_DBUS_UNLIKELY ((s + 1) == end))
                    return FALSE;
                else if (_DBUS_UNLIKELY (!VALID_INITIAL_NAME_CHARACTER (*(s
+ 1))))
                    return FALSE;
                last_dot = s;
                ++s; /* we just validated the next char, so skip two */
            }
        else if (_DBUS_UNLIKELY (!VALID_NAME_CHARACTER (*s)))
            {
                return FALSE;
            }

        ++s;
    }

if (_DBUS_UNLIKELY (last_dot == NULL))
    return FALSE;

return TRUE;
}

/**
 * Checks that the given range of the string is a valid member name
 * in the D-Bus protocol. This includes a length restriction, etc.,
 * see the specification.
 */

```



```

* @todo this is inconsistent with most of DBusString in that
* it allows a start,len range that extends past the string end.
*
* @param str the string
* @param start first byte index to check
* @param len number of bytes to check
* @returns #TRUE if the byte range exists and is a valid name
*/
dbus_bool_t
_dbus_validate_member (const DBusString *str,
                      int start,
                      int len)
{
    const unsigned char *s;
    const unsigned char *end;
    const unsigned char *member;

    _dbus_assert (start >= 0);
    _dbus_assert (len >= 0);
    _dbus_assert (start <= _dbus_string_get_length (str));

    if (len > _dbus_string_get_length (str) - start)
        return FALSE;

    if (len > DBUS_MAXIMUM_NAME_LENGTH)
        return FALSE;

    if (len == 0)
        return FALSE;

    member = _dbus_string_get_const_data (str) + start;
    end = member + len;
    s = member;

    /* check special cases of first char so it doesn't have to be done
     * in the loop. Note we know len > 0
     */

    if (_DBUS_UNLIKELY (!VALID_INITIAL_NAME_CHARACTER (*s)))
        return FALSE;
    else
        ++s;

    while (s != end)
    {
        if (_DBUS_UNLIKELY (!VALID_NAME_CHARACTER (*s)))
        {
            return FALSE;
        }

        ++s;
    }
}

```



```

const unsigned char *s;
const unsigned char *end;
const unsigned char *iface;
const unsigned char *last_dot;

_dbus_assert (start >= 0);
_dbus_assert (len >= 0);
_dbus_assert (start <= _dbus_string_get_length (str));

if (len > _dbus_string_get_length (str) - start)
    return FALSE;

if (len > DBUS_MAXIMUM_NAME_LENGTH)
    return FALSE;

if (len == 0)
    return FALSE;

last_dot = NULL;
iface = _dbus_string_get_const_data (str) + start;
end = iface + len;
s = iface;

/* check special cases of first char so it doesn't have to be done
 * in the loop. Note we know len > 0
 */
if (*s == ':')
{
    /* unique name */
    ++s;
    while (s != end)
    {
        if (*s == '.')
        {
            if (_DBUS_UNLIKELY ((s + 1) == end))
                return FALSE;
            if (_DBUS_UNLIKELY (!VALID_BUS_NAME_CHARACTER (*(s + 1))))
                return FALSE;
            ++s; /* we just validated the next char, so skip two */
        }
        else if (_DBUS_UNLIKELY (!VALID_BUS_NAME_CHARACTER (*s)))
        {
            return FALSE;
        }

        ++s;
    }

    return TRUE;
}
else if (_DBUS_UNLIKELY (*s == '.')) /* disallow starting with a .
*/

```

```

    return FALSE;
else if (_DBUS_UNLIKELY (!VALID_INITIAL_BUS_NAME_CHARACTER (*s)))
    return FALSE;
else
    ++s;

while (s != end)
{
    if (*s == '.')
    {
        if (_DBUS_UNLIKELY ((s + 1) == end))
            return FALSE;
        else if (_DBUS_UNLIKELY (!VALID_INITIAL_BUS_NAME_CHARACTER
(*s + 1))))
            return FALSE;
        last_dot = s;
        ++s; /* we just validated the next char, so skip two */
    }
    else if (_DBUS_UNLIKELY (!VALID_BUS_NAME_CHARACTER (*s)))
    {
        return FALSE;
    }

    ++s;
}

if (!is_namespace && _DBUS_UNLIKELY (last_dot == NULL))
    return FALSE;

return TRUE;
}

/**
 * Checks that the given range of the string is a valid bus name in
 * the D-Bus protocol. This includes a length restriction, etc., see
 * the specification.
 *
 * @todo this is inconsistent with most of DBusString in that
 * it allows a start,len range that extends past the string end.
 *
 * @param str the string
 * @param start first byte index to check
 * @param len number of bytes to check
 * @returns #TRUE if the byte range exists and is a valid name
 */
dbus_bool_t
_dbus_validate_bus_name (const DBusString *str,
                        int start,
                        int len)
{
    return _dbus_validate_bus_name_full (str, start, len, FALSE);
}

```

```

/**
 * Checks that the given range of the string is a prefix of a valid
bus name in
 * the D-Bus protocol. Unlike _dbus_validate_bus_name(), this accepts
strings
 * with only one period-separated component.
 *
 * @todo this is inconsistent with most of DBusString in that
 * it allows a start,len range that extends past the string end.
 *
 * @param str the string
 * @param start first byte index to check
 * @param len number of bytes to check
 * @returns #TRUE if the byte range exists and is a valid name
 */
dbus_bool_t
_dbus_validate_bus_namespace (const DBusString *str,
                             int start,
                             int len)
{
    return _dbus_validate_bus_name_full (str, start, len, TRUE);
}

/**
 * Checks that the given range of the string is a valid message type
 * signature in the D-Bus protocol.
 *
 * @todo this is inconsistent with most of DBusString in that
 * it allows a start,len range that extends past the string end.
 *
 * @param str the string
 * @param start first byte index to check
 * @param len number of bytes to check
 * @returns #TRUE if the byte range exists and is a valid signature
 */
dbus_bool_t
_dbus_validate_signature (const DBusString *str,
                          int start,
                          int len)
{
    _dbus_assert (start >= 0);
    _dbus_assert (start <= _dbus_string_get_length (str));
    _dbus_assert (len >= 0);

    if (len > _dbus_string_get_length (str) - start)
        return FALSE;

    return _dbus_validate_signature_with_reason (str, start, len) ==
DBUS_VALID;
}

```

```

/** define _dbus_check_is_valid_path() */
DEFINE_DBUS_NAME_CHECK(path)
/** define _dbus_check_is_valid_interface() */
DEFINE_DBUS_NAME_CHECK(interface)
/** define _dbus_check_is_valid_member() */
DEFINE_DBUS_NAME_CHECK(member)
/** define _dbus_check_is_valid_error_name() */
DEFINE_DBUS_NAME_CHECK(error_name)
/** define _dbus_check_is_valid_bus_name() */
DEFINE_DBUS_NAME_CHECK(bus_name)
/** define _dbus_check_is_valid_signature() */
DEFINE_DBUS_NAME_CHECK(signature)
/** define _dbus_check_is_valid_utf8() */
DEFINE_DBUS_NAME_CHECK(utf8)

/** @} */

/* tests in dbus-marshal-validate-util.c */

```

File = dbus-marshal-validate.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-marshal-validate.h Validation routines for marshaled data
 *
 * Copyright (C) 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *
 */

#ifndef DBUS_MARSHAL_VALIDATE_H
#define DBUS_MARSHAL_VALIDATE_H

```

```

/**
 * @addtogroup DbusMarshal
 *
 * @{
 */

/**
 * This is used rather than a bool for high visibility
 */
typedef enum
{
    DBUS_VALIDATION_MODE_WE_TRUST_THIS_DATA_ABSOLUTELY,
    DBUS_VALIDATION_MODE_DATA_IS_UNTRUSTED
} DbusValidationMode;

/**
 * This is primarily used in unit testing, so we can verify that each
 * invalid message is invalid for the expected reasons. Thus we really
 * want a distinct enum value for every codepath leaving the validator
 * functions. Enum values are specified manually for ease of debugging
 * (so you can see the enum value given a printf)
 */
typedef enum
{
#define _DBUS_NEGATIVE_VALIDITY_COUNT 4
    DBUS_VALIDITY_UNKNOWN_OOM_ERROR = -4, /**< can't determine validity
due to OOM */
    DBUS_INVALID_FOR_UNKNOWN_REASON = -3,
    DBUS_VALID_BUT_INCOMPLETE = -2,
    DBUS_VALIDITY_UNKNOWN = -1,
    DBUS_VALID = 0, /**< the data is valid */
    DBUS_INVALID_UNKNOWN_TYPECODE = 1,
    DBUS_INVALID_MISSING_ARRAY_ELEMENT_TYPE = 2,
    DBUS_INVALID_SIGNATURE_TOO_LONG = 3, /* this one is impossible right
now since
value in a byte
* you can't put a too-long
*/
    DBUS_INVALID_EXCEEDED_MAXIMUM_ARRAY_RECURSION = 4,
    DBUS_INVALID_EXCEEDED_MAXIMUM_STRUCT_RECURSION = 5,
    DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED = 6,
    DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED = 7,
    DBUS_INVALID_STRUCT_HAS_NO_FIELDS = 8,
    DBUS_INVALID_ALIGNMENT_PADDING_NOT_NUL = 9,
    DBUS_INVALID_BOOLEAN_NOT_ZERO_OR_ONE = 10,
    DBUS_INVALID_NOT_ENOUGH_DATA = 11,
    DBUS_INVALID_TOO_MUCH_DATA = 12, /**< trailing junk makes it invalid
*/
    DBUS_INVALID_BAD_BYTE_ORDER = 13,
    DBUS_INVALID_BAD_PROTOCOL_VERSION = 14,
    DBUS_INVALID_BAD_MESSAGE_TYPE = 15,
    DBUS_INVALID_BAD_SERIAL = 16,

```

```

DBUS_INVALID_INSANE_FIELDS_ARRAY_LENGTH = 17,
DBUS_INVALID_INSANE_BODY_LENGTH = 18,
DBUS_INVALID_MESSAGE_TOO_LONG = 19,
DBUS_INVALID_HEADER_FIELD_CODE = 20,
DBUS_INVALID_HEADER_FIELD_HAS_WRONG_TYPE = 21,
DBUS_INVALID_USES_LOCAL_INTERFACE = 22,
DBUS_INVALID_USES_LOCAL_PATH = 23,
DBUS_INVALID_HEADER_FIELD_APPEARS_TWICE = 24,
DBUS_INVALID_BAD_DESTINATION = 25,
DBUS_INVALID_BAD_INTERFACE = 26,
DBUS_INVALID_BAD_MEMBER = 27,
DBUS_INVALID_BAD_ERROR_NAME = 28,
DBUS_INVALID_BAD_SENDER = 29,
DBUS_INVALID_MISSING_PATH = 30,
DBUS_INVALID_MISSING_INTERFACE = 31,
DBUS_INVALID_MISSING_MEMBER = 32,
DBUS_INVALID_MISSING_ERROR_NAME = 33,
DBUS_INVALID_MISSING_REPLY_SERIAL = 34,
DBUS_INVALID_LENGTH_OUT_OF_BOUNDS = 35,
DBUS_INVALID_ARRAY_LENGTH_EXCEEDS_MAXIMUM = 36,
DBUS_INVALID_BAD_PATH = 37,
DBUS_INVALID_SIGNATURE_LENGTH_OUT_OF_BOUNDS = 38,
DBUS_INVALID_BAD_UTF8_IN_STRING = 39,
DBUS_INVALID_ARRAY_LENGTH_INCORRECT = 40,
DBUS_INVALID_VARIANT_SIGNATURE_LENGTH_OUT_OF_BOUNDS = 41,
DBUS_INVALID_VARIANT_SIGNATURE_BAD = 42,
DBUS_INVALID_VARIANT_SIGNATURE_EMPTY = 43,
DBUS_INVALID_VARIANT_SIGNATURE_SPECIFIES_MULTIPLE_VALUES = 44,
DBUS_INVALID_VARIANT_SIGNATURE_MISSING_NUL = 45,
DBUS_INVALID_STRING_MISSING_NUL = 46,
DBUS_INVALID_SIGNATURE_MISSING_NUL = 47,
DBUS_INVALID_EXCEEDED_MAXIMUM_DICT_ENTRY_RECURSION = 48,
DBUS_INVALID_DICT_ENTRY_ENDED_BUT_NOT_STARTED = 49,
DBUS_INVALID_DICT_ENTRY_STARTED_BUT_NOT_ENDED = 50,
DBUS_INVALID_DICT_ENTRY_HAS_NO_FIELDS = 51,
DBUS_INVALID_DICT_ENTRY_HAS_ONLY_ONE_FIELD = 52,
DBUS_INVALID_DICT_ENTRY_HAS_TOO_MANY_FIELDS = 53,
DBUS_INVALID_DICT_ENTRY_NOT_INSIDE_ARRAY = 54,
DBUS_INVALID_DICT_KEY_MUST_BE_BASIC_TYPE = 55,
DBUS_INVALID_MISSING_UNIX_FDS = 56,
DBUS_INVALID_NESTED_TOO_DEEPLY = 57,
DBUS_VALIDITY_LAST
} DBusValidity;

DBusValidity _dbus_validate_signature_with_reason (const DBusString
*type_str,
                                                int
type_pos,
                                                int
len);
DBusValidity _dbus_validate_body_with_reason (const DBusString
*expected_signature,

```



```

expected_signature_start,          int
byte_order,                        int
*bytes_remaining,                  int
*value_str,                         const DBusString
value_pos,                          int
len);                                int

const char *_dbus_validity_to_error_message (DBusValidity validity);

dbus_bool_t _dbus_validate_path      (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_interface (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_member   (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_error_name (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_bus_name  (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_bus_namespace (const DBusString *str,
                                     int                start,
                                     int                len);
dbus_bool_t _dbus_validate_signature (const DBusString *str,
                                     int                start,
                                     int                len);

/* just to have a name consistent with the above: */
#define _dbus_validate_utf8(s,b,e) _dbus_string_validate_utf8 (s, b,
e)

#ifdef DBUS_DISABLE_CHECKS

/* Be sure they don't exist, since we don't want to use them outside
of checks
* and so we want the compile failure.
*/
#define DECLARE_DBUS_NAME_CHECK(what)
#define DEFINE_DBUS_NAME_CHECK(what)

#else /* !DBUS_DISABLE_CHECKS */

/** A name check is used in _dbus_return_if_fail(), it's not suitable

```



```
#endif /* DBUS_MARSHAL_VALIDATE_H */
```

```
File = dbus-memory.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* dbus-memory.c D-Bus memory handling  
*  
* Copyright (C) 2002, 2003 Red Hat Inc.  
*  
* Licensed under the Academic Free License version 2.1  
*  
* This program is free software; you can redistribute it and/or  
modify  
* it under the terms of the GNU General Public License as published  
by  
* the Free Software Foundation; either version 2 of the License, or  
* (at your option) any later version.  
*  
* This program is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program; if not, write to the Free Software  
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA  
*  
*/
```

```
#include <config.h>  
#include "dbus-memory.h"  
#include "dbus-internals.h"  
#include "dbus-sysdeps.h"  
#include "dbus-list.h"  
#include <stdlib.h>
```

```
/**  
* @defgroup DBusMemory Memory Allocation  
* @ingroup DBus  
* @brief dbus_malloc(), dbus_free(), etc.  
*  
* Functions and macros related to allocating and releasing  
* blocks of memory.  
*  
*/
```

```
/**
```

```

* @defgroup DBusMemoryInternals Memory allocation implementation
details
* @ingroup DBusInternals
* @brief internals of dbus_malloc() etc.
*
* Implementation details related to allocating and releasing blocks
* of memory.
*/

/**
* @addtogroup DBusMemory
*
* @{
*/

/**
* @def dbus_new
*
* Safe macro for using dbus_malloc(). Accepts the type
* to allocate and the number of type instances to
* allocate as arguments, and returns a memory block
* cast to the desired type, instead of as a void*.
*
* @param type type name to allocate
* @param count number of instances in the allocated array
* @returns the new memory block or #NULL on failure
*/

/**
* @def dbus_new0
*
* Safe macro for using dbus_malloc0(). Accepts the type
* to allocate and the number of type instances to
* allocate as arguments, and returns a memory block
* cast to the desired type, instead of as a void*.
* The allocated array is initialized to all-bits-zero.
*
* @param type type name to allocate
* @param count number of instances in the allocated array
* @returns the new memory block or #NULL on failure
*/

/**
* @typedef DBusFreeFunction
*
* The type of a function which frees a block of memory.
*
* @param memory the memory to free
*/

/** @} */ /* end of public API docs */

```

```

/**
 * @addtogroup DBusMemoryInternals
 *
 * @{
 */

#ifdef DBUS_BUILD_TESTS
static dbus_bool_t debug_initialized = FALSE;
static int fail_nth = -1;
static size_t fail_size = 0;
static int fail_alloc_counter = _DBUS_INT_MAX;
static int n_failures_per_failure = 1;
static int n_failures_this_failure = 0;
static dbus_bool_t guards = FALSE;
static dbus_bool_t disable_mem_pools = FALSE;
static dbus_bool_t backtrace_on_fail_alloc = FALSE;
static dbus_bool_t malloc_cannot_fail = FALSE;
static DBusAtomic n_blocks_outstanding = {0};

/** value stored in guard padding for debugging buffer overrun */
#define GUARD_VALUE 0xdeadbeef
/** size of the information about the block stored in guard mode */
#define GUARD_INFO_SIZE 8
/** size of the GUARD_VALUE-filled padding after the header info */
#define GUARD_START_PAD 16
/** size of the GUARD_VALUE-filled padding at the end of the block */
#define GUARD_END_PAD 16
/** size of stuff at start of block */
#define GUARD_START_OFFSET (GUARD_START_PAD + GUARD_INFO_SIZE)
/** total extra size over the requested allocation for guard stuff */
#define GUARD_EXTRA_SIZE (GUARD_START_OFFSET + GUARD_END_PAD)

static void
_dbus_initialize_malloc_debug (void)
{
    if (!debug_initialized)
    {
        debug_initialized = TRUE;

        if (_dbus_getenv ("DBUS_MALLOC_FAIL_NTH") != NULL)
        {
            fail_nth = atoi (_dbus_getenv ("DBUS_MALLOC_FAIL_NTH"));
            fail_alloc_counter = fail_nth;
            _dbus_verbose ("Will fail dbus_malloc every %d times\n",
fail_nth);
        }

        if (_dbus_getenv ("DBUS_MALLOC_FAIL_GREATER_THAN") != NULL)
        {
            fail_size = atoi (_dbus_getenv
("DBUS_MALLOC_FAIL_GREATER_THAN"));
            _dbus_verbose ("Will fail mallocs over %ld bytes\n",

```

```

        (long) fail_size);
    }

    if (_dbus_getenv ("DBUS_MALLOC_GUARDS") != NULL)
    {
        guards = TRUE;
        _dbus_verbose ("Will use dbus_malloc guards\n");
    }

    if (_dbus_getenv ("DBUS_DISABLE_MEM_POOLS") != NULL)
    {
        disable_mem_pools = TRUE;
        _dbus_verbose ("Will disable memory pools\n");
    }

    if (_dbus_getenv ("DBUS_MALLOC_BACKTRACES") != NULL)
    {
        backtrace_on_fail_alloc = TRUE;
        _dbus_verbose ("Will backtrace on failing a dbus_malloc\n");
    }

    if (_dbus_getenv ("DBUS_MALLOC_CANNOT_FAIL") != NULL)
    {
        malloc_cannot_fail = TRUE;
        _dbus_verbose ("Will abort if system malloc() and friends
fail\n");
    }
}

/**
 * Whether to turn off mem pools, useful for leak checking.
 *
 * @returns #TRUE if mempools should not be used.
 */
dbus_bool_t
_dbus_disable_mem_pools (void)
{
    _dbus_initialize_malloc_debug ();
    return disable_mem_pools;
}

/**
 * Sets the number of allocations until we simulate a failed
 * allocation. If set to 0, the next allocation to run
 * fails; if set to 1, one succeeds then the next fails; etc.
 * Set to _DBUS_INT_MAX to not fail anything.
 *
 * @param until_next_fail number of successful allocs before one fails
 */
void
_dbus_set_fail_alloc_counter (int until_next_fail)

```

```

{
    _dbus_initialize_malloc_debug ();

    fail_alloc_counter = until_next_fail;

#ifdef 0
    _dbus_verbose ("Set fail alloc counter = %d\n", fail_alloc_counter);
#endif
}

/**
 * Gets the number of successful allocs until we'll simulate
 * a failed alloc.
 *
 * @returns current counter value
 */
int
_dbus_get_fail_alloc_counter (void)
{
    _dbus_initialize_malloc_debug ();

    return fail_alloc_counter;
}

/**
 * Sets how many mallocs to fail when the fail alloc counter reaches
 * 0.
 *
 * @param failures_per_failure number to fail
 */
void
_dbus_set_fail_alloc_failures (int failures_per_failure)
{
    n_failures_per_failure = failures_per_failure;
}

/**
 * Gets the number of failures we'll have when the fail malloc
 * counter reaches 0.
 *
 * @returns number of failures planned
 */
int
_dbus_get_fail_alloc_failures (void)
{
    return n_failures_per_failure;
}

#ifdef DBUS_BUILD_TESTS
/**
 * Called when about to alloc some memory; if
 * it returns #TRUE, then the allocation should

```

```

* fail. If it returns #FALSE, then the allocation
* should not fail.
*
* @returns #TRUE if this alloc should fail
*/
dbus_bool_t
_dbus_decrement_fail_alloc_counter (void)
{
    _dbus_initialize_malloc_debug ();
#ifdef DBUS_WIN_FIXME
    {
        static dbus_bool_t called = 0;

        if (!called)
        {
            _dbus_verbose("TODO: memory allocation testing errors disabled
for now\n");
            called = 1;
        }
        return FALSE;
    }
#endif

    if (fail_alloc_counter <= 0)
    {
        if (backtrace_on_fail_alloc)
            _dbus_print_backtrace ();

        _dbus_verbose ("failure %d\n", n_failures_this_failure);

        n_failures_this_failure += 1;
        if (n_failures_this_failure >= n_failures_per_failure)
        {
            if (fail_nth >= 0)
                fail_alloc_counter = fail_nth;
            else
                fail_alloc_counter = _DBUS_INT_MAX;

            n_failures_this_failure = 0;

            _dbus_verbose ("reset fail alloc counter to %d\n",
fail_alloc_counter);
        }

        return TRUE;
    }
    else
    {
        fail_alloc_counter -= 1;
        return FALSE;
    }
}

```



```

#endif /* DBUS_BUILD_TESTS */

/**
 * Get the number of outstanding malloc()'d blocks.
 *
 * @returns number of blocks
 */
int
_dbus_get_malloc_blocks_outstanding (void)
{
    return _dbus_atomic_get (&n_blocks_outstanding);
}

/**
 * Where the block came from.
 */
typedef enum
{
    SOURCE_UNKNOWN,
    SOURCE_MALLOC,
    SOURCE_REALLOC,
    SOURCE_MALLOC_ZERO,
    SOURCE_REALLOC_NULL
} BlockSource;

static const char*
source_string (BlockSource source)
{
    switch (source)
    {
        case SOURCE_UNKNOWN:
            return "unknown";
        case SOURCE_MALLOC:
            return "malloc";
        case SOURCE_REALLOC:
            return "realloc";
        case SOURCE_MALLOC_ZERO:
            return "malloc0";
        case SOURCE_REALLOC_NULL:
            return "realloc(NULL)";
    }
    _dbus_assert_not_reached ("Invalid malloc block source ID");
    return "invalid!";
}

static void
check_guards (void          *free_block,
              dbus_bool_t overwrite)
{
    if (free_block != NULL)
    {

```

```

    unsigned char *block = ((unsigned char*)free_block) -
GUARD_START_OFFSET;
    size_t requested_bytes = *(dbus_uint32_t*)block;
    BlockSource source = *(dbus_uint32_t*)(block + 4);
    unsigned int i;
    dbus_bool_t failed;

    failed = FALSE;

#ifdef 0
    _dbus_verbose ("Checking %d bytes request from source %s\n",
        requested_bytes, source_string (source));
#endif

    i = GUARD_INFO_SIZE;
    while (i < GUARD_START_OFFSET)
    {
        dbus_uint32_t value = *(dbus_uint32_t*) &block[i];
        if (value != GUARD_VALUE)
        {
            _dbus_warn ("Block of %lu bytes from %s had start guard
value 0x%x at %d expected 0x%x\n",
                (long) requested_bytes, source_string
(source),
                value, i, GUARD_VALUE);
            failed = TRUE;
        }

        i += 4;
    }

    i = GUARD_START_OFFSET + requested_bytes;
    while (i < (GUARD_START_OFFSET + requested_bytes +
GUARD_END_PAD))
    {
        dbus_uint32_t value = *(dbus_uint32_t*) &block[i];
        if (value != GUARD_VALUE)
        {
            _dbus_warn ("Block of %lu bytes from %s had end guard
value 0x%x at %d expected 0x%x\n",
                (long) requested_bytes, source_string
(source),
                value, i, GUARD_VALUE);
            failed = TRUE;
        }

        i += 4;
    }

    /* set memory to anything but nul bytes */
    if (overwrite)
        memset (free_block, 'g', requested_bytes);

```

```

        if (failed)
            _dbus_assert_not_reached ("guard value corruption");
    }
}

static void*
set_guards (void          *real_block,
            size_t        requested_bytes,
            BlockSource source)
{
    unsigned char *block = real_block;
    unsigned int i;

    if (block == NULL)
        return NULL;

    _dbus_assert (GUARD_START_OFFSET + GUARD_END_PAD ==
GUARD_EXTRA_SIZE);

    *((dbus_uint32_t*)block) = requested_bytes;
    *((dbus_uint32_t*)(block + 4)) = source;

    i = GUARD_INFO_SIZE;
    while (i < GUARD_START_OFFSET)
        {
            *((dbus_uint32_t*) &block[i]) = GUARD_VALUE;

            i += 4;
        }

    i = GUARD_START_OFFSET + requested_bytes;
    while (i < (GUARD_START_OFFSET + requested_bytes + GUARD_END_PAD))
        {
            *((dbus_uint32_t*) &block[i]) = GUARD_VALUE;

            i += 4;
        }

    check_guards (block + GUARD_START_OFFSET, FALSE);

    return block + GUARD_START_OFFSET;
}

#endif

/** @} */ /* End of internals docs */

/**
 * @addtogroup DBusMemory
 *

```

```

* @{
*/

/**
* Allocates the given number of bytes, as with standard
* malloc(). Guaranteed to return #NULL if bytes is zero
* on all platforms. Returns #NULL if the allocation fails.
* The memory must be released with dbus_free().
*
* dbus_malloc() memory is NOT safe to free with regular free() from
* the C library. Free it with dbus_free() only.
*
* @param bytes number of bytes to allocate
* @return allocated memory, or #NULL if the allocation fails.
*/
void*
dbus_malloc (size_t bytes)
{
#ifdef DBUS_BUILD_TESTS
    _dbus_initialize_malloc_debug ();

    if (_dbus_decrement_fail_alloc_counter ())
    {
        _dbus_verbose (" FAILING malloc of %ld bytes\n", (long) bytes);
        return NULL;
    }
#endif

    if (bytes == 0) /* some system mallocs handle this, some don't */
        return NULL;
#ifdef DBUS_BUILD_TESTS
    else if (fail_size != 0 && bytes > fail_size)
        return NULL;
    else if (guards)
    {
        void *block;

        block = malloc (bytes + GUARD_EXTRA_SIZE);
        if (block)
        {
            _dbus_atomic_inc (&n_blocks_outstanding);
        }
        else if (malloc_cannot_fail)
        {
            _dbus_warn ("out of memory: malloc (%ld + %ld)\n",
                (long) bytes, (long) GUARD_EXTRA_SIZE);
            _dbus_abort ();
        }

        return set_guards (block, bytes, SOURCE_MALLOC);
    }
#endif
}
#endif

```

```

else
{
    void *mem;
    mem = malloc (bytes);

#ifdef DBUS_BUILD_TESTS
    if (mem)
        {
            _dbus_atomic_inc (&n_blocks_outstanding);
        }
    else if (malloc_cannot_fail)
        {
            _dbus_warn ("out of memory: malloc (%ld)\n", (long) bytes);
            _dbus_abort ();
        }
#endif

    return mem;
}

/**
 * Allocates the given number of bytes, as with standard malloc(), but
 * all bytes are initialized to zero as with calloc(). Guaranteed to
 * return #NULL if bytes is zero on all platforms. Returns #NULL if
the
 * allocation fails. The memory must be released with dbus_free().
 *
 * dbus_malloc0() memory is NOT safe to free with regular free() from
 * the C library. Free it with dbus_free() only.
 *
 * @param bytes number of bytes to allocate
 * @return allocated memory, or #NULL if the allocation fails.
 */
void*
dbus_malloc0 (size_t bytes)
{
#ifdef DBUS_BUILD_TESTS
    _dbus_initialize_malloc_debug ();

    if (_dbus_decrement_fail_alloc_counter ())
        {
            _dbus_verbose (" FAILING malloc0 of %ld bytes\n", (long) bytes);

            return NULL;
        }
#endif

    if (bytes == 0)
        return NULL;
#ifdef DBUS_BUILD_TESTS
    else if (fail_size != 0 && bytes > fail_size)

```

```

    return NULL;
else if (guards)
    {
        void *block;

        block = calloc (bytes + GUARD_EXTRA_SIZE, 1);

        if (block)
            {
                _dbus_atomic_inc (&n_blocks_outstanding);
            }
        else if (malloc_cannot_fail)
            {
                _dbus_warn ("out of memory: calloc (%ld + %ld, 1)\n",
                    (long) bytes, (long) GUARD_EXTRA_SIZE);
                _dbus_abort ();
            }

        return set_guards (block, bytes, SOURCE_MALLOC_ZERO);
    }
#endif
else
    {
        void *mem;
        mem = calloc (bytes, 1);

#ifdef DBUS_BUILD_TESTS
        if (mem)
            {
                _dbus_atomic_inc (&n_blocks_outstanding);
            }
        else if (malloc_cannot_fail)
            {
                _dbus_warn ("out of memory: calloc (%ld)\n", (long) bytes);
                _dbus_abort ();
            }
#endif

        return mem;
    }
}

/**
 * Resizes a block of memory previously allocated by dbus_malloc() or
 * dbus_malloc0(). Guaranteed to free the memory and return #NULL if
bytes
 * is zero on all platforms. Returns #NULL if the resize fails.
 * If the resize fails, the memory is not freed.
 *
 * @param memory block to be resized
 * @param bytes new size of the memory block
 * @return allocated memory, or #NULL if the resize fails.

```

```

*/
void*
dbus_realloc (void *memory,
              size_t bytes)
{
#ifdef DBUS_BUILD_TESTS
    _dbus_initialize_malloc_debug ();

    if (_dbus_decrement_fail_alloc_counter ())
    {
        _dbus_verbose (" FAILING realloc of %ld bytes\n", (long) bytes);

        return NULL;
    }
#endif

    if (bytes == 0) /* guarantee this is safe */
    {
        dbus_free (memory);
        return NULL;
    }
#ifdef DBUS_BUILD_TESTS
    else if (fail_size != 0 && bytes > fail_size)
        return NULL;
    else if (guards)
    {
        if (memory)
        {
            size_t old_bytes;
            void *block;

            check_guards (memory, FALSE);

            block = realloc (((unsigned char*)memory) -
GUARD_START_OFFSET,
                            bytes + GUARD_EXTRA_SIZE);

            if (block == NULL)
            {
                if (malloc_cannot_fail)
                {
                    _dbus_warn ("out of memory: realloc (%p, %ld +
%ld)\n",
                                memory, (long) bytes, (long) GUARD_EXTRA_SIZE);
                    _dbus_abort ();
                }
            }

            return NULL;
        }

        old_bytes = *(dbus_uint32_t*)block;
        if (bytes >= old_bytes)

```

```

        /* old guards shouldn't have moved */
        check_guards (((unsigned char*)block) +
GUARD_START_OFFSET, FALSE);

        return set_guards (block, bytes, SOURCE_REALLOC);
    }
    else
    {
        void *block;

        block = malloc (bytes + GUARD_EXTRA_SIZE);

        if (block)
        {
            _dbus_atomic_inc (&n_blocks_outstanding);
        }
        else if (malloc_cannot_fail)
        {
            _dbus_warn ("out of memory: malloc (%ld + %ld)\n",
                (long) bytes, (long) GUARD_EXTRA_SIZE);
            _dbus_abort ();
        }

        return set_guards (block, bytes, SOURCE_REALLOC_NULL);
    }
}
#endif
else
{
    void *mem;
    mem = realloc (memory, bytes);

#ifdef DBUS_BUILD_TESTS
    if (mem == NULL && malloc_cannot_fail)
    {
        _dbus_warn ("out of memory: malloc (%ld)\n", (long) bytes);
        _dbus_abort ();
    }

    if (memory == NULL && mem != NULL)
        _dbus_atomic_inc (&n_blocks_outstanding);
#endif
    return mem;
}
}

/**
 * Frees a block of memory previously allocated by dbus_malloc() or
 * dbus_malloc0(). If passed #NULL, does nothing.
 *
 * @param memory block to be freed
 */

```



```

void
dbus_free (void *memory)
{
#ifdef DBUS_BUILD_TESTS
    if (guards)
    {
        check_guards (memory, TRUE);
        if (memory)
        {
#ifdef DBUS_DISABLE_ASSERT
            _dbus_atomic_dec (&n_blocks_outstanding);
#else
            dbus_int32_t old_value;

            old_value = _dbus_atomic_dec (&n_blocks_outstanding);
            _dbus_assert (old_value >= 1);
#endif
#endif

            free (((unsigned char*)memory) - GUARD_START_OFFSET);
        }

        return;
    }
#endif

    if (memory) /* we guarantee it's safe to free (NULL) */
    {
#ifdef DBUS_BUILD_TESTS
#ifdef DBUS_DISABLE_ASSERT
            _dbus_atomic_dec (&n_blocks_outstanding);
#else
            dbus_int32_t old_value;

            old_value = _dbus_atomic_dec (&n_blocks_outstanding);
            _dbus_assert (old_value >= 1);
#endif
#endif
            free (memory);
        }
    }

/**
 * Frees a #NULL-terminated array of strings.
 * If passed #NULL, does nothing.
 *
 * @param str_array the array to be freed
 */
void
dbus_free_string_array (char **str_array)
{
    if (str_array)

```

```

    {
        int i;

        i = 0;
        while (str_array[i])
        {
            dbus_free (str_array[i]);
            i++;
        }

        dbus_free (str_array);
    }
}

/** @} */ /* End of public API docs block */

/**
 * @addtogroup DBusMemoryInternals
 *
 * @{
 */

/**
 * _dbus_current_generation is used to track each
 * time that dbus_shutdown() is called, so we can
 * reinit things after it's been called. It is simply
 * incremented each time we shut down.
 */
int _dbus_current_generation = 1;

/**
 * Represents a function to be called on shutdown.
 */
typedef struct ShutdownClosure ShutdownClosure;

/**
 * This struct represents a function to be called on shutdown.
 */
struct ShutdownClosure
{
    ShutdownClosure *next;      /**< Next ShutdownClosure */
    DBusShutdownFunction func; /**< Function to call */
    void *data;                 /**< Data for function */
};

_DBUS_DEFINE_GLOBAL_LOCK (shutdown_funcs);
static ShutdownClosure *registered_globals = NULL;

/**
 * Register a cleanup function to be called exactly once
 * the next time dbus_shutdown() is called.

```

```

*
* @param func the function
* @param data data to pass to the function
* @returns #FALSE on not enough memory
*/
dbus_bool_t
_dbus_register_shutdown_func (DBusShutdownFunction func,
                              void *data)
{
    ShutdownClosure *c;

    c = dbus_new (ShutdownClosure, 1);

    if (c == NULL)
        return FALSE;

    c->func = func;
    c->data = data;

    _DBUS_LOCK (shutdown_funcs);

    c->next = registered_globals;
    registered_globals = c;

    _DBUS_UNLOCK (shutdown_funcs);

    return TRUE;
}

/** @} */ /* End of private API docs block */

/**
 * @addtogroup DBusMemory
 *
 * @{
 */

/**
 * Frees all memory allocated internally by libdbus and
 * reverses the effects of dbus_threads_init(). libdbus keeps internal
 * global variables, for example caches and thread locks, and it
 * can be useful to free these internal data structures.
 *
 * dbus_shutdown() does NOT free memory that was returned
 * to the application. It only returns libdbus-internal
 * data structures.
 *
 * You MUST free all memory and release all reference counts
 * returned to you by libdbus prior to calling dbus_shutdown().
 *
 * You can't continue to use any D-Bus objects, such as connections,

```

```

* that were allocated prior to dbus_shutdown(). You can, however,
* start over; call dbus_threads_init() again, create new connections,
* and so forth.
*
* WARNING: dbus_shutdown() is NOT thread safe, it must be called
* while NO other threads are using D-Bus. (Remember, you have to free
* all D-Bus objects and memory before you call dbus_shutdown(), so no
* thread can be using libdbus.)
*
* The purpose of dbus_shutdown() is to allow applications to get
* clean output from memory leak checkers. dbus_shutdown() may also be
* useful if you want to dlopen() libdbus instead of linking to it,
* and want to be able to unload the library again.
*
* There is absolutely no requirement to call dbus_shutdown() - in
fact,
* most applications won't bother and should not feel guilty.
*
* You have to know that nobody is using libdbus in your application's
* process before you can call dbus_shutdown(). One implication of
this
* is that calling dbus_shutdown() from a library is almost certainly
* wrong, since you don't know what the rest of the app is up to.
*
*/
void
dbus_shutdown (void)
{
    while (registered_globals != NULL)
    {
        ShutdownClosure *c;

        c = registered_globals;
        registered_globals = c->next;

        (* c->func) (c->data);

        dbus_free (c);
    }

    _dbus_current_generation += 1;
}

/** @} */ /** End of public API docs block */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"

/**
 * @ingroup DBusMemoryInternals
 * Unit test for DBusMemory
 * @returns #TRUE on success.

```

```

*/
dbus_bool_t
_dbus_memory_test (void)
{
    dbus_bool_t old_guards;
    void *p;
    size_t size;

    old_guards = guards;
    guards = TRUE;
    p = dbus_malloc (4);
    if (p == NULL)
        _dbus_assert_not_reached ("no memory");
    for (size = 4; size < 256; size += 4)
        {
            p = dbus_realloc (p, size);
            if (p == NULL)
                _dbus_assert_not_reached ("no memory");
        }
    for (size = 256; size != 0; size -= 4)
        {
            p = dbus_realloc (p, size);
            if (p == NULL)
                _dbus_assert_not_reached ("no memory");
        }
    dbus_free (p);
    guards = old_guards;
    return TRUE;
}

#endif

```

File = dbus-memory.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-memory.h  D-Bus memory handling
 *
 * Copyright (C) 2002  Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of

```

```

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_MEMORY_H
#define DBUS_MEMORY_H

#include <dbus/dbus-macros.h>
#include <stddef.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusMemory
 * @{
 */

DBUS_EXPORT
DBUS_MALLOC
DBUS_ALLOC_SIZE(1)
void* dbus_malloc          (size_t bytes);

DBUS_EXPORT
DBUS_MALLOC
DBUS_ALLOC_SIZE(1)
void* dbus_malloc0        (size_t bytes);

DBUS_EXPORT
DBUS_MALLOC
DBUS_ALLOC_SIZE(2)
void* dbus_realloc        (void *memory,
                           size_t bytes);

DBUS_EXPORT
void dbus_free            (void *memory);

#define dbus_new(type, count) ((type*)dbus_malloc (sizeof (type) *
(count))
#define dbus_new0(type, count) ((type*)dbus_malloc0 (sizeof (type) *
(count))

DBUS_EXPORT
void dbus_free_string_array (char **str_array);

```

```

typedef void (* DbusFreeFunction) (void *memory);

DBUS_EXPORT
void dbus_shutdown (void);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_MEMORY_H */

File = dbus-mempool.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-mempool.h Memory pools
 *
 * Copyright (C) 2002, 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#include "dbus-mempool.h"
#include "dbus-internals.h"
#include "dbus-valgrind-internal.h"

/**
 * @defgroup DbusMemPool memory pools
 * @ingroup DbusInternals
 * @brief DbusMemPool object
 */

```

```

* Types and functions related to DBusMemPool. A memory pool is used
* to decrease memory fragmentation/overhead and increase speed for
* blocks of small uniformly-sized objects. The main point is to avoid
* the overhead of a malloc block for each small object, speed is
* secondary.
*/

/**
 * @defgroup DBusMemPoolInternals Memory pool implementation details
 * @ingroup DBusInternals
 * @brief DBusMemPool implementation details
 *
 * The guts of DBusMemPool.
 *
 * @{
 */

/**
 * typedef so DBusFreedElement struct can refer to itself.
 */
typedef struct DBusFreedElement DBusFreedElement;

/**
 * struct representing an element on the free list.
 * We just cast freed elements to this so we can
 * make a list out of them.
 */
struct DBusFreedElement
{
    DBusFreedElement *next; /**< next element of the free list */
};

/**
 * The dummy size of the variable-length "elements"
 * field in DBusMemBlock
 */
#define ELEMENT_PADDING 4

/**
 * Typedef for DBusMemBlock so the struct can recursively
 * point to itself.
 */
typedef struct DBusMemBlock DBusMemBlock;

/**
 * DBusMemBlock object represents a single malloc()-returned
 * block that gets chunked up into objects in the memory pool.
 */
struct DBusMemBlock
{
    DBusMemBlock *next; /**< next block in the list, which is already
used up;

```



```

        *   only saved so we can free all the blocks
        *   when we free the mem pool.
        */

    /* this is a long so that "elements" is aligned */
    long used_so_far;    /**< bytes of this block already allocated as
elements. */

    unsigned char elements[ELEMENT_PADDING]; /**< the block data,
actually allocated to required size */
};

/**
 * Internals fields of DBusMemPool
 */
struct DBusMemPool
{
    int element_size;           /**< size of a single object in the
pool */
    int block_size;           /**< size of most recently
allocated block */
    unsigned int zero_elements : 1; /**< whether to zero-init allocated
elements */

    DBusFreedElement *free_elements; /**< a free list of elements to
recycle */
    DBusMemBlock *blocks;      /**< blocks of memory from malloc()
*/
    int allocated_elements;    /**< Count of outstanding allocated
elements */
};

/** @} */

/**
 * @addtogroup DBusMemPool
 *
 * @{
 */

/**
 * @typedef DBusMemPool
 *
 * Opaque object representing a memory pool. Memory pools allow
 * avoiding per-malloc-block memory overhead when allocating a lot of
 * small objects that are all the same size. They are slightly
 * faster than calling malloc() also.
 */

/**
 * Creates a new memory pool, or returns #NULL on failure. Objects in
 * the pool must be at least sizeof(void*) bytes each, due to the way

```

```

* memory pools work. To avoid creating 64 bit problems, this means at
* least 8 bytes on all platforms, unless you are 4 bytes on 32-bit
* and 8 bytes on 64-bit.
*
* @param element_size size of an element allocated from the pool.
* @param zero_elements whether to zero-initialize elements
* @returns the new pool or #NULL
*/
DBusMemPool*
_dbus_mem_pool_new (int element_size,
                   dbus_bool_t zero_elements)
{
    DBusMemPool *pool;

    pool = dbus_new0 (DBusMemPool, 1);
    if (pool == NULL)
        return NULL;

    /* Make the element size at least 8 bytes. */
    if (element_size < 8)
        element_size = 8;

    /* these assertions are equivalent but the first is more clear
    * to programmers that see it fail.
    */
    _dbus_assert (element_size >= (int) sizeof (void*));
    _dbus_assert (element_size >= (int) sizeof (DBusFreedElement));

    /* align the element size to a pointer boundary so we won't get bus
    * errors under other architectures.
    */
    pool->element_size = _DBUS_ALIGN_VALUE (element_size, sizeof (void
*)));

    pool->zero_elements = zero_elements != FALSE;

    pool->allocated_elements = 0;

    /* pick a size for the first block; it increases
    * for each block we need to allocate. This is
    * actually half the initial block size
    * since _dbus_mem_pool_alloc() unconditionally
    * doubles it prior to creating a new block. */
    pool->block_size = pool->element_size * 8;

    _dbus_assert ((pool->block_size %
                   pool->element_size) == 0);

    VALGRIND_CREATE_MEMPOOL (pool, 0, zero_elements)

    return pool;
}

```

```

/**
 * Frees a memory pool (and all elements allocated from it).
 *
 * @param pool the memory pool.
 */
void
_dbus_mem_pool_free (DBusMemPool *pool)
{
    DBusMemBlock *block;

    VALGRIND_DESTROY_MEMPOOL (pool)

    block = pool->blocks;
    while (block != NULL)
    {
        DBusMemBlock *next = block->next;

        dbus_free (block);

        block = next;
    }

    dbus_free (pool);
}

/**
 * Allocates an object from the memory pool.
 * The object must be freed with _dbus_mem_pool_dealloc().
 *
 * @param pool the memory pool
 * @returns the allocated object or #NULL if no memory.
 */
void*
_dbus_mem_pool_alloc (DBusMemPool *pool)
{
#ifdef DBUS_BUILD_TESTS
    if (_dbus_disable_mem_pools ())
    {
        DBusMemBlock *block;
        int alloc_size;

        /* This is obviously really silly, but it's
         * debug-mode-only code that is compiled out
         * when tests are disabled (_dbus_disable_mem_pools()
         * is a constant expression FALSE so this block
         * should vanish)
         */

        alloc_size = sizeof (DBusMemBlock) - ELEMENT_PADDING +
            pool->element_size;
    }

```

```

if (pool->zero_elements)
    block = dbus_malloc0 (alloc_size);
else
    block = dbus_malloc (alloc_size);

if (block != NULL)
    {
    block->next = pool->blocks;
    pool->blocks = block;
    pool->allocated_elements += 1;

    VALGRIND_MEMPOOL_ALLOC (pool, (void *) &block->elements[0],
        pool->element_size)
    return (void*) &block->elements[0];
    }
else
    return NULL;
}
else
#endif
{
    if (_dbus_decrement_fail_alloc_counter ())
        {
        _dbus_verbose (" FAILING mempool alloc\n");
        return NULL;
        }
    else if (pool->free_elements)
        {
        DBusFreedElement *element = pool->free_elements;

        pool->free_elements = pool->free_elements->next;

        VALGRIND_MEMPOOL_ALLOC (pool, element, pool->element_size)

        if (pool->zero_elements)
            memset (element, '\0', pool->element_size);

        pool->allocated_elements += 1;

        return element;
        }
    else
        {
        void *element;

        if (pool->blocks == NULL ||
            pool->blocks->used_so_far == pool->block_size)
            {
            /* Need a new block */
            DBusMemBlock *block;
            int alloc_size;
#endif DBUS_BUILD_TESTS

```

```

        int saved_counter;
#endif

        if (pool->block_size <= _DBUS_INT_MAX / 4) /* avoid
overflow */
        {
            /* use a larger block size for our next block */
            pool->block_size *= 2;
            _dbus_assert ((pool->block_size %
                pool->element_size) == 0);
        }

        alloc_size = sizeof (DBusMemBlock) - ELEMENT_PADDING +
pool->block_size;

#ifdef DBUS_BUILD_TESTS
        /* We save/restore the counter, so that memory pools
won't
        * cause a given function to have different number of
        * allocations on different invocations. i.e. when
testing
        * we want consistent alloc patterns. So we skip our
        * malloc here for purposes of failed alloc simulation.
        */
        saved_counter = _dbus_get_fail_alloc_counter ();
        _dbus_set_fail_alloc_counter (_DBUS_INT_MAX);
#endif

        if (pool->zero_elements)
            block = dbus_malloc0 (alloc_size);
        else
            block = dbus_malloc (alloc_size);

#ifdef DBUS_BUILD_TESTS
        _dbus_set_fail_alloc_counter (saved_counter);
        _dbus_assert (saved_counter ==
_dbus_get_fail_alloc_counter ());
#endif

        if (block == NULL)
            return NULL;

        block->used_so_far = 0;
        block->next = pool->blocks;
        pool->blocks = block;
    }

    element = &pool->blocks->elements[pool->blocks-
>used_so_far];

    pool->blocks->used_so_far += pool->element_size;

```

```

        pool->allocated_elements += 1;

        VALGRIND_MEMPOOL_ALLOC (pool, element, pool->element_size)
        return element;
    }
}

/**
 * Deallocates an object previously created with
 * _dbus_mem_pool_alloc(). The previous object
 * must have come from this same pool.
 * @param pool the memory pool
 * @param element the element earlier allocated.
 * @returns #TRUE if there are no remaining allocated elements
 */
dbus_bool_t
_dbus_mem_pool_dealloc (DBusMemPool *pool,
                       void *element)
{
    VALGRIND_MEMPOOL_FREE (pool, element)

#ifdef DBUS_BUILD_TESTS
    if (_dbus_disable_mem_pools ())
    {
        DBusMemBlock *block;
        DBusMemBlock *prev;

        /* mmm, fast. ;-) debug-only code, so doesn't matter. */

        prev = NULL;
        block = pool->blocks;

        while (block != NULL)
        {
            if (block->elements == (unsigned char*) element)
            {
                if (prev)
                    prev->next = block->next;
                else
                    pool->blocks = block->next;

                dbus_free (block);

                _dbus_assert (pool->allocated_elements > 0);
                pool->allocated_elements -= 1;

                if (pool->allocated_elements == 0)
                    _dbus_assert (pool->blocks == NULL);

                return pool->blocks == NULL;
            }
        }
    }
#endif
}

```

```

        prev = block;
        block = block->next;
    }

    _dbus_assert_not_reached ("freed nonexistent block");
    return FALSE;
}
else
#endif
{
    DBusFreedElement *freed;

    freed = element;
    /* used for internal mempool administration */
    VALGRIND_MAKE_MEM_UNDEFINED (freed, sizeof (freed));

    freed->next = pool->free_elements;
    pool->free_elements = freed;

    _dbus_assert (pool->allocated_elements > 0);
    pool->allocated_elements -= 1;

    return pool->allocated_elements == 0;
}
}

#ifdef DBUS_ENABLE_STATS
void
_dbus_mem_pool_get_stats (DBusMemPool *pool,
                        dbus_uint32_t *in_use_p,
                        dbus_uint32_t *in_free_list_p,
                        dbus_uint32_t *allocated_p)
{
    DBusMemBlock *block;
    DBusFreedElement *freed;
    dbus_uint32_t in_use = 0;
    dbus_uint32_t in_free_list = 0;
    dbus_uint32_t allocated = 0;

    if (pool != NULL)
    {
        in_use = pool->element_size * pool->allocated_elements;

        for (freed = pool->free_elements; freed != NULL; freed = freed->next)
        {
            in_free_list += pool->element_size;
        }

        for (block = pool->blocks; block != NULL; block = block->next)
        {
            if (block == pool->blocks)

```

```

        allocated += pool->block_size;
    else
        allocated += block->used_so_far;
    }
}

if (in_use_p != NULL)
    *in_use_p = in_use;

if (in_free_list_p != NULL)
    *in_free_list_p = in_free_list;

if (allocated_p != NULL)
    *allocated_p = allocated;
}
#endif /* DBUS_ENABLE_STATS */

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>
#include <time.h>

static void
time_for_size (int size)
{
    int i;
    int j;
    clock_t start;
    clock_t end;
#define FREE_ARRAY_SIZE 512
#define N_ITERATIONS FREE_ARRAY_SIZE * 512
    void *to_free[FREE_ARRAY_SIZE];
    DBusMemPool *pool;

    _dbus_verbose ("Timings for size %d\n", size);

    _dbus_verbose (" malloc\n");

    start = clock ();

    i = 0;
    j = 0;
    while (i < N_ITERATIONS)
    {
        to_free[j] = dbus_malloc (size);
        _dbus_assert (to_free[j] != NULL); /* in a real app of course
this is wrong */

        ++j;

```



```

    if (j == FREE_ARRAY_SIZE)
    {
        j = 0;
        while (j < FREE_ARRAY_SIZE)
        {
            dbus_free (to_free[j]);
            ++j;
        }

        j = 0;
    }

    ++i;
}

end = clock ();

_dbus_verbose ("  created/destroyed %d elements in %g seconds\n",
               N_ITERATIONS, (end - start) / (double)
CLOCKS_PER_SEC);

_dbus_verbose (" mempools\n");

start = clock ();

pool = _dbus_mem_pool_new (size, FALSE);

i = 0;
j = 0;
while (i < N_ITERATIONS)
{
    to_free[j] = _dbus_mem_pool_alloc (pool);
    _dbus_assert (to_free[j] != NULL); /* in a real app of course
this is wrong */

    ++j;

    if (j == FREE_ARRAY_SIZE)
    {
        j = 0;
        while (j < FREE_ARRAY_SIZE)
        {
            _dbus_mem_pool_dealloc (pool, to_free[j]);
            ++j;
        }

        j = 0;
    }

    ++i;
}

```

```

    }

    _dbus_mem_pool_free (pool);

    end = clock ();

    _dbus_verbose ("  created/destroyed %d elements in %g seconds\n",
                  N_ITERATIONS, (end - start) / (double)
CLOCKS_PER_SEC);

    _dbus_verbose (" zeroed malloc\n");

    start = clock ();

    i = 0;
    j = 0;
    while (i < N_ITERATIONS)
    {
        to_free[j] = dbus_malloc0 (size);
        _dbus_assert (to_free[j] != NULL); /* in a real app of course
this is wrong */

        ++j;

        if (j == FREE_ARRAY_SIZE)
        {
            j = 0;
            while (j < FREE_ARRAY_SIZE)
            {
                dbus_free (to_free[j]);
                ++j;
            }

            j = 0;
        }

        ++i;
    }

    end = clock ();

    _dbus_verbose ("  created/destroyed %d elements in %g seconds\n",
                  N_ITERATIONS, (end - start) / (double)
CLOCKS_PER_SEC);

    _dbus_verbose (" zeroed mempools\n");

    start = clock ();

    pool = _dbus_mem_pool_new (size, TRUE);

    i = 0;

```

```

j = 0;
while (i < N_ITERATIONS)
{
    to_free[j] = _dbus_mem_pool_alloc (pool);
    _dbus_assert (to_free[j] != NULL); /* in a real app of course
this is wrong */

    ++j;

    if (j == FREE_ARRAY_SIZE)
    {
        j = 0;
        while (j < FREE_ARRAY_SIZE)
        {
            _dbus_mem_pool_dealloc (pool, to_free[j]);
            ++j;
        }

        j = 0;
    }

    ++i;
}

_dbus_mem_pool_free (pool);

end = clock ();

_dbus_verbose (" created/destroyed %d elements in %g seconds\n",
                N_ITERATIONS, (end - start) / (double)
CLOCKS_PER_SEC);
}

/**
 * @ingroup DBusMemPoolInternals
 * Unit test for DBusMemPool
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_mem_pool_test (void)
{
    int i;
    int element_sizes[] = { 4, 8, 16, 50, 124 };

    i = 0;
    while (i < _DBUS_N_ELEMENTS (element_sizes))
    {
        time_for_size (element_sizes[i]);
        ++i;
    }

    return TRUE;
}

```



```
/* if DBUS_ENABLE_STATS */
void      _dbus_mem_pool_get_stats (DBusMemPool  *pool,
                                   dbus_uint32_t *in_use_p,
                                   dbus_uint32_t *in_free_list_p,
                                   dbus_uint32_t *allocated_p);
```

```
DBUS_END_DECLS
```

```
#endif /* DBUS_MEMPOOL_H */
```

```
File = dbus-message-factory.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message-factory.c Generator of valid and invalid message data
for test suite
```

```
*
* Copyright (C) 2005 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*/
```

```
#include <config.h>
```

```
#ifndef DOXYGEN_SHOULD_SKIP_THIS
```

```
#ifdef DBUS_BUILD_TESTS
#include "dbus-message-factory.h"
#include "dbus-message-private.h"
#include "dbus-signature.h"
#include "dbus-test.h"
#include <stdio.h>
```

```
typedef enum
```

```

    {
        CHANGE_TYPE_ADJUST,
        CHANGE_TYPE_ABSOLUTE
    } ChangeType;

#define BYTE_ORDER_OFFSET 0
#define TYPE_OFFSET 1
#define BODY_LENGTH_OFFSET 4
#define FIELDS_ARRAY_LENGTH_OFFSET 12

static void
iter_recurse (DBusMessageDataIter *iter)
{
    iter->depth += 1;
    _dbus_assert (iter->depth < _DBUS_MESSAGE_DATA_MAX_NESTING);
    _dbus_assert (iter->sequence_nos[iter->depth] >= 0);
}

static int
iter_get_sequence (DBusMessageDataIter *iter)
{
    _dbus_assert (iter->sequence_nos[iter->depth] >= 0);
    return iter->sequence_nos[iter->depth];
}

static void
iter_set_sequence (DBusMessageDataIter *iter,
                  int sequence)
{
    _dbus_assert (sequence >= 0);
    iter->sequence_nos[iter->depth] = sequence;
}

static void
iter_unrecurse (DBusMessageDataIter *iter)
{
    iter->depth -= 1;
    _dbus_assert (iter->depth >= 0);
}

static void
iter_next (DBusMessageDataIter *iter)
{
    iter->sequence_nos[iter->depth] += 1;
}

static dbus_bool_t
iter_first_in_series (DBusMessageDataIter *iter)
{
    int i;

    i = iter->depth;

```

```

while (i < _DBUS_MESSAGE_DATA_MAX_NESTING)
{
    if (iter->sequence_nos[i] != 0)
        return FALSE;
    ++i;
}
return TRUE;
}

typedef dbus_bool_t (* DBusInnerGeneratorFunc) (DBusMessageDataIter
*iter,
                                                DBusMessage
**message_p);
typedef dbus_bool_t (* DBusMessageGeneratorFunc) (DBusMessageDataIter
*iter,
                                                DBusString
*data,
                                                DBusValidity
*expected_validity);

static void
set_reply_serial (DBusMessage *message)
{
    if (message == NULL)
        _dbus_assert_not_reached ("oom");
    if (!dbus_message_set_reply_serial (message, 100))
        _dbus_assert_not_reached ("oom");
}

static dbus_bool_t
generate_trivial_inner (DBusMessageDataIter *iter,
                       DBusMessage
                       **message_p)
{
    DBusMessage *message;

    switch (iter_get_sequence (iter))
    {
        case 0:
            message = dbus_message_new_method_call
("org.freedesktop.TextEditor",
                                                "/foo/bar",
"org.freedesktop.DocumentFactory",
                                                "Create");
            break;
        case 1:
            message = dbus_message_new (DBUS_MESSAGE_TYPE_METHOD_RETURN);
            set_reply_serial (message);
            break;
        case 2:
            message = dbus_message_new_signal ("/foo/bar",

```



```

    _dbus_assert_not_reached ("oom");

byte_order = _dbus_header_get_byte_order (&message->header);

set_reply_serial (message);

if (!_dbus_string_init (&signature) || !_dbus_string_init (&body))
    _dbus_assert_not_reached ("oom");

if (dbus_internal_do_not_use_generate_bodies (iter_get_sequence
(iter),
                                             byte_order,
                                             &signature, &body))
{
    const char *v_SIGNATURE;

    v_SIGNATURE = _dbus_string_get_const_data (&signature);
    if (!_dbus_header_set_field_basic (&message->header,
                                     DBUS_HEADER_FIELD_SIGNATURE,
                                     DBUS_TYPE_SIGNATURE,
                                     &v_SIGNATURE))
        _dbus_assert_not_reached ("oom");

    if (!_dbus_string_move (&body, 0, &message->body, 0))
        _dbus_assert_not_reached ("oom");

    _dbus_marshal_set_uint32 (&message->header.data,
BODY_LENGTH_OFFSET,
                             _dbus_string_get_length (&message-
>body),
                             byte_order);

    *message_p = message;
}
else
{
    dbus_message_unref (message);
    *message_p = NULL;
}

_dbus_string_free (&signature);
_dbus_string_free (&body);

return *message_p != NULL;
}

static void
generate_from_message (DBusString          *data,
                      DBusValidity        *expected_validity,
                      DBusMessage        *message)
{
    dbus_message_set_serial (message, 1);
}

```

```

dbus_message_lock (message);

*expected_validity = DBUS_VALID;

/* move for efficiency, since we'll nuke the message anyway */
if (!_dbus_string_move (&message->header.data, 0,
                      data, 0))
    _dbus_assert_not_reached ("oom");

if (!_dbus_string_copy (&message->body, 0,
                      data, _dbus_string_get_length (data)))
    _dbus_assert_not_reached ("oom");
}

static dbus_bool_t
generate_outer (DBusMessageDataIter *iter,
               DBusString *data,
               DBusValidity *expected_validity,
               DBusInnerGeneratorFunc func)
{
    DBusMessage *message;

    message = NULL;
    if (!( *func)(iter, &message))
        return FALSE;

    iter_next (iter);

    _dbus_assert (message != NULL);

    generate_from_message (data, expected_validity, message);

    dbus_message_unref (message);

    return TRUE;
}

static dbus_bool_t
generate_trivial (DBusMessageDataIter *iter,
                 DBusString *data,
                 DBusValidity *expected_validity)
{
    return generate_outer (iter, data, expected_validity,
                          generate_trivial_inner);
}

static dbus_bool_t
generate_many_bodies (DBusMessageDataIter *iter,
                     DBusString *data,
                     DBusValidity *expected_validity)
{
    return generate_outer (iter, data, expected_validity,

```

```

        generate_many_bodies_inner);
}

static DBusMessage*
simple_method_call (void)
{
    DBusMessage *message;
    /* Keeping this small makes stuff go faster */
    message = dbus_message_new_method_call ("o.b.Q",
                                           "/f/b",
                                           "o.b.Z",
                                           "Fro");

    if (message == NULL)
        _dbus_assert_not_reached ("oom");
    return message;
}

static DBusMessage*
simple_signal (void)
{
    DBusMessage *message;
    message = dbus_message_new_signal ("/f/b",
                                       "o.b.Z",
                                       "Fro");

    if (message == NULL)
        _dbus_assert_not_reached ("oom");
    return message;
}

static DBusMessage*
simple_method_return (void)
{
    DBusMessage *message;
    message = dbus_message_new (DBUS_MESSAGE_TYPE_METHOD_RETURN);
    if (message == NULL)
        _dbus_assert_not_reached ("oom");

    set_reply_serial (message);

    return message;
}

static DBusMessage*
simple_error (void)
{
    DBusMessage *message;
    message = dbus_message_new (DBUS_MESSAGE_TYPE_ERROR);
    if (message == NULL)
        _dbus_assert_not_reached ("oom");

    if (!dbus_message_set_error_name (message, "foo.bar"))
        _dbus_assert_not_reached ("oom");
}

```

```

    set_reply_serial (message);

    return message;
}

static DBusMessage*
message_with_nesting_levels (int levels)
{
    DBusMessage *message;
    dbus_int32_t v_INT32;
    DBusMessageIter *parents;
    DBusMessageIter *children;
    int i;

    /* If levels is higher it breaks sig_refcount in DBusMessageRealIter
     * in dbus-message.c, this assert is just to help you know you need
     * to fix that if you hit it
     */
    _dbus_assert (levels < 256);

    parents = dbus_new(DBusMessageIter, levels + 1);
    children = dbus_new(DBusMessageIter, levels + 1);

    v_INT32 = 42;
    message = simple_method_call ();

    i = 0;
    dbus_message_iter_init_append (message, &parents[i]);
    while (i < levels)
    {
        dbus_message_iter_open_container (&parents[i],
        DBUS_TYPE_VARIANT,
        i == (levels - 1) ?
        DBUS_TYPE_INT32_AS_STRING :
        DBUS_TYPE_VARIANT_AS_STRING,
        &children[i]);

        ++i;
        parents[i] = children[i-1];
    }
    --i;
    dbus_message_iter_append_basic (&children[i], DBUS_TYPE_INT32,
    &v_INT32);
    while (i >= 0)
    {
        dbus_message_iter_close_container (&parents[i], &children[i]);
        --i;
    }

    dbus_free (parents);
    dbus_free (children);
}

```



```

    _dbus_assert_not_reached ("oom");

    i = 0;
    while (i < (DBUS_MAXIMUM_TYPE_RECURSION_DEPTH + 1))
    {
        long_sig[i] = DBUS_TYPE_ARRAY;
        ++i;
    }
    long_sig[i] = DBUS_TYPE_INVALID;

    v_STRING = long_sig;
    if (!_dbus_header_set_field_basic (&message->header,
                                      DBUS_HEADER_FIELD_SIGNATURE,
                                      DBUS_TYPE_SIGNATURE,
                                      &v_STRING))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    *expected_validity =
    DBUS_INVALID_EXCEEDED_MAXIMUM_ARRAY_RECURSION;
}
else if (item_seq == 2)
{
    char long_sig[DBUS_MAXIMUM_TYPE_RECURSION_DEPTH*2+4];
    const char *v_STRING;
    int i;

    message = simple_method_call ();
    if (!dbus_message_append_args (message,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    i = 0;
    while (i <= (DBUS_MAXIMUM_TYPE_RECURSION_DEPTH + 1))
    {
        long_sig[i] = DBUS_STRUCT_BEGIN_CHAR;
        ++i;
    }

    long_sig[i] = DBUS_TYPE_INT32;
    ++i;

    while (i < (DBUS_MAXIMUM_TYPE_RECURSION_DEPTH*2 + 3))
    {
        long_sig[i] = DBUS_STRUCT_END_CHAR;

```

```

        ++i;
    }
    long_sig[i] = DBUS_TYPE_INVALID;

    v_STRING = long_sig;
    if (!_dbus_header_set_field_basic (&message->header,
                                      DBUS_HEADER_FIELD_SIGNATURE,
                                      DBUS_TYPE_SIGNATURE,
                                      &v_STRING))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    *expected_validity =
DBUS_INVALID_EXCEEDED_MAXIMUM_STRUCT_RECURSION;
}
else if (item_seq == 3)
{
    message = simple_method_call ();
    if (!_dbus_message_append_args (message,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, pos + 1, DBUS_STRUCT_BEGIN_CHAR);

    *expected_validity = DBUS_INVALID_STRUCT_STARTED_BUT_NOT_ENDED;
}
else if (item_seq == 4)
{
    message = simple_method_call ();
    if (!_dbus_message_append_args (message,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

```

```

    _dbus_string_set_byte (data, pos + 1, DBUS_STRUCT_END_CHAR);

    *expected_validity = DBUS_INVALID_STRUCT_ENDED_BUT_NOT_STARTED;
}
else if (item_seq == 5)
{
    message = simple_method_call ();
    if (!dbus_message_append_args (message,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INT32, &v_INT32,
                                   DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, pos + 1, DBUS_STRUCT_BEGIN_CHAR);
    _dbus_string_set_byte (data, pos + 2, DBUS_STRUCT_END_CHAR);

    *expected_validity = DBUS_INVALID_STRUCT_HAS_NO_FIELDS;
}
else if (item_seq == 6)
{
    message = simple_method_call ();
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, TYPE_OFFSET,
DBUS_MESSAGE_TYPE_INVALID);

    *expected_validity = DBUS_INVALID_BAD_MESSAGE_TYPE;
}
else if (item_seq == 7)
{
    /* Messages of unknown type are considered valid */
    message = simple_method_call ();
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, TYPE_OFFSET, 100);

    *expected_validity = DBUS_VALID;
}
else if (item_seq == 8)
{
    char byte_order;

    message = simple_method_call ();
    byte_order = _dbus_header_get_byte_order (&message->header);
    generate_from_message (data, expected_validity, message);
}

```



```

    _dbus_marshal_set_uint32 (data, BODY_LENGTH_OFFSET,
                              DBUS_MAXIMUM_MESSAGE_LENGTH / 2 + 4,
                              byte_order);
    _dbus_marshal_set_uint32 (data, FIELDS_ARRAY_LENGTH_OFFSET,
                              DBUS_MAXIMUM_MESSAGE_LENGTH / 2 + 4,
                              byte_order);
    *expected_validity = DBUS_INVALID_MESSAGE_TOO_LONG;
}
else if (item_seq == 9)
{
    const char *v_STRING = "not a valid bus name";
    message = simple_method_call ();

    if (!_dbus_header_set_field_basic (&message->header,
                                       DBUS_HEADER_FIELD_SENDER,
                                       DBUS_TYPE_STRING, &v_STRING))
        _dbus_assert_not_reached ("oom");

    generate_from_message (data, expected_validity, message);

    *expected_validity = DBUS_INVALID_BAD_SENDER;
}
else if (item_seq == 10)
{
    message = simple_method_call ();

    if (!dbus_message_set_interface (message, DBUS_INTERFACE_LOCAL))
        _dbus_assert_not_reached ("oom");

    generate_from_message (data, expected_validity, message);

    *expected_validity = DBUS_INVALID_USES_LOCAL_INTERFACE;
}
else if (item_seq == 11)
{
    message = simple_method_call ();

    if (!dbus_message_set_path (message, DBUS_PATH_LOCAL))
        _dbus_assert_not_reached ("oom");

    generate_from_message (data, expected_validity, message);

    *expected_validity = DBUS_INVALID_USES_LOCAL_PATH;
}
else if (item_seq == 12)
{
    /* Method calls don't have to have interface */
    message = simple_method_call ();

    if (!dbus_message_set_interface (message, NULL))
        _dbus_assert_not_reached ("oom");
}

```



```

                                DBUS_TYPE_INVALID))
    _dbus_assert_not_reached ("oom");

i = 0;
while (i <= (DBUS_MAXIMUM_TYPE_RECURSION_DEPTH*3 + 3))
{
    long_sig[i] = DBUS_TYPE_ARRAY;
    ++i;
    long_sig[i] = DBUS_DICT_ENTRY_BEGIN_CHAR;
    ++i;
    long_sig[i] = DBUS_TYPE_INT32;
    ++i;
}
n_begins = i / 3;

long_sig[i] = DBUS_TYPE_INT32;
++i;

while (n_begins > 0)
{
    long_sig[i] = DBUS_DICT_ENTRY_END_CHAR;
    ++i;
    n_begins -= 1;
}
long_sig[i] = DBUS_TYPE_INVALID;

v_STRING = long_sig;
if (!_dbus_header_set_field_basic (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                DBUS_TYPE_SIGNATURE,
                                &v_STRING))
    _dbus_assert_not_reached ("oom");

_dbus_header_get_field_raw (&message->header,
                            DBUS_HEADER_FIELD_SIGNATURE,
                            NULL, &pos);
generate_from_message (data, expected_validity, message);

*expected_validity =
DBUS_INVALID_EXCEEDED_MAXIMUM_DICT_ENTRY_RECURSION;
}
else if (item_seq == 17)
{
    message = simple_method_call ();
    if (!_dbus_message_append_args (message,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,

```

```

                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, pos + 1, DBUS_TYPE_ARRAY);
    _dbus_string_set_byte (data, pos + 2,
DBUS_DICT_ENTRY_BEGIN_CHAR);

    *expected_validity =
DBUS_INVALID_DICT_ENTRY_STARTED_BUT_NOT_ENDED;
}
else if (item_seq == 18)
{
    message = simple_method_call ();
    if (!dbus_message_append_args (message,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, pos + 1, DBUS_DICT_ENTRY_END_CHAR);

    *expected_validity =
DBUS_INVALID_DICT_ENTRY_ENDED_BUT_NOT_STARTED;
}
else if (item_seq == 19)
{
    message = simple_method_call ();
    if (!dbus_message_append_args (message,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INT32, &v_INT32,
                                    DBUS_TYPE_INVALID))
        _dbus_assert_not_reached ("oom");

    _dbus_header_get_field_raw (&message->header,
                                DBUS_HEADER_FIELD_SIGNATURE,
                                NULL, &pos);
    generate_from_message (data, expected_validity, message);

    _dbus_string_set_byte (data, pos + 1, DBUS_TYPE_ARRAY);
    _dbus_string_set_byte (data, pos + 2,
DBUS_DICT_ENTRY_BEGIN_CHAR);
    _dbus_string_set_byte (data, pos + 3, DBUS_DICT_ENTRY_END_CHAR);

    *expected_validity = DBUS_INVALID_DICT_ENTRY_HAS_NO_FIELDS;

```

```

    }
else if (item_seq == 20)
    {
        /* 64 levels of nesting is OK */
        message = message_with_nesting_levels(64);

        generate_from_message (data, expected_validity, message);

        *expected_validity = DBUS_VALID;
    }
else if (item_seq == 21)
    {
        /* 65 levels of nesting is not OK */
        message = message_with_nesting_levels(65);

        generate_from_message (data, expected_validity, message);

        *expected_validity = DBUS_INVALID_NESTED_TOO_DEEPLY;
    }
else
    {
        return FALSE;
    }

if (message)
    dbus_message_unref (message);

iter_next (iter);
return TRUE;
}

static dbus_bool_t
generate_wrong_length (DBusMessageDataIter *iter,
                      DBusString          *data,
                      DBusValidity        *expected_validity)
{
    int lengths[] = { -42, -17, -16, -15, -9, -8, -7, -6, -5, -4, -3, -
2, -1,
                    1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 30 };

    int adjust;
    int len_seq;

restart:
    len_seq = iter_get_sequence (iter);
    if (len_seq == _DBUS_N_ELEMENTS (lengths))
        return FALSE;

    _dbus_assert (len_seq < _DBUS_N_ELEMENTS (lengths));

    iter_recurse (iter);
    if (!generate_many_bodies (iter, data, expected_validity))
        {

```

```

        iter_set_sequence (iter, 0); /* reset to first body */
        iter_unrecurse (iter);
        iter_next (iter);          /* next length adjustment */
        goto restart;
    }
    iter_unrecurse (iter);

    adjust = lengths[len_seq];

    if (adjust < 0)
    {
        if ((_dbus_string_get_length (data) + adjust) <
            DBUS_MINIMUM_HEADER_SIZE)
            _dbus_string_set_length (data, DBUS_MINIMUM_HEADER_SIZE);
        else
            _dbus_string_shorten (data, - adjust);
        *expected_validity = DBUS_INVALID_FOR_UNKNOWN_REASON;
    }
    else
    {
        if (!_dbus_string_lengthen (data, adjust))
            _dbus_assert_not_reached ("oom");
        *expected_validity = DBUS_INVALID_TOO_MUCH_DATA;
    }

    /* Fixup lengths */
    {
        int old_body_len;
        int new_body_len;
        int byte_order;

        _dbus_assert (_dbus_string_get_length (data) >=
            DBUS_MINIMUM_HEADER_SIZE);

        byte_order = _dbus_string_get_byte (data, BYTE_ORDER_OFFSET);
        old_body_len = _dbus_marshal_read_uint32 (data,
                                                    BODY_LENGTH_OFFSET,
                                                    byte_order,
                                                    NULL);

        _dbus_assert (old_body_len < _dbus_string_get_length (data));
        new_body_len = old_body_len + adjust;
        if (new_body_len < 0)
        {
            new_body_len = 0;
            /* we just munged the header, and aren't sure how */
            *expected_validity = DBUS_VALIDITY_UNKNOWN;
        }

        _dbus_verbose ("changing body len from %u to %u by adjust %d\n",
            old_body_len, new_body_len, adjust);

        _dbus_marshal_set_uint32 (data, BODY_LENGTH_OFFSET,

```

```

        new_body_len,
        byte_order);
    }

    return TRUE;
}

static dbus_bool_t
generate_byte_changed (DBusMessageDataIter *iter,
                      DBusString          *data,
                      DBusValidity        *expected_validity)
{
    int byte_seq;
    int v_BYTE;

    /* This is a little convoluted to make the bodies the
     * outer loop and each byte of each body the inner
     * loop
     */

restart:
    if (!generate_many_bodies (iter, data, expected_validity))
        return FALSE;

    iter_recurse (iter);
    byte_seq = iter_get_sequence (iter);
    iter_next (iter);
    iter_unrecurse (iter);

    if (byte_seq == _dbus_string_get_length (data))
    {
        _dbus_string_set_length (data, 0);
        /* reset byte count */
        iter_recurse (iter);
        iter_set_sequence (iter, 0);
        iter_unrecurse (iter);
        goto restart;
    }
    else
    {
        /* Undo the "next" in generate_many_bodies */
        iter_set_sequence (iter, iter_get_sequence (iter) - 1);
    }

    _dbus_assert (byte_seq < _dbus_string_get_length (data));
    v_BYTE = _dbus_string_get_byte (data, byte_seq);
    v_BYTE += byte_seq; /* arbitrary but deterministic change to the
    byte */
    _dbus_string_set_byte (data, byte_seq, v_BYTE);
    *expected_validity = DBUS_VALIDITY_UNKNOWN;

    return TRUE;
}

```

```

}

#if 0
/* This is really expensive and doesn't add too much coverage */

static dbus_bool_t
find_next_typecode (DBusMessageDataIter *iter,
                   DBusString          *data,
                   DBusValidity        *expected_validity)
{
    int body_seq;
    int byte_seq;
    int base_depth;

    base_depth = iter->depth;

restart:
    _dbus_assert (iter->depth == (base_depth + 0));
    _dbus_string_set_length (data, 0);

    body_seq = iter_get_sequence (iter);

    if (!generate_many_bodies (iter, data, expected_validity))
        return FALSE;
    /* Undo the "next" in generate_many_bodies */
    iter_set_sequence (iter, body_seq);

    iter_recurse (iter);
    while (TRUE)
    {
        _dbus_assert (iter->depth == (base_depth + 1));

        byte_seq = iter_get_sequence (iter);

        _dbus_assert (byte_seq <= _dbus_string_get_length (data));

        if (byte_seq == _dbus_string_get_length (data))
        {
            /* reset byte count */
            iter_set_sequence (iter, 0);
            iter_unrecurse (iter);
            _dbus_assert (iter->depth == (base_depth + 0));
            iter_next (iter); /* go to the next body */
            goto restart;
        }

        _dbus_assert (byte_seq < _dbus_string_get_length (data));

        if (dbus_type_is_valid (_dbus_string_get_byte (data, byte_seq)))
            break;
        else
            iter_next (iter);
    }
}

```



```

    }

    _dbus_assert (byte_seq == iter_get_sequence (iter));
    _dbus_assert (byte_seq < _dbus_string_get_length (data));

    iter_unrecurse (iter);

    _dbus_assert (iter->depth == (base_depth + 0));

    return TRUE;
}

static const int typecodes[] = {
    DBUS_TYPE_INVALID,
    DBUS_TYPE_BYTE,
    DBUS_TYPE_BOOLEAN,
    DBUS_TYPE_INT16,
    DBUS_TYPE_UINT16,
    DBUS_TYPE_INT32,
    DBUS_TYPE_UINT32,
    DBUS_TYPE_INT64,
    DBUS_TYPE_UINT64,
    DBUS_TYPE_DOUBLE,
    DBUS_TYPE_STRING,
    DBUS_TYPE_OBJECT_PATH,
    DBUS_TYPE_SIGNATURE,
    DBUS_TYPE_ARRAY,
    DBUS_TYPE_VARIANT,
    DBUS_STRUCT_BEGIN_CHAR,
    DBUS_STRUCT_END_CHAR,
    DBUS_DICT_ENTRY_BEGIN_CHAR,
    DBUS_DICT_ENTRY_END_CHAR,
    DBUS_TYPE_UNIX_FD,
    255 /* random invalid typecode */
};

static dbus_bool_t
generate_typecode_changed (DBusMessageDataIter *iter,
                           DBusString          *data,
                           DBusValidity        *expected_validity)
{
    int byte_seq;
    int typecode_seq;
    int base_depth;

    base_depth = iter->depth;

restart:
    _dbus_assert (iter->depth == (base_depth + 0));
    _dbus_string_set_length (data, 0);

    if (!find_next_typecode (iter, data, expected_validity))

```

```

    return FALSE;

    iter_recurse (iter);
    byte_seq = iter_get_sequence (iter);

    _dbus_assert (byte_seq < _dbus_string_get_length (data));

    iter_recurse (iter);
    typecode_seq = iter_get_sequence (iter);
    iter_next (iter);

    _dbus_assert (typecode_seq <= _DBUS_N_ELEMENTS (typecodes));

    if (typecode_seq == _DBUS_N_ELEMENTS (typecodes))
    {
        _dbus_assert (iter->depth == (base_depth + 2));
        iter_set_sequence (iter, 0); /* reset typecode sequence */
        iter_unrecurse (iter);
        _dbus_assert (iter->depth == (base_depth + 1));
        iter_next (iter); /* go to the next byte_seq */
        iter_unrecurse (iter);
        _dbus_assert (iter->depth == (base_depth + 0));
        goto restart;
    }

    _dbus_assert (iter->depth == (base_depth + 2));
    iter_unrecurse (iter);
    _dbus_assert (iter->depth == (base_depth + 1));
    iter_unrecurse (iter);
    _dbus_assert (iter->depth == (base_depth + 0));

#if 0
    printf ("Changing byte %d in message %d to %c\n",
           byte_seq, iter_get_sequence (iter),
           typecodes[typecode_seq]);
#endif

    _dbus_string_set_byte (data, byte_seq, typecodes[typecode_seq]);
    *expected_validity = DBUS_VALIDITY_UNKNOWN;
    return TRUE;
}
#endif

typedef struct
{
    ChangeType type;
    dbus_uint32_t value; /* cast to signed for adjusts */
} UIntChange;

static const UIntChange uint32_changes[] = {
    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) -1 },
    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) -2 },

```

```

    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) -3 },
    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) 1 },
    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) 2 },
    { CHANGE_TYPE_ADJUST, (dbus_uint32_t) 3 },
    { CHANGE_TYPE_ABSOLUTE, _DBUS_UINT32_MAX },
    { CHANGE_TYPE_ABSOLUTE, 0 },
    { CHANGE_TYPE_ABSOLUTE, 1 },
    { CHANGE_TYPE_ABSOLUTE, _DBUS_UINT32_MAX - 1 },
    { CHANGE_TYPE_ABSOLUTE, _DBUS_UINT32_MAX - 5 }
};

static dbus_bool_t
generate_uint32_changed (DBusMessageDataIter *iter,
                        DBusString          *data,
                        DBusValidity        *expected_validity)
{
    int body_seq;
    int byte_seq;
    int change_seq;
    dbus_uint32_t v_UINT32;
    int byte_order;
    const UIntChange *change;
    int base_depth;

    /* Outer loop is each body, next loop is each change,
     * inner loop is each change location
     */

    base_depth = iter->depth;

next_body:
    _dbus_assert (iter->depth == (base_depth + 0));
    _dbus_string_set_length (data, 0);
    body_seq = iter_get_sequence (iter);

    if (!generate_many_bodies (iter, data, expected_validity))
        return FALSE;

    _dbus_assert (iter->depth == (base_depth + 0));

    iter_set_sequence (iter, body_seq); /* undo the "next" from
generate_many_bodies */
    iter_recurse (iter);
next_change:
    _dbus_assert (iter->depth == (base_depth + 1));
    change_seq = iter_get_sequence (iter);

    if (change_seq == _DBUS_N_ELEMENTS (uint32_changes))
    {
        /* Reset change count */
        iter_set_sequence (iter, 0);
        iter_unrecurse (iter);
    }
}

```

```

        iter_next (iter);
        goto next_body;
    }

    _dbus_assert (iter->depth == (base_depth + 1));

    iter_recurse (iter);
    _dbus_assert (iter->depth == (base_depth + 2));
    byte_seq = iter_get_sequence (iter);
    /* skip 4 bytes at a time */
    iter_next (iter);
    iter_next (iter);
    iter_next (iter);
    iter_next (iter);
    iter_unrecurse (iter);

    _dbus_assert (_DBUS_ALIGN_VALUE (byte_seq, 4) == (unsigned)
byte_seq);
    if (byte_seq >= (_dbus_string_get_length (data) - 4))
    {
        /* reset byte count */
        _dbus_assert (iter->depth == (base_depth + 1));
        iter_recurse (iter);
        _dbus_assert (iter->depth == (base_depth + 2));
        iter_set_sequence (iter, 0);
        iter_unrecurse (iter);
        _dbus_assert (iter->depth == (base_depth + 1));
        iter_next (iter);
        goto next_change;
    }

    _dbus_assert (byte_seq <= (_dbus_string_get_length (data) - 4));

    byte_order = _dbus_string_get_byte (data, BYTE_ORDER_OFFSET);

    v_UINT32 = _dbus_marshal_read_uint32 (data, byte_seq, byte_order,
NULL);

    change = &uint32_changes[change_seq];

    if (change->type == CHANGE_TYPE_ADJUST)
    {
        v_UINT32 += (int) change->value;
    }
    else
    {
        v_UINT32 = change->value;
    }

#ifdef 0
    printf ("body %d change %d pos %d ",
            body_seq, change_seq, byte_seq);
#endif

```

```

    if (change->type == CHANGE_TYPE_ADJUST)
        printf ("adjust by %d", (int) change->value);
    else
        printf ("set to %u", change->value);

    printf (" \t%u -> %u\n",
        _dbus_marshal_read_uint32 (data, byte_seq, byte_order,
NULL),
        v_UINT32);
#endif

    _dbus_marshal_set_uint32 (data, byte_seq, v_UINT32, byte_order);
    *expected_validity = DBUS_VALIDITY_UNKNOWN;

    _dbus_assert (iter->depth == (base_depth + 1));
    iter_unrecurse (iter);
    _dbus_assert (iter->depth == (base_depth + 0));

    return TRUE;
}

typedef struct
{
    const char *name;
    DBusMessageGeneratorFunc func;
} DBusMessageGenerator;

static const DBusMessageGenerator generators[] = {
    { "trivial example of each message type", generate_trivial },
    { "assorted arguments", generate_many_bodies },
    { "assorted special cases", generate_special },
    { "each uint32 modified", generate_uint32_changed },
    { "wrong body lengths", generate_wrong_length },
    { "each byte modified", generate_byte_changed },
#ifdef 0
    /* This is really expensive and doesn't add too much coverage */
    { "change each typecode", generate_typecode_changed }
#endif
};

void
_dbus_message_data_free (DBusMessageData *data)
{
    _dbus_string_free (&data->data);
}

void
_dbus_message_data_iter_init (DBusMessageDataIter *iter)
{
    int i;

```

```

iter->depth = 0;
i = 0;
while (i < _DBUS_MESSAGE_DATA_MAX_NESTING)
    {
        iter->sequence_nos[i] = 0;
        ++i;
    }
iter->count = 0;
}

dbus_bool_t
_dbus_message_data_iter_get_and_next (DBusMessageDataIter *iter,
                                       DBusMessageData *data)
{
    DBusMessageGeneratorFunc func;
    int generator;

restart:
    generator = iter_get_sequence (iter);

    if (generator == _DBUS_N_ELEMENTS (generators))
        return FALSE;

    iter_recurse (iter);

    if (iter_first_in_series (iter))
        {
            printf (" testing message loading: %s ",
generators[generator].name);
            fflush (stdout);
        }

    func = generators[generator].func;

    if (!_dbus_string_init (&data->data))
        _dbus_assert_not_reached ("oom");

    if ((*func)(iter, &data->data, &data->expected_validity))
        ;
    else
        {
            iter_set_sequence (iter, 0);
            iter_unrecurse (iter);
            iter_next (iter); /* next generator */
            _dbus_string_free (&data->data);
            printf ("%d test loads cumulative\n", iter->count);
            goto restart;
        }
    iter_unrecurse (iter);

    iter->count += 1;
    return TRUE;
}

```

```

}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif /* DBUS_BUILD_TESTS */

File = dbus-message-factory.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message-factory.h Generator of valid and invalid message data
for test suite
*
* Copyright (C) 2005 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef DBUS_MESSAGE_FACTORY_H
#define DBUS_MESSAGE_FACTORY_H

#ifdef DBUS_BUILD_TESTS

#include <dbus/dbus-string.h>
#include <dbus/dbus-marshal-basic.h>
#include <dbus/dbus-marshal-validate.h>

DBUS_BEGIN_DECLS

typedef struct
{
    DBusValidity expected_validity;

```

```

    DBusString data;

} DBusMessageData;

#define _DBUS_MESSAGE_DATA_MAX_NESTING 10
typedef struct
{
    int sequence_nos[_DBUS_MESSAGE_DATA_MAX_NESTING];
    int depth;
    int count;
} DBusMessageDataIter;

void      _dbus_message_data_free      (DBusMessageData
*data);
void      _dbus_message_data_iter_init (DBusMessageDataIter
*iter);
dbus_bool_t _dbus_message_data_iter_get_and_next (DBusMessageDataIter
*iter,
                                                DBusMessageData
*data);

DBUS_END_DECLS

#endif /* DBUS_BUILD_TESTS */

#endif /* DBUS_MESSAGE_FACTORY_H */

```

File = dbus-message-internal.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message-internal.h DBusMessage object internal interfaces
 *
 * Copyright (C) 2002 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License

```



```

* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_MESSAGE_INTERNAL_H
#define DBUS_MESSAGE_INTERNAL_H

#include <dbus/dbus-marshal-validate.h>
#include <dbus/dbus-message.h>
#include <dbus/dbus-resources.h>
#include <dbus/dbus-list.h>

DBUS_BEGIN_DECLS

#ifdef DBUS_ENABLE_VERBOSE_MODE
void _dbus_message_trace_ref (DBusMessage *message,
                              int          old_refcount,
                              int          new_refcount,
                              const char  *why);
#else
/* this bypasses any "unused" warnings for the old and new refcount */
#define _dbus_message_trace_ref(m, o, n, w) \
do \
{ \
    (void) (o); \
    (void) (n); \
} while (0)
#endif

typedef struct DBusMessageLoader DBusMessageLoader;

void _dbus_message_get_network_data (DBusMessage *message,
                                     const DBusString **header,
                                     const DBusString **body);
void _dbus_message_get_unix_fds (DBusMessage *message,
                                 const int **fds,
                                 unsigned *n_fds);

void _dbus_message_lock (DBusMessage
                        *message);
void _dbus_message_unlock (DBusMessage
                          *message);
dbus_bool_t _dbus_message_add_counter (DBusMessage
                                       *message,
                                       DBusCounter
                                       *counter);
void _dbus_message_add_counter_link (DBusMessage
                                     *message,
                                     DBusList
                                     *link);
void _dbus_message_remove_counter (DBusMessage
                                   *message,

```

```

DBusCounter

*counter);

DBusMessageLoader* _dbus_message_loader_new          (void);
DBusMessageLoader* _dbus_message_loader_ref
(DBusMessageLoader *loader);
void _dbus_message_loader_unref
(DBusMessageLoader *loader);

void _dbus_message_loader_get_buffer
(DBusMessageLoader *loader,

DBusString **buffer);
void _dbus_message_loader_return_buffer
(DBusMessageLoader *loader,

DBusString *buffer,                                int
bytes_read);

dbus_bool_t _dbus_message_loader_get_unix_fds
(DBusMessageLoader *loader,                                int

**fds,

unsigned *max_n_fds);
void _dbus_message_loader_return_unix_fds
(DBusMessageLoader *loader,                                int

*fds,

unsigned n_fds);

dbus_bool_t _dbus_message_loader_queue_messages
(DBusMessageLoader *loader);
DBusMessage* _dbus_message_loader_peek_message
(DBusMessageLoader *loader);
DBusMessage* _dbus_message_loader_pop_message
(DBusMessageLoader *loader);
DBusList* _dbus_message_loader_pop_message_link
(DBusMessageLoader *loader);
void _dbus_message_loader_putback_message_link
(DBusMessageLoader *loader,

DBusList *link);

dbus_bool_t _dbus_message_loader_get_is_corrupted
(DBusMessageLoader *loader);
DBusValidity _dbus_message_loader_get_corruption_reason
(DBusMessageLoader *loader);

```

```

void      _dbus_message_loader_set_max_message_size
(DBusMessageLoader *loader,
                                                long
size);
long      _dbus_message_loader_get_max_message_size
(DBusMessageLoader *loader);

void
_dbus_message_loader_set_max_message_unix_fds(DBusMessageLoader
*loader,
                                                long
n);
long
_dbus_message_loader_get_max_message_unix_fds(DBusMessageLoader
*loader);

typedef struct DBusInitialFDs DBusInitialFDs;
DBusInitialFDs *_dbus_check_fdleaks_enter (void);
void      _dbus_check_fdleaks_leave (DBusInitialFDs *fds);

DBUS_END_DECLS

#endif /* DBUS_MESSAGE_INTERNAL_H */

```

File = dbus-message-private.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message-private.h header shared between dbus-message.c and
dbus-message-util.c
*
* Copyright (C) 2005 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

```

```

*
*/
#ifndef DBUS_MESSAGE_PRIVATE_H
#define DBUS_MESSAGE_PRIVATE_H

#include <dbus/dbus-message.h>
#include <dbus/dbus-message-internal.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-dataslot.h>
#include <dbus/dbus-marshal-header.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusMessageInternals
 * @{
 */

/**
 * @typedef DBusMessageLoader
 *
 * The DBusMessageLoader object encapsulates the process of converting
 * a byte stream into a series of DBusMessage. It buffers the incoming
 * bytes as efficiently as possible, and generates a queue of
 * messages. DBusMessageLoader is typically used as part of a
 * DBusTransport implementation. The DBusTransport then hands off
 * the loaded messages to a DBusConnection, making the messages
 * visible to the application.
 *
 * @todo write tests for break-loader that a) randomly delete header
 * fields and b) set string fields to zero-length and other funky
 * values.
 */

/**
 * Implementation details of DBusMessageLoader.
 * All members are private.
 */
struct DBusMessageLoader
{
    int refcount;          /**< Reference count. */

    DBusString data;      /**< Buffered data */

    DBusList *messages;  /**< Complete messages. */

    long max_message_size; /**< Maximum size of a message */
    long max_message_unix_fds; /**< Maximum unix fds in a message */

    DBusValidity corruption_reason; /**< why we were corrupted */

```

```

    unsigned int corrupted : 1; /**< We got broken data, and are no
longer working */

    unsigned int buffer_outstanding : 1; /**< Someone is using the
buffer to read */

#ifdef HAVE_UNIX_FD_PASSING
    unsigned int unix_fds_outstanding : 1; /**< Someone is using the
unix fd array to read */

    int *unix_fds; /**< File descriptors that have been read from the
transport but not yet been handed to any message. Array will be
allocated at first use. */
    unsigned n_unix_fds_allocated; /**< Number of file descriptors this
array has space for */
    unsigned n_unix_fds; /**< Number of valid file descriptors in array
*/
#endif
};

/** How many bits are in the changed_stamp used to validate iterators
*/
#define CHANGED_STAMP_BITS 21

/**
 * @brief Internals of DBusMessage
 *
 * Object representing a message received from or to be sent to
 * another application. This is an opaque object, all members
 * are private.
 */
struct DBusMessage
{
    DBusAtomic refcount; /**< Reference count */

    DBusHeader header; /**< Header network data and associated cache */

    DBusString body; /**< Body network data. */

    unsigned int locked : 1; /**< Message being sent, no modifications
allowed. */

#ifdef DBUS_DISABLE_CHECKS
    unsigned int in_cache : 1; /**< Has been "freed" since it's in the
cache (this is a debug feature) */
#endif

    DBusList *counters; /**< 0-N DBusCounter used to track message
size/unix fds. */
    long size_counter_delta; /**< Size we incremented the size
counters by. */

```

```

    dbus_uint32_t changed_stamp : CHANGED_STAMP_BITS; /**< Incremented
when iterators are invalidated. */

    DBusDataSlotList slot_list; /**< Data stored by allocated integer
ID */

#ifdef DBUS_DISABLE_CHECKS
    int generation; /**< _dbus_current_generation when message was
created */
#endif

#ifdef HAVE_UNIX_FD_PASSING
    int *unix_fds;
    /**< Unix file descriptors associated with this message. These are
        closed when the message is destroyed, hence make sure to dup()
        them when adding or removing them here. */
    unsigned n_unix_fds; /**< Number of valid fds in the array */
    unsigned n_unix_fds_allocated; /**< Allocated size of the array */

    long unix_fd_counter_delta; /**< Size we incremented the unix fd
counter by */
#endif
};

dbus_bool_t _dbus_message_iter_get_args_valist (DBusMessageIter *iter,
                                                DBusError
                                                *error,
                                                int
                                                first_arg_type,
                                                va_list
                                                var_args);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_MESSAGE_H */

```

File = dbus-message-util.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message-util.c Would be in dbus-message.c, but only used by
bus/tests
*
* Copyright (C) 2002, 2003, 2004, 2005 Red Hat Inc.
* Copyright (C) 2002, 2003 CodeFactory AB
*
* Licensed under the Academic Free License version 2.1
*

```

```
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-internals.h"
#include "dbus-test.h"
#include "dbus-message-private.h"
#include "dbus-marshall-recursive.h"
#include "dbus-string.h"
#ifdef HAVE_UNIX_FD_PASSING
#include "dbus-sysdeps-unix.h"
#endif
```

```
#ifdef __linux__
/* Necessary for the Linux-specific fd leak checking code only */
#include <sys/types.h>
#include <dirent.h>
#include <stdlib.h>
#include <errno.h>
#endif
```

```
/**
 * @addtogroup DBusMessage
 * @{
 */
```

```
#ifdef DBUS_BUILD_TESTS
/**
 * Reads arguments from a message iterator given a variable argument
 * list. Only arguments of basic type and arrays of fixed-length
 * basic type may be read with this function. See
 * dbus_message_get_args() for more details.
 *
 * @param iter the message iterator
 * @param error error to be filled in on failure
 * @param first_arg_type the first argument type
```

```

* @param ... location for first argument value, then list of type-
location pairs
* @returns #FALSE if the error was set
*/
static dbus_bool_t
dbus_message_iter_get_args (DBusMessageIter *iter,
                           DBusError      *error,
                           int            first_arg_type,
                           ...)
{
    dbus_bool_t retval;
    va_list var_args;

    _dbus_return_val_if_fail (iter != NULL, FALSE);
    _dbus_return_val_if_error_is_set (error, FALSE);

    va_start (var_args, first_arg_type);
    retval = _dbus_message_iter_get_args_valist (iter, error,
first_arg_type, var_args);
    va_end (var_args);

    return retval;
}
#endif /* DBUS_BUILD_TESTS */

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include "dbus-message-factory.h"
#include <stdio.h>
#include <stdlib.h>

static int validities_seen[DBUS_VALIDITY_LAST +
_DBUS_NEGATIVE_VALIDITY_COUNT];

static void
reset_validities_seen (void)
{
    int i;
    i = 0;
    while (i < _DBUS_N_ELEMENTS (validities_seen))
    {
        validities_seen[i] = 0;
        ++i;
    }
}

static void
record_validity_seen (DBusValidity validity)
{
    validities_seen[validity + _DBUS_NEGATIVE_VALIDITY_COUNT] += 1;
}

```



```

}

static void
print_validities_seen (dbus_bool_t not_seen)
{
    int i;
    i = 0;
    while (i < _DBUS_N_ELEMENTS (validities_seen))
    {
        if ((i - _DBUS_NEGATIVE_VALIDITY_COUNT) == DBUS_VALIDITY_UNKNOWN
||
        (i - _DBUS_NEGATIVE_VALIDITY_COUNT) ==
DBUS_INVALID_FOR_UNKNOWN_REASON)
            ;
        else if ((not_seen && validities_seen[i] == 0) ||
        (!not_seen && validities_seen[i] > 0))
            printf ("validity %3d seen %d times\n",
                i - _DBUS_NEGATIVE_VALIDITY_COUNT,
                validities_seen[i]);
        ++i;
    }
}

static void
check_memleaks (void)
{
    dbus_shutdown ();

    if (_dbus_get_malloc_blocks_outstanding () != 0)
    {
        _dbus_warn ("%d dbus_malloc blocks were not freed in %s\n",
            _dbus_get_malloc_blocks_outstanding (), __FILE__);
        _dbus_assert_not_reached ("memleaks");
    }
}

#ifdef __linux__
struct DBusInitialFDs {
    fd_set set;
};
#endif

DBusInitialFDs *
_dbus_check_fdleaks_enter (void)
{
#ifdef __linux__
    DIR *d;
    DBusInitialFDs *fds;

    /* this is plain malloc so it won't interfere with leak checking */
    fds = malloc (sizeof (DBusInitialFDs));
    _dbus_assert (fds != NULL);

```

```

/* This works on Linux only */

if ((d = opendir ("/proc/self/fd"))
    {
    struct dirent *de;

    while ((de = readdir(d))
        {
        long l;
        char *e = NULL;
        int fd;

        if (de->d_name[0] == '.')
            continue;

        errno = 0;
        l = strtol (de->d_name, &e, 10);
        _dbus_assert (errno == 0 && e && !*e);

        fd = (int) l;

        if (fd < 3)
            continue;

        if (fd == dirfd (d))
            continue;

        FD_SET (fd, &fds->set);
        }

    closedir (d);
    }

    return fds;
#else
    return NULL;
#endif
}

void
_dbus_check_fdleaks_leave (DBusInitialFDs *fds)
{
#ifdef __linux__
    DIR *d;

    /* This works on Linux only */

    if ((d = opendir ("/proc/self/fd"))
        {
        struct dirent *de;

```

```

while ((de = readdir(d))
{
    long l;
    char *e = NULL;
    int fd;

    if (de->d_name[0] == '.')
        continue;

    errno = 0;
    l = strtol (de->d_name, &e, 10);
    _dbus_assert (errno == 0 && e && !*e);

    fd = (int) l;

    if (fd < 3)
        continue;

    if (fd == dirfd (d))
        continue;

    if (FD_ISSET (fd, &fds->set))
        continue;

    _dbus_warn ("file descriptor %i leaked in %s.\n", fd,
__FILE__);
    _dbus_assert_not_reached ("fdleaks");
}

    closedir (d);
}

free (fds);
#else
_dbus_assert (fds == NULL);
#endif
}

static dbus_bool_t
check_have_valid_message (DBusMessageLoader *loader)
{
    DBusMessage *message;
    dbus_bool_t retval;

    message = NULL;
    retval = FALSE;

    if (_dbus_message_loader_get_is_corrupted (loader))
    {
        _dbus_warn ("loader corrupted on message that was expected to be
valid; invalid reason %d\n",
                    loader->corruption_reason);
    }
}

```

```

        goto failed;
    }

    message = _dbus_message_loader_pop_message (loader);
    if (message == NULL)
    {
        _dbus_warn ("didn't load message that was expected to be valid
(message not popped)\n");
        goto failed;
    }

    if (_dbus_string_get_length (&loader->data) > 0)
    {
        _dbus_warn ("had leftover bytes from expected-to-be-valid single
message\n");
        goto failed;
    }

#ifdef 0
    /* FIXME */
    /* Verify that we're able to properly deal with the message.
    * For example, this would detect improper handling of messages
    * in nonstandard byte order.
    */
    if (!check_message_handling (message))
        goto failed;
#endif

    record_validity_seen (DBUS_VALID);

    retval = TRUE;

failed:
    if (message)
        dbus_message_unref (message);

    return retval;
}

static dbus_bool_t
check_invalid_message (DBusMessageLoader *loader,
                      DBusValidity      expected_validity)
{
    dbus_bool_t retval;

    retval = FALSE;

    if (!_dbus_message_loader_get_is_corrupted (loader))
    {
        _dbus_warn ("loader not corrupted on message that was expected
to be invalid\n");
        goto failed;
    }

```

```

    }

    record_validity_seen (loader->corruption_reason);

    if (expected_validity != DBUS_INVALID_FOR_UNKNOWN_REASON &&
        loader->corruption_reason != expected_validity)
    {
        _dbus_warn ("expected message to be corrupted for reason %d and
was corrupted for %d instead\n",
                    expected_validity, loader->corruption_reason);
        goto failed;
    }

    retval = TRUE;

failed:
    return retval;
}

static dbus_bool_t
check_incomplete_message (DBusMessageLoader *loader)
{
    DBusMessage *message;
    dbus_bool_t retval;

    message = NULL;
    retval = FALSE;

    if (_dbus_message_loader_get_is_corrupted (loader))
    {
        _dbus_warn ("loader corrupted on message that was expected to be
valid (but incomplete), corruption reason %d\n",
                    loader->corruption_reason);
        goto failed;
    }

    message = _dbus_message_loader_pop_message (loader);
    if (message != NULL)
    {
        _dbus_warn ("loaded message that was expected to be
incomplete\n");
        goto failed;
    }

    record_validity_seen (DBUS_VALID_BUT_INCOMPLETE);
    retval = TRUE;

failed:
    if (message)
        dbus_message_unref (message);
    return retval;
}

```

```

static dbus_bool_t
check_loader_results (DBusMessageLoader      *loader,
                    DBusValidity            expected_validity)
{
    if (!_dbus_message_loader_queue_messages (loader))
        _dbus_assert_not_reached ("no memory to queue messages");

    if (expected_validity == DBUS_VALID)
        return check_have_valid_message (loader);
    else if (expected_validity == DBUS_VALID_BUT_INCOMPLETE)
        return check_incomplete_message (loader);
    else if (expected_validity == DBUS_VALIDITY_UNKNOWN)
    {
        /* here we just know we didn't segfault and that was the
         * only test. Also, we record that we got coverage
         * for the validity reason.
         */
        if (_dbus_message_loader_get_is_corrupted (loader))
            record_validity_seen (loader->corruption_reason);

        return TRUE;
    }
    else
        return check_invalid_message (loader, expected_validity);
}

/**
 * Loads the message in the given message file.
 *
 * @param filename filename to load
 * @param data string to load message into
 * @returns #TRUE if the message was loaded
 */
dbus_bool_t
dbus_internal_do_not_use_load_message_file (const DBusString
*filename,
                                           DBusString      *data)
{
    dbus_bool_t retval;
    DBusError error = DBUS_ERROR_INIT;

    retval = FALSE;

    _dbus_verbose ("Loading raw %s\n", _dbus_string_get_const_data
(filename));
    if (!_dbus_file_get_contents (data, filename, &error))
    {
        _dbus_warn ("Could not load message file %s: %s\n",
                    _dbus_string_get_const_data (filename),
                    error.message);
        dbus_error_free (&error);
    }
}

```

```

        goto failed;
    }

    retval = TRUE;

failed:

    return retval;
}

/**
 * Tries loading the message in the given message file
 * and verifies that DbusMessageLoader can handle it.
 *
 * @param filename filename to load
 * @param expected_validity what the message has to be like to return
#TRUE
 * @returns #TRUE if the message has the expected validity
 */
dbus_bool_t
dbus_internal_do_not_use_try_message_file (const DbusString
*filename,
                                           DbusValidity
expected_validity)
{
    DbusString data;
    dbus_bool_t retval;

    retval = FALSE;

    if (!_dbus_string_init (&data))
        _dbus_assert_not_reached ("could not allocate string\n");

    if (!dbus_internal_do_not_use_load_message_file (filename, &data))
        goto failed;

    retval = dbus_internal_do_not_use_try_message_data (&data,
expected_validity);

failed:

    if (!retval)
    {
        if (_dbus_string_get_length (&data) > 0)
            _dbus_verbose_bytes_of_string (&data, 0,
                                           _dbus_string_get_length
(&data));

        _dbus_warn ("Failed message loader test on %s\n",
                    _dbus_string_get_const_data (filename));
    }
}

```

```

    _dbus_string_free (&data);

    return retval;
}

/**
 * Tries loading the given message data.
 *
 *
 * @param data the message data
 * @param expected_validity what the message has to be like to return
#TRUE
 * @returns #TRUE if the message has the expected validity
 */
dbus_bool_t
dbus_internal_do_not_use_try_message_data (const DBusString      *data,
                                           DBusValidity
                                           expected_validity)
{
    DBusMessageLoader *loader;
    dbus_bool_t retval;
    int len;
    int i;

    loader = NULL;
    retval = FALSE;

    /* Write the data one byte at a time */

    loader = _dbus_message_loader_new ();

    /* check some trivial loader functions */
    _dbus_message_loader_ref (loader);
    _dbus_message_loader_unref (loader);
    _dbus_message_loader_get_max_message_size (loader);

    len = _dbus_string_get_length (data);
    for (i = 0; i < len; i++)
    {
        DBusString *buffer;

        _dbus_message_loader_get_buffer (loader, &buffer);
        _dbus_string_append_byte (buffer,
                                  _dbus_string_get_byte (data, i));
        _dbus_message_loader_return_buffer (loader, buffer, 1);
    }

    if (!check_loader_results (loader, expected_validity))
        goto failed;

    _dbus_message_loader_unref (loader);
    loader = NULL;
}

```



```

/* Write the data all at once */

loader = _dbus_message_loader_new ();

{
    DBusString *buffer;

    _dbus_message_loader_get_buffer (loader, &buffer);
    _dbus_string_copy (data, 0, buffer,
                      _dbus_string_get_length (buffer));
    _dbus_message_loader_return_buffer (loader, buffer, 1);
}

if (!check_loader_results (loader, expected_validity))
    goto failed;

_dbus_message_loader_unref (loader);
loader = NULL;

/* Write the data 2 bytes at a time */

loader = _dbus_message_loader_new ();

len = _dbus_string_get_length (data);
for (i = 0; i < len; i += 2)
    {
        DBusString *buffer;

        _dbus_message_loader_get_buffer (loader, &buffer);
        _dbus_string_append_byte (buffer,
                                  _dbus_string_get_byte (data, i));
        if ((i+1) < len)
            _dbus_string_append_byte (buffer,
                                      _dbus_string_get_byte (data, i+1));
        _dbus_message_loader_return_buffer (loader, buffer, 1);
    }

if (!check_loader_results (loader, expected_validity))
    goto failed;

_dbus_message_loader_unref (loader);
loader = NULL;

retval = TRUE;

failed:

if (loader)
    _dbus_message_loader_unref (loader);

return retval;

```

```

}

static dbus_bool_t
process_test_subdir (const DBusString      *test_base_dir,
                    const char            *subdir,
                    DBusValidity          expected_validity,
                    DBusForeachMessageFileFunc function,
                    void                   *user_data)
{
    DBusString test_directory;
    DBusString filename;
    DBusDirIter *dir;
    dbus_bool_t retval;
    DBusError error = DBUS_ERROR_INIT;

    retval = FALSE;
    dir = NULL;

    if (!_dbus_string_init (&test_directory))
        _dbus_assert_not_reached ("didn't allocate test_directory\n");

    _dbus_string_init_const (&filename, subdir);

    if (!_dbus_string_copy (test_base_dir, 0,
                           &test_directory, 0))
        _dbus_assert_not_reached ("couldn't copy test_base_dir to
test_directory");

    if (!_dbus_concat_dir_and_file (&test_directory, &filename))
        _dbus_assert_not_reached ("couldn't allocate full path");

    _dbus_string_free (&filename);
    if (!_dbus_string_init (&filename))
        _dbus_assert_not_reached ("didn't allocate filename string\n");

    dir = _dbus_directory_open (&test_directory, &error);
    if (dir == NULL)
    {
        _dbus_warn ("Could not open %s: %s\n",
                   _dbus_string_get_const_data (&test_directory),
                   error.message);
        dbus_error_free (&error);
        goto failed;
    }

    printf ("Testing %s:\n", subdir);

next:
    while (_dbus_directory_get_next_file (dir, &filename, &error))
    {
        DBusString full_path;

```

```

if (!_dbus_string_init (&full_path))
    _dbus_assert_not_reached ("couldn't init string");

if (!_dbus_string_copy (&test_directory, 0, &full_path, 0))
    _dbus_assert_not_reached ("couldn't copy dir to full_path");

if (!_dbus_concat_dir_and_file (&full_path, &filename))
    _dbus_assert_not_reached ("couldn't concat file to dir");

if (_dbus_string_ends_with_c_str (&filename, ".message-raw"))
    ;
else
    {
        if (_dbus_string_ends_with_c_str (&filename, ".message"))
            {
                printf ("SKIP: Could not load %s, message builder
language no longer supported\n",
                        _dbus_string_get_const_data (&filename));
            }

        _dbus_verbose ("Skipping non-.message file %s\n",
                        _dbus_string_get_const_data (&filename));
        _dbus_string_free (&full_path);
        goto next;
    }

printf ("    %s\n",
        _dbus_string_get_const_data (&filename));

if (! (*function) (&full_path,
                  expected_validity, user_data))
    {
        _dbus_string_free (&full_path);
        goto failed;
    }
else
    _dbus_string_free (&full_path);
}

if (dbus_error_is_set (&error))
    {
        _dbus_warn ("Could not get next file in %s: %s\n",
                    _dbus_string_get_const_data (&test_directory),
                    error.message);
        dbus_error_free (&error);
        goto failed;
    }

retval = TRUE;

failed:

```

```

    if (dir)
        _dbus_directory_close (dir);
    _dbus_string_free (&test_directory);
    _dbus_string_free (&filename);

    return retval;
}

/**
 * Runs the given function on every message file in the test suite.
 * The function should return #FALSE on test failure or fatal error.
 *
 * @param test_data_dir root dir of the test suite data files
 * (top_srcdir/test/data)
 * @param func the function to run
 * @param user_data data for function
 * @returns #FALSE if there's a failure
 */
dbus_bool_t
dbus_internal_do_not_use_foreach_message_file (const char
*test_data_dir,

DBusForeachMessageFileFunc func,

                                void

*user_data)
{
    DBusString test_directory;
    dbus_bool_t retval;

    retval = FALSE;

    _dbus_string_init_const (&test_directory, test_data_dir);

    if (!process_test_subdir (&test_directory, "valid-messages",
                                DBUS_VALID, func, user_data))
        goto failed;

    check_memleaks ();

    if (!process_test_subdir (&test_directory, "invalid-messages",
                                DBUS_INVALID_FOR_UNKNOWN_REASON, func,
user_data))
        goto failed;

    check_memleaks ();

    if (!process_test_subdir (&test_directory, "incomplete-messages",
                                DBUS_VALID_BUT_INCOMPLETE, func,
user_data))
        goto failed;

    check_memleaks ();
}

```

```

    retval = TRUE;

failed:

    _dbus_string_free (&test_directory);

    return retval;
}

#if 0
#define GET_AND_CHECK(iter, typename, literal)
\
do {
\
    if (dbus_message_iter_get_arg_type (&iter) !=
DBUS_TYPE_##typename) \
        _dbus_assert_not_reached ("got wrong argument type from message
iter"); \
        dbus_message_iter_get_basic (&iter, &v_##typename);
\
    if (v_##typename != literal)
\
        _dbus_assert_not_reached ("got wrong value from message iter");
\
} while (0)

#define GET_AND_CHECK_STRCMP(iter, typename, literal)
\
do {
\
    if (dbus_message_iter_get_arg_type (&iter) !=
DBUS_TYPE_##typename) \
        _dbus_assert_not_reached ("got wrong argument type from message
iter"); \
        dbus_message_iter_get_basic (&iter, &v_##typename);
\
    if (strcmp (v_##typename, literal) != 0)
\
        _dbus_assert_not_reached ("got wrong value from message iter");
\
} while (0)

#define GET_AND_CHECK_AND_NEXT(iter, typename, literal) \
do { \
    GET_AND_CHECK(iter, typename, literal); \
    if (!dbus_message_iter_next (&iter)) \
        _dbus_assert_not_reached ("failed to move iter to next"); \
} while (0)

#define GET_AND_CHECK_STRCMP_AND_NEXT(iter, typename, literal) \
do { \

```

```

    GET_AND_CHECK_STRCMP(iter, typename, literal);           \
    if (!dbus_message_iter_next (&iter))                   \
        _dbus_assert_not_reached ("failed to move iter to next"); \
} while (0)

static void
message_iter_test (DBusMessage *message)
{
    DBusMessageIter iter, array, array2;
    const char *v_STRING;
    double v_DOUBLE;
    dbus_int16_t v_INT16;
    dbus_uint16_t v_UINT16;
    dbus_int32_t v_INT32;
    dbus_uint32_t v_UINT32;
#ifdef DBUS_HAVE_INT64
    dbus_int64_t v_INT64;
    dbus_uint64_t v_UINT64;
#endif
    unsigned char v_BYTE;
    dbus_bool_t v_BOOLEAN;

    const dbus_int32_t *our_int_array;
    int len;

    dbus_message_iter_init (message, &iter);

    GET_AND_CHECK_STRCMP_AND_NEXT (iter, STRING, "Test string");
    GET_AND_CHECK_AND_NEXT (iter, INT32, -0x12345678);
    GET_AND_CHECK_AND_NEXT (iter, UINT32, 0xeddle);
    GET_AND_CHECK_AND_NEXT (iter, DOUBLE, 3.14159);

    if (dbus_message_iter_get_arg_type (&iter) != DBUS_TYPE_ARRAY)
        _dbus_assert_not_reached ("Argument type not an array");

    if (dbus_message_iter_get_element_type (&iter) != DBUS_TYPE_DOUBLE)
        _dbus_assert_not_reached ("Array type not double");

    dbus_message_iter_recurse (&iter, &array);

    GET_AND_CHECK_AND_NEXT (array, DOUBLE, 1.5);
    GET_AND_CHECK (array, DOUBLE, 2.5);

    if (dbus_message_iter_next (&array))
        _dbus_assert_not_reached ("Didn't reach end of array");

    if (!dbus_message_iter_next (&iter))
        _dbus_assert_not_reached ("Reached end of arguments");

    GET_AND_CHECK_AND_NEXT (iter, BYTE, 0xF0);

    if (dbus_message_iter_get_arg_type (&iter) != DBUS_TYPE_ARRAY)

```

```

    _dbus_assert_not_reached ("no array");

    if (dbus_message_iter_get_element_type (&iter) != DBUS_TYPE_INT32)
        _dbus_assert_not_reached ("Array type not int32");

    /* Empty array */
    dbus_message_iter_recurse (&iter, &array);

    if (dbus_message_iter_next (&array))
        _dbus_assert_not_reached ("Didn't reach end of array");

    if (!dbus_message_iter_next (&iter))
        _dbus_assert_not_reached ("Reached end of arguments");

    GET_AND_CHECK (iter, BYTE, 0xF0);

    if (dbus_message_iter_next (&iter))
        _dbus_assert_not_reached ("Didn't reach end of arguments");
}
#endif

static void
verify_test_message (DBusMessage *message)
{
    DBusMessageIter iter;
    DBusError error = DBUS_ERROR_INIT;
    dbus_int16_t our_int16;
    dbus_uint16_t our_uint16;
    dbus_int32_t our_int;
    dbus_uint32_t our_uint;
    const char *our_str;
    double our_double;
    double v_DOUBLE;
    dbus_bool_t our_bool;
    unsigned char our_byte_1, our_byte_2;
    const dbus_uint32_t *our_uint32_array = (void*)0xdeadbeef;
    int our_uint32_array_len;
    dbus_int32_t *our_int32_array = (void*)0xdeadbeef;
    int our_int32_array_len;
#ifdef DBUS_HAVE_INT64
    dbus_int64_t our_int64;
    dbus_uint64_t our_uint64;
    dbus_int64_t *our_uint64_array = (void*)0xdeadbeef;
    int our_uint64_array_len;
    const dbus_int64_t *our_int64_array = (void*)0xdeadbeef;
    int our_int64_array_len;
#endif
    const double *our_double_array = (void*)0xdeadbeef;
    int our_double_array_len;
    const unsigned char *our_byte_array = (void*)0xdeadbeef;
    int our_byte_array_len;
    const dbus_bool_t *our_boolean_array = (void*)0xdeadbeef;

```

```

int our_boolean_array_len;
char **our_string_array;
int our_string_array_len;

dbus_message_iter_init (message, &iter);

if (!dbus_message_iter_get_args (&iter, &error,
                                DBUS_TYPE_INT16, &our_int16,
                                DBUS_TYPE_UINT16, &our_uint16,
                                DBUS_TYPE_INT32, &our_int,
                                DBUS_TYPE_UINT32, &our_uint,
#ifdef DBUS_HAVE_INT64
                                DBUS_TYPE_INT64, &our_int64,
                                DBUS_TYPE_UINT64, &our_uint64,
#endif
                                DBUS_TYPE_STRING, &our_str,
                                DBUS_TYPE_DOUBLE, &our_double,
                                DBUS_TYPE_BOOLEAN, &our_bool,
                                DBUS_TYPE_BYTE, &our_byte_1,
                                DBUS_TYPE_BYTE, &our_byte_2,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_UINT32,
                                &our_uint32_array,
                                &our_uint32_array_len,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_INT32,
                                &our_int32_array,
                                &our_int32_array_len,
#ifdef DBUS_HAVE_INT64
                                DBUS_TYPE_ARRAY, DBUS_TYPE_UINT64,
                                &our_uint64_array,
                                &our_uint64_array_len,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_INT64,
                                &our_int64_array,
                                &our_int64_array_len,
#endif
                                DBUS_TYPE_ARRAY, DBUS_TYPE_DOUBLE,
                                &our_double_array,
                                &our_double_array_len,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_BYTE,
                                &our_byte_array,
                                &our_byte_array_len,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_BOOLEAN,
                                &our_boolean_array,
                                &our_boolean_array_len,
                                DBUS_TYPE_ARRAY, DBUS_TYPE_STRING,
                                &our_string_array,
                                &our_string_array_len,
                                0))
{
    _dbus_warn ("error: %s - %s\n", error.name,
                (error.message != NULL) ? error.message : "no
message");
    _dbus_assert_not_reached ("Could not get arguments");
}

```



```

    }

    if (our_int16 != -0x123)
        _dbus_assert_not_reached ("16-bit integers differ!");

    if (our_uint16 != 0x123)
        _dbus_assert_not_reached ("16-bit uints differ!");

    if (our_int != -0x12345678)
        _dbus_assert_not_reached ("integers differ!");

    if (our_uint != 0x12300042)
        _dbus_assert_not_reached ("uints differ!");

#ifdef DBUS_HAVE_INT64
    if (our_int64 != DBUS_INT64_CONSTANT (-0x123456789abcd))
        _dbus_assert_not_reached ("64-bit integers differ!");
    if (our_uint64 != DBUS_UINT64_CONSTANT (0x123456789abcd))
        _dbus_assert_not_reached ("64-bit unsigned integers differ!");
#endif

    v_DOUBLE = 3.14159;
    if (! _DBUS_DOUBLES_BITWISE_EQUAL (our_double, v_DOUBLE))
        _dbus_assert_not_reached ("doubles differ!");

    if (strcmp (our_str, "Test string") != 0)
        _dbus_assert_not_reached ("strings differ!");

    if (!our_bool)
        _dbus_assert_not_reached ("booleans differ");

    if (our_byte_1 != 42)
        _dbus_assert_not_reached ("bytes differ!");

    if (our_byte_2 != 24)
        _dbus_assert_not_reached ("bytes differ!");

    if (our_uint32_array_len != 4 ||
        our_uint32_array[0] != 0x12345678 ||
        our_uint32_array[1] != 0x23456781 ||
        our_uint32_array[2] != 0x34567812 ||
        our_uint32_array[3] != 0x45678123)
        _dbus_assert_not_reached ("uint array differs");

    if (our_int32_array_len != 4 ||
        our_int32_array[0] != 0x12345678 ||
        our_int32_array[1] != -0x23456781 ||
        our_int32_array[2] != 0x34567812 ||
        our_int32_array[3] != -0x45678123)
        _dbus_assert_not_reached ("int array differs");

#ifdef DBUS_HAVE_INT64

```

```

if (our_uint64_array_len != 4 ||
    our_uint64_array[0] != 0x12345678 ||
    our_uint64_array[1] != 0x23456781 ||
    our_uint64_array[2] != 0x34567812 ||
    our_uint64_array[3] != 0x45678123)
    _dbus_assert_not_reached ("uint64 array differs");

if (our_int64_array_len != 4 ||
    our_int64_array[0] != 0x12345678 ||
    our_int64_array[1] != -0x23456781 ||
    our_int64_array[2] != 0x34567812 ||
    our_int64_array[3] != -0x45678123)
    _dbus_assert_not_reached ("int64 array differs");
#endif /* DBUS_HAVE_INT64 */

if (our_double_array_len != 3)
    _dbus_assert_not_reached ("double array had wrong length");

/* On all IEEE machines (i.e. everything sane) exact equality
 * should be preserved over the wire
 */
v_DOUBLE = 0.1234;
if (! _DBUS_DOUBLES_BITWISE_EQUAL (our_double_array[0], v_DOUBLE))
    _dbus_assert_not_reached ("double array had wrong values");
v_DOUBLE = 9876.54321;
if (! _DBUS_DOUBLES_BITWISE_EQUAL (our_double_array[1], v_DOUBLE))
    _dbus_assert_not_reached ("double array had wrong values");
v_DOUBLE = -300.0;
if (! _DBUS_DOUBLES_BITWISE_EQUAL (our_double_array[2], v_DOUBLE))
    _dbus_assert_not_reached ("double array had wrong values");

if (our_byte_array_len != 4)
    _dbus_assert_not_reached ("byte array had wrong length");

if (our_byte_array[0] != 'a' ||
    our_byte_array[1] != 'b' ||
    our_byte_array[2] != 'c' ||
    our_byte_array[3] != 234)
    _dbus_assert_not_reached ("byte array had wrong values");

if (our_boolean_array_len != 5)
    _dbus_assert_not_reached ("bool array had wrong length");

if (our_boolean_array[0] != TRUE ||
    our_boolean_array[1] != FALSE ||
    our_boolean_array[2] != TRUE ||
    our_boolean_array[3] != TRUE ||
    our_boolean_array[4] != FALSE)
    _dbus_assert_not_reached ("bool array had wrong values");

if (our_string_array_len != 4)
    _dbus_assert_not_reached ("string array was wrong length");

```

```

if (strcmp (our_string_array[0], "Foo") != 0 ||
    strcmp (our_string_array[1], "bar") != 0 ||
    strcmp (our_string_array[2], "") != 0 ||
    strcmp (our_string_array[3], "woo woo woo woo") != 0)
    _dbus_assert_not_reached ("string array had wrong values");

dbus_free_string_array (our_string_array);

if (dbus_message_iter_next (&iter))
    _dbus_assert_not_reached ("Didn't reach end of arguments");
}

/**
 * @ingroup DBusMessageInternals
 * Unit test for DBusMessage.
 *
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_message_test (const char *test_data_dir)
{
    DBusMessage *message, *message_without_unix_fds;
    DBusMessageLoader *loader;
    int i;
    const char *data;
    DBusMessage *copy;
    const char *name1;
    const char *name2;
    const dbus_uint32_t our_uint32_array[] =
        { 0x12345678, 0x23456781, 0x34567812, 0x45678123 };
    const dbus_int32_t our_int32_array[] =
        { 0x12345678, -0x23456781, 0x34567812, -0x45678123 };
    const dbus_uint32_t *v_ARRAY_UINT32 = our_uint32_array;
    const dbus_int32_t *v_ARRAY_INT32 = our_int32_array;
#ifdef DBUS_HAVE_INT64
    const dbus_uint64_t our_uint64_array[] =
        { 0x12345678, 0x23456781, 0x34567812, 0x45678123 };
    const dbus_int64_t our_int64_array[] =
        { 0x12345678, -0x23456781, 0x34567812, -0x45678123 };
    const dbus_uint64_t *v_ARRAY_UINT64 = our_uint64_array;
    const dbus_int64_t *v_ARRAY_INT64 = our_int64_array;
#endif
    const char *our_string_array[] = { "Foo", "bar", "", "woo woo woo
woo" };
    const char **v_ARRAY_STRING = our_string_array;
    const double our_double_array[] = { 0.1234, 9876.54321, -300.0 };
    const double *v_ARRAY_DOUBLE = our_double_array;
    const unsigned char our_byte_array[] = { 'a', 'b', 'c', 234 };
    const unsigned char *v_ARRAY_BYTE = our_byte_array;
    const dbus_bool_t our_boolean_array[] = { TRUE, FALSE, TRUE, TRUE,
FALSE };

```

```

const dbus_bool_t *v_ARRAY_BOOLEAN = our_boolean_array;
char sig[64];
const char *s;
const char *v_STRING;
double v_DOUBLE;
dbus_int16_t v_INT16;
dbus_uint16_t v_UINT16;
dbus_int32_t v_INT32;
dbus_uint32_t v_UINT32;
#ifdef DBUS_HAVE_INT64
dbus_int64_t v_INT64;
dbus_uint64_t v_UINT64;
#endif
unsigned char v_BYTE;
unsigned char v2_BYTE;
dbus_bool_t v_BOOLEAN;
DBusMessageIter iter, array_iter, struct_iter;
#ifdef HAVE_UNIX_FD_PASSING
int v_UNIX_FD;
#endif
char **decomposed;
DBusInitialFDs *initial_fds;

initial_fds = _dbus_check_fdleaks_enter ();

message = dbus_message_new_method_call
("org.freedesktop.DBus.TestService",
                                     "/org/freedesktop/TestPath",
                                     "Foo.TestInterface",
                                     "TestMethod");
_dbus_assert (dbus_message_has_destination (message,
"org.freedesktop.DBus.TestService"));
_dbus_assert (dbus_message_is_method_call (message,
"Foo.TestInterface",
                                     "TestMethod"));
_dbus_assert (strcmp (dbus_message_get_path (message),
                                     "/org/freedesktop/TestPath") == 0);
dbus_message_set_serial (message, 1234);

/* string length including nul byte not a multiple of 4 */
if (!dbus_message_set_sender (message, "org.foo.bar1"))
    _dbus_assert_not_reached ("out of memory");

_dbus_assert (dbus_message_has_sender (message, "org.foo.bar1"));
dbus_message_set_reply_serial (message, 5678);

_dbus_verbose_bytes_of_string (&message->header.data, 0,
    _dbus_string_get_length (&message-
>header.data));
_dbus_verbose_bytes_of_string (&message->body, 0,
    _dbus_string_get_length (&message-
>body));

```

```

if (!dbus_message_set_sender (message, NULL))
    _dbus_assert_not_reached ("out of memory");

    _dbus_verbose_bytes_of_string (&message->header.data, 0,
        _dbus_string_get_length (&message-
>header.data));
    _dbus_verbose_bytes_of_string (&message->body, 0,
        _dbus_string_get_length (&message-
>body));

    _dbus_assert (!dbus_message_has_sender (message, "org.foo.bar1"));
    _dbus_assert (dbus_message_get_serial (message) == 1234);
    _dbus_assert (dbus_message_get_reply_serial (message) == 5678);
    _dbus_assert (dbus_message_has_destination (message,
"org.freedesktop.DBus.TestService"));

    _dbus_assert (dbus_message_get_no_reply (message) == FALSE);
    dbus_message_set_no_reply (message, TRUE);
    _dbus_assert (dbus_message_get_no_reply (message) == TRUE);
    dbus_message_set_no_reply (message, FALSE);
    _dbus_assert (dbus_message_get_no_reply (message) == FALSE);

/* Set/get some header fields */

if (!dbus_message_set_path (message, "/foo"))
    _dbus_assert_not_reached ("out of memory");
    _dbus_assert (strcmp (dbus_message_get_path (message),
        "/foo") == 0);

if (!dbus_message_set_interface (message, "org.Foo"))
    _dbus_assert_not_reached ("out of memory");
    _dbus_assert (strcmp (dbus_message_get_interface (message),
        "org.Foo") == 0);

if (!dbus_message_set_member (message, "Bar"))
    _dbus_assert_not_reached ("out of memory");
    _dbus_assert (strcmp (dbus_message_get_member (message),
        "Bar") == 0);

/* Set/get them with longer values */
if (!dbus_message_set_path (message, "/foo/bar"))
    _dbus_assert_not_reached ("out of memory");
    _dbus_assert (strcmp (dbus_message_get_path (message),
        "/foo/bar") == 0);

if (!dbus_message_set_interface (message, "org.Foo.Bar"))
    _dbus_assert_not_reached ("out of memory");
    _dbus_assert (strcmp (dbus_message_get_interface (message),
        "org.Foo.Bar") == 0);

```

```

if (!dbus_message_set_member (message, "BarFoo"))
    _dbus_assert_not_reached ("out of memory");
_dbus_assert (strcmp (dbus_message_get_member (message),
                    "BarFoo") == 0);

/* Realloc shorter again */

if (!dbus_message_set_path (message, "/foo"))
    _dbus_assert_not_reached ("out of memory");
_dbus_assert (strcmp (dbus_message_get_path (message),
                    "/foo") == 0);

if (!dbus_message_set_interface (message, "org.Foo"))
    _dbus_assert_not_reached ("out of memory");
_dbus_assert (strcmp (dbus_message_get_interface (message),
                    "org.Foo") == 0);

if (!dbus_message_set_member (message, "Bar"))
    _dbus_assert_not_reached ("out of memory");
_dbus_assert (strcmp (dbus_message_get_member (message),
                    "Bar") == 0);

/* Path decomposing */
dbus_message_set_path (message, NULL);
dbus_message_get_path_decomposed (message, &decomposed);
_dbus_assert (decomposed == NULL);
dbus_free_string_array (decomposed);

dbus_message_set_path (message, "/");
dbus_message_get_path_decomposed (message, &decomposed);
_dbus_assert (decomposed != NULL);
_dbus_assert (decomposed[0] == NULL);
dbus_free_string_array (decomposed);

dbus_message_set_path (message, "/a/b");
dbus_message_get_path_decomposed (message, &decomposed);
_dbus_assert (decomposed != NULL);
_dbus_assert (strcmp (decomposed[0], "a") == 0);
_dbus_assert (strcmp (decomposed[1], "b") == 0);
_dbus_assert (decomposed[2] == NULL);
dbus_free_string_array (decomposed);

dbus_message_set_path (message, "/spam/eggs");
dbus_message_get_path_decomposed (message, &decomposed);
_dbus_assert (decomposed != NULL);
_dbus_assert (strcmp (decomposed[0], "spam") == 0);
_dbus_assert (strcmp (decomposed[1], "eggs") == 0);
_dbus_assert (decomposed[2] == NULL);
dbus_free_string_array (decomposed);

dbus_message_unref (message);

```



```

        DBUS_TYPE_ARRAY, DBUS_TYPE_INT64,
&v_ARRAY_INT64,
        _DBUS_N_ELEMENTS (our_int64_array),
#endif
        DBUS_TYPE_ARRAY, DBUS_TYPE_DOUBLE,
&v_ARRAY_DOUBLE,
        _DBUS_N_ELEMENTS (our_double_array),
        DBUS_TYPE_ARRAY, DBUS_TYPE_BYTE,
&v_ARRAY_BYTE,
        _DBUS_N_ELEMENTS (our_byte_array),
        DBUS_TYPE_ARRAY, DBUS_TYPE_BOOLEAN,
&v_ARRAY_BOOLEAN,
        _DBUS_N_ELEMENTS (our_boolean_array),
        DBUS_TYPE_ARRAY, DBUS_TYPE_STRING,
&v_ARRAY_STRING,
        _DBUS_N_ELEMENTS (our_string_array),

        DBUS_TYPE_INVALID);

```

```

i = 0;
sig[i++] = DBUS_TYPE_INT16;
sig[i++] = DBUS_TYPE_UINT16;
sig[i++] = DBUS_TYPE_INT32;
sig[i++] = DBUS_TYPE_UINT32;
#ifdef DBUS_HAVE_INT64
sig[i++] = DBUS_TYPE_INT64;
sig[i++] = DBUS_TYPE_UINT64;
#endif
sig[i++] = DBUS_TYPE_STRING;
sig[i++] = DBUS_TYPE_DOUBLE;
sig[i++] = DBUS_TYPE_BOOLEAN;
sig[i++] = DBUS_TYPE_BYTE;
sig[i++] = DBUS_TYPE_BYTE;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_UINT32;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_INT32;
#ifdef DBUS_HAVE_INT64
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_UINT64;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_INT64;
#endif
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_DOUBLE;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_BYTE;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_BOOLEAN;
sig[i++] = DBUS_TYPE_ARRAY;
sig[i++] = DBUS_TYPE_STRING;

```



```

message_without_unix_fds = dbus_message_copy(message);
_dbus_assert(message_without_unix_fds);
#ifdef HAVE_UNIX_FD_PASSING
    dbus_message_append_args (message,
                              DBUS_TYPE_UNIX_FD, &v_UNIX_FD,
                              DBUS_TYPE_INVALID);
    sig[i++] = DBUS_TYPE_UNIX_FD;
#else
    sig[i++] = DBUS_TYPE_INVALID;

    _dbus_assert (i < (int) _DBUS_N_ELEMENTS (sig));

    _dbus_verbose ("HEADER\n");
    _dbus_verbose_bytes_of_string (&message->header.data, 0,
                                   _dbus_string_get_length (&message-
>header.data));
    _dbus_verbose ("BODY\n");
    _dbus_verbose_bytes_of_string (&message->body, 0,
                                   _dbus_string_get_length (&message-
>body));

    _dbus_verbose ("Signature expected \"%s\" actual \"%s\"\n",
                  sig, dbus_message_get_signature (message));

    s = dbus_message_get_signature (message);

    _dbus_assert (dbus_message_has_signature (message, sig));
    _dbus_assert (strcmp (s, sig) == 0);

    verify_test_message (message);

    copy = dbus_message_copy (message);

    _dbus_assert (dbus_message_get_reply_serial (message) ==
                  dbus_message_get_reply_serial (copy));
    _dbus_assert (message->header.padding == copy->header.padding);

    _dbus_assert (_dbus_string_get_length (&message->header.data) ==
                  _dbus_string_get_length (&copy->header.data));

    _dbus_assert (_dbus_string_get_length (&message->body) ==
                  _dbus_string_get_length (&copy->body));

    verify_test_message (copy);

    name1 = dbus_message_get_interface (message);
    name2 = dbus_message_get_interface (copy);

    _dbus_assert (strcmp (name1, name2) == 0);

    name1 = dbus_message_get_member (message);
    name2 = dbus_message_get_member (copy);

```

```

_dbus_assert (strcmp (name1, name2) == 0);

dbus_message_unref (copy);

/* Message loader test */
dbus_message_lock (message);
loader = _dbus_message_loader_new ();

/* check ref/unref */
_dbus_message_loader_ref (loader);
_dbus_message_loader_unref (loader);

/* Write the header data one byte at a time */
data = _dbus_string_get_const_data (&message->header.data);
for (i = 0; i < _dbus_string_get_length (&message->header.data);
i++)
{
    DBusString *buffer;

    _dbus_message_loader_get_buffer (loader, &buffer);
    _dbus_string_append_byte (buffer, data[i]);
    _dbus_message_loader_return_buffer (loader, buffer, 1);
}

/* Write the body data one byte at a time */
data = _dbus_string_get_const_data (&message->body);
for (i = 0; i < _dbus_string_get_length (&message->body); i++)
{
    DBusString *buffer;

    _dbus_message_loader_get_buffer (loader, &buffer);
    _dbus_string_append_byte (buffer, data[i]);
    _dbus_message_loader_return_buffer (loader, buffer, 1);
}

#ifdef HAVE_UNIX_FD_PASSING
{
    int *unix_fds;
    unsigned n_unix_fds;
    /* Write unix fd */
    _dbus_message_loader_get_unix_fds(loader, &unix_fds, &n_unix_fds);
    _dbus_assert(n_unix_fds > 0);
    _dbus_assert(message->n_unix_fds == 1);
    unix_fds[0] = _dbus_dup(message->unix_fds[0], NULL);
    _dbus_assert(unix_fds[0] >= 0);
    _dbus_message_loader_return_unix_fds(loader, unix_fds, 1);
}
#endif

dbus_message_unref (message);

```

```

/* Now pop back the message */
if (!_dbus_message_loader_queue_messages (loader))
    _dbus_assert_not_reached ("no memory to queue messages");

if (_dbus_message_loader_get_is_corrupted (loader))
    _dbus_assert_not_reached ("message loader corrupted");

message = _dbus_message_loader_pop_message (loader);
if (!message)
    _dbus_assert_not_reached ("received a NULL message");

if (dbus_message_get_reply_serial (message) != 5678)
    _dbus_assert_not_reached ("reply serial fields differ");

dbus_message_unref (message);

/* override the serial, since it was reset by dbus_message_copy() */
dbus_message_set_serial(message_without_unix_fds, 8901);

dbus_message_lock (message_without_unix_fds);

verify_test_message (message_without_unix_fds);

{
    /* Marshal and demarshal the message. */

    DBusMessage *message2;
    DBusError error = DBUS_ERROR_INIT;
    char *marshalled = NULL;
    int len = 0;
    char garbage_header[DBUS_MINIMUM_HEADER_SIZE] = "xxx";

    if (!_dbus_message_marshal (message_without_unix_fds,
&marshalled, &len))
        _dbus_assert_not_reached ("failed to marshal message");

    _dbus_assert (len != 0);
    _dbus_assert (marshalled != NULL);

    _dbus_assert (dbus_message_demarshal_bytes_needed (marshalled,
len) == len);
    message2 = dbus_message_demarshal (marshalled, len, &error);

    _dbus_assert (message2 != NULL);
    _dbus_assert (!dbus_error_is_set (&error));
    verify_test_message (message2);

    dbus_message_unref (message2);
    dbus_free (marshalled);

    /* Demarshal invalid message. */

```

```

message2 = dbus_message_demarshal ("invalid", 7, &error);
_dbus_assert (message2 == NULL);
_dbus_assert (dbus_error_is_set (&error));
dbus_error_free (&error);

/* Demarshal invalid (empty) message. */

message2 = dbus_message_demarshal ("", 0, &error);
_dbus_assert (message2 == NULL);
_dbus_assert (dbus_error_is_set (&error));
dbus_error_free (&error);

/* Bytes needed to demarshal empty message: 0 (more) */

_dbus_assert (dbus_message_demarshal_bytes_needed ("", 0) == 0);

/* Bytes needed to demarshal invalid message: -1 (error). */

_dbus_assert (dbus_message_demarshal_bytes_needed
(garbage_header, DBUS_MINIMUM_HEADER_SIZE) == -1);
}

dbus_message_unref (message_without_unix_fds);
_dbus_message_loader_unref (loader);

check_memleaks ();
_dbus_check_fdleaks_leave (initial_fds);
initial_fds = _dbus_check_fdleaks_enter ();

/* Check that we can abandon a container */
message = dbus_message_new_method_call
("org.freedesktop.DBus.TestService",
    "/org/freedesktop/TestPath",
    "Foo.TestInterface",
    "Method");

dbus_message_iter_init_append (message, &iter);

_dbus_assert (dbus_message_iter_open_container (&iter,
DBUS_TYPE_ARRAY,
    (DBUS_STRUCT_BEGIN_CHAR_AS_STRING
    DBUS_TYPE_STRING_AS_STRING
    DBUS_TYPE_STRING_AS_STRING
    DBUS_STRUCT_END_CHAR_AS_STRING),
    &array_iter));

_dbus_assert (dbus_message_iter_open_container (&array_iter,
DBUS_TYPE_STRUCT,
    NULL, &struct_iter));

s = "peaches";
_dbus_assert (dbus_message_iter_append_basic (&struct_iter,
DBUS_TYPE_STRING,

```

```

                                &s));

/* uh-oh, error, try and unwind */

dbus_message_iter_abandon_container (&array_iter, &struct_iter);
dbus_message_iter_abandon_container (&array_iter, &iter);

dbus_message_unref (message);

/* Load all the sample messages from the message factory */
{
    DBusMessageDataIter diter;
    DBusMessageData mdata;
    int count;

    reset_validities_seen ();

    count = 0;
    _dbus_message_data_iter_init (&diter);

    while (_dbus_message_data_iter_get_and_next (&diter,
                                                &mdata))
    {
        if (!dbus_internal_do_not_use_try_message_data (&mdata.data,
mdata.expected_validity))
        {
            _dbus_warn ("expected validity %d and did not get it\n",
                        mdata.expected_validity);
            _dbus_assert_not_reached ("message data failed");
        }

        _dbus_message_data_free (&mdata);

        count += 1;
    }

    printf ("%d sample messages tested\n", count);

    print_validities_seen (FALSE);
    print_validities_seen (TRUE);
}

check_memleaks ();
_dbus_check_fdleaks_leave (initial_fds);

/* Now load every message in test_data_dir if we have one */
if (test_data_dir == NULL)
    return TRUE;

initial_fds = _dbus_check_fdleaks_enter ();

```

```

    if (!dbus_internal_do_not_use_foreach_message_file (test_data_dir,
(DBusForeachMessageFileFunc)
dbus_internal_do_not_use_try_message_file,
                                NULL))
        _dbus_assert_not_reached ("foreach_message_file test failed");
    _dbus_check_fdleaks_leave (initial_fds);
    return TRUE;
}

```

```
#endif /* DBUS_BUILD_TESTS */
```

```
File = dbus-message.c
```

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message.c  DBusMessage object
 *
 * Copyright (C) 2002, 2003, 2004, 2005 Red Hat Inc.
 * Copyright (C) 2002, 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

```

```

#include <config.h>
#include "dbus-internals.h"
#include "dbus-marshall-recursive.h"
#include "dbus-marshall-validate.h"
#include "dbus-marshall-byteswap.h"
#include "dbus-marshall-header.h"

```

```

#include "dbus-signature.h"
#include "dbus-message-private.h"
#include "dbus-object-tree.h"
#include "dbus-memory.h"
#include "dbus-list.h"
#include "dbus-threads-internal.h"
#ifdef HAVE_UNIX_FD_PASSING
#include "dbus-sysdeps-unix.h"
#endif

#include <string.h>

#define _DBUS_TYPE_IS_STRINGLIKE(type) \
    (type == DBUS_TYPE_STRING || type == DBUS_TYPE_SIGNATURE || \
     type == DBUS_TYPE_OBJECT_PATH)

static void dbus_message_finalize (DBusMessage *message);

/**
 * @defgroup DBusMessageInternals DBusMessage implementation details
 * @ingroup DBusInternals
 * @brief DBusMessage private implementation details.
 *
 * The guts of DBusMessage and its methods.
 *
 * @{
 */

#ifdef DBUS_BUILD_TESTS
static dbus_bool_t
_dbus_enable_message_cache (void)
{
    static int enabled = -1;

    if (enabled < 0)
    {
        const char *s = _dbus_getenv ("DBUS_MESSAGE_CACHE");

        enabled = TRUE;

        if (s && *s)
        {
            if (*s == '0')
                enabled = FALSE;
            else if (*s == '1')
                enabled = TRUE;
            else
                _dbus_warn ("DBUS_MESSAGE_CACHE should be 0 or 1 if set,
not '%s'",
                            s);
        }
    }
}

```

```

    return enabled;
}
#else
    /* constant expression, should be optimized away */
#   define _dbus_enable_message_cache() (TRUE)
#endif

#ifdef _dbus_message_trace_ref
void
_dbus_message_trace_ref (DBusMessage *message,
                        int          old_refcount,
                        int          new_refcount,
                        const char   *why)
{
    static int enabled = -1;

    _dbus_trace_ref ("DBusMessage", message, old_refcount, new_refcount,
                    "DBUS_MESSAGE_TRACE", &enabled);
}
#endif

/* Not thread locked, but strictly const/read-only so should be OK
 */
/** An static string representing an empty signature */
_DBUS_STRING_DEFINE_STATIC(_dbus_empty_signature_str, "");

/* these have wacky values to help trap uninitialized iterators;
 * but has to fit in 3 bits
 */
enum {
    DBUS_MESSAGE_ITER_TYPE_READER = 3,
    DBUS_MESSAGE_ITER_TYPE_WRITER = 7
};

/** typedef for internals of message iterator */
typedef struct DBusMessageRealIter DBusMessageRealIter;

/**
 * @brief Internals of DBusMessageIter
 *
 * Object representing a position in a message. All fields are
 * internal.
 */
struct DBusMessageRealIter
{
    DBusMessage *message; /**< Message used */
    dbus_uint32_t changed_stamp : CHANGED_STAMP_BITS; /**< stamp to
detect invalid iters */
    dbus_uint32_t iter_type : 3;          /**< whether this is a reader or
writer iter */

```



```

        &message->body, 0);

    _dbus_header_byteswap (&message->header, DBUS_COMPILER_BYTE_ORDER);
    _dbus_assert (_dbus_header_get_byte_order (&message->header) ==
                  DBUS_COMPILER_BYTE_ORDER);
}

/** byte-swap the message if it doesn't match our byte order.
 * Called only when we need the message in our own byte order,
 * normally when reading arrays of integers or doubles.
 * Otherwise should not be called since it would do needless
 * work.
 */
#define ensure_byte_order(message) _dbus_message_byteswap (message)

/**
 * Gets the data to be sent over the network for this message.
 * The header and then the body should be written out.
 * This function is guaranteed to always return the same
 * data once a message is locked (with dbus_message_lock()).
 *
 * @param message the message.
 * @param header return location for message header data.
 * @param body return location for message body data.
 */
void
_dbus_message_get_network_data (DBusMessage      *message,
                               const DBusString **header,
                               const DBusString **body)
{
    _dbus_assert (message->locked);

    *header = &message->header.data;
    *body = &message->body;
}

/**
 * Gets the unix fds to be sent over the network for this message.
 * This function is guaranteed to always return the same data once a
 * message is locked (with dbus_message_lock()).
 *
 * @param message the message.
 * @param fds return location of unix fd array
 * @param n_fds return number of entries in array
 */
void _dbus_message_get_unix_fds (DBusMessage *message,
                                const int    **fds,
                                unsigned     *n_fds)
{
    _dbus_assert (message->locked);

#ifdef HAVE_UNIX_FD_PASSING

```

```

    *fds = message->unix_fds;
    *n_fds = message->n_unix_fds;
#else
    *fds = NULL;
    *n_fds = 0;
#endif
}

/**
 * Sets the serial number of a message.
 * This can only be done once on a message.
 *
 * DbusConnection will automatically set the serial to an appropriate
value
 * when the message is sent; this function is only needed when
encapsulating
 * messages in another protocol, or otherwise bypassing
DBusConnection.
 *
 * @param message the message
 * @param serial the serial
 */
void
dbus_message_set_serial (DBusMessage *message,
                        dbus_uint32_t serial)
{
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (!message->locked);

    _dbus_header_set_serial (&message->header, serial);
}

/**
 * Adds a counter to be incremented immediately with the size/unix fds
 * of this message, and decremented by the size/unix fds of this
 * message when this message is finalized. The link contains a
 * counter with its refcount already incremented, but the counter
 * itself not incremented. Ownership of link and counter refcount is
 * passed to the message.
 *
 * This function may be called with locks held. As a result, the
counter's
 * notify function is not called; the caller is expected to either
call
 * _dbus_counter_notify() on the counter when they are no longer
holding
 * locks, or take the same action that would be taken by the notify
function.
 *
 * @param message the message
 * @param link link with counter as data
 */

```

```

void
_dbus_message_add_counter_link (DBusMessage *message,
                                DBusList *link)
{
    /* right now we don't recompute the delta when message
     * size changes, and that's OK for current purposes
     * I think, but could be important to change later.
     * Do recompute it whenever there are no outstanding counters,
     * since it's basically free.
     */
    if (message->counters == NULL)
    {
        message->size_counter_delta =
            _dbus_string_get_length (&message->header.data) +
            _dbus_string_get_length (&message->body);

#ifdef HAVE_UNIX_FD_PASSING
        message->unix_fd_counter_delta = message->n_unix_fds;
#endif

#ifdef 0
        _dbus_verbose ("message has size %ld\n",
                       message->size_counter_delta);
#endif
    }

    _dbus_list_append_link (&message->counters, link);

    _dbus_counter_adjust_size (link->data, message->size_counter_delta);

#ifdef HAVE_UNIX_FD_PASSING
    _dbus_counter_adjust_unix_fd (link->data, message-
>unix_fd_counter_delta);
#endif
}

/**
 * Adds a counter to be incremented immediately with the size/unix fds
 * of this message, and decremented by the size/unix fds of this
 * message when this message is finalized.
 *
 * This function may be called with locks held. As a result, the
 * counter's
 * notify function is not called; the caller is expected to either
 * call
 * _dbus_counter_notify() on the counter when they are no longer
 * holding
 * locks, or take the same action that would be taken by the notify
 * function.
 *
 * @param message the message
 * @param counter the counter

```

```

    * @returns #FALSE if no memory
    */
dbus_bool_t
_dbus_message_add_counter (DBusMessage *message,
                           DBusCounter *counter)
{
    DBusList *link;

    link = _dbus_list_alloc_link (counter);
    if (link == NULL)
        return FALSE;

    _dbus_counter_ref (counter);
    _dbus_message_add_counter_link (message, link);

    return TRUE;
}

/**
 * Removes a counter tracking the size/unix fds of this message, and
 * decrements the counter by the size/unix fds of this message.
 *
 * @param message the message
 * @param counter the counter
 */
void
_dbus_message_remove_counter (DBusMessage *message,
                              DBusCounter *counter)
{
    DBusList *link;

    link = _dbus_list_find_last (&message->counters,
                                counter);
    _dbus_assert (link != NULL);

    _dbus_list_remove_link (&message->counters, link);

    _dbus_counter_adjust_size (counter, - message->size_counter_delta);

#ifdef HAVE_UNIX_FD_PASSING
    _dbus_counter_adjust_unix_fd (counter, - message->
    >unix_fd_counter_delta);
#endif

    _dbus_counter_notify (counter);
    _dbus_counter_unref (counter);
}

/**
 * Locks a message. Allows checking that applications don't keep a
 * reference to a message in the outgoing queue and change it
 * underneath us. Messages are locked when they enter the outgoing

```

```

* queue (dbus_connection_send_message()), and the library complains
* if the message is modified while locked. This function may also
* called externally, for applications wrapping D-Bus in another
protocol.
*
* @param message the message to lock.
*/
void
dbus_message_lock (DBusMessage *message)
{
    if (!message->locked)
    {
        _dbus_header_update_lengths (&message->header,
                                     _dbus_string_get_length (&message-
>body));

        /* must have a signature if you have a body */
        _dbus_assert (_dbus_string_get_length (&message->body) == 0 ||
                     dbus_message_get_signature (message) != NULL);

        message->locked = TRUE;
    }
}

static dbus_bool_t
set_or_delete_string_field (DBusMessage *message,
                            int          field,
                            int          typecode,
                            const char *value)
{
    if (value == NULL)
        return _dbus_header_delete_field (&message->header, field);
    else
        return _dbus_header_set_field_basic (&message->header,
                                             field,
                                             typecode,
                                             &value);
}

#if 0
/* Probably we don't need to use this */
/**
 * Sets the signature of the message, i.e. the arguments in the
 * message payload. The signature includes only "in" arguments for
 * #DBUS_MESSAGE_TYPE_METHOD_CALL and only "out" arguments for
 * #DBUS_MESSAGE_TYPE_METHOD_RETURN, so is slightly different from
 * what you might expect (it does not include the signature of the
 * entire C++-style method).
 *
 * The signature is a string made up of type codes such as
 * #DBUS_TYPE_INT32. The string is terminated with nul (nul is also
 * the value of #DBUS_TYPE_INVALID). The macros such as

```

```

* #DBUS_TYPE_INT32 evaluate to integers; to assemble a signature you
* may find it useful to use the string forms, such as
* #DBUS_TYPE_INT32_AS_STRING.
*
* An "unset" or #NULL signature is considered the same as an empty
* signature. In fact dbus_message_get_signature() will never return
* #NULL.
*
* @param message the message
* @param signature the type signature or #NULL to unset
* @returns #FALSE if no memory
*/
static dbus_bool_t
_dbus_message_set_signature (DBusMessage *message,
                             const char *signature)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (signature == NULL ||
                              _dbus_check_is_valid_signature
                              (signature));
    /* can't delete the signature if you have a message body */
    _dbus_return_val_if_fail (_dbus_string_get_length (&message->body)
                              == 0 ||
                              signature != NULL);

    return set_or_delete_string_field (message,
                                       DBUS_HEADER_FIELD_SIGNATURE,
                                       DBUS_TYPE_SIGNATURE,
                                       signature);
}
#endif

/* Message Cache
*
* We cache some DBusMessage to reduce the overhead of allocating
* them. In my profiling this consistently made about an 8%
* difference. It avoids the malloc for the message, the malloc for
* the slot list, the malloc for the header string and body string,
* and the associated free() calls. It does introduce another global
* lock which could be a performance issue in certain cases.
*
* For the echo client/server the round trip time goes from around
* .000077 to .000069 with the message cache on my laptop. The sysprof
* change is as follows (numbers are cumulative percentage):
*
* with message cache implemented as array as it is now (0.000069
per):
*   new_empty_header          1.46
*   mutex_lock                0.56   # i.e.
_DBUS_LOCK(message_cache)
*   mutex_unlock              0.25

```

```

*      self                0.41
*      unref                2.24
*      self                0.68
*      list_clear          0.43
*      mutex_lock          0.33      # i.e.
_DBUS_LOCK(message_cache)
*      mutex_unlock        0.25
*
* with message cache implemented as list (0.000070 per roundtrip):
*      new_empty_header    2.72
*      list_pop_first      1.88
*      unref                3.3
*      list_prepend        1.63
*
* without cache (0.000077 per roundtrip):
*      new_empty_header    6.7
*      string_init_preallocated 3.43
*      dbus_malloc         2.43
*      dbus_malloc0        2.59
*
*      unref                4.02
*      string_free         1.82
*      dbus_free           1.63
*      dbus_free           0.71
*
* If you implement the message_cache with a list, the primary reason
* it's slower is that you add another thread lock (on the DBusList
* mempool).
*/

/** Avoid caching huge messages */
#define MAX_MESSAGE_SIZE_TO_CACHE 10 * _DBUS_ONE_KILOBYTE

/** Avoid caching too many messages */
#define MAX_MESSAGE_CACHE_SIZE 5

_DBUS_DEFINE_GLOBAL_LOCK (message_cache);
static DBusMessage *message_cache[MAX_MESSAGE_CACHE_SIZE];
static int message_cache_count = 0;
static dbus_bool_t message_cache_shutdown_registered = FALSE;

static void
dbus_message_cache_shutdown (void *data)
{
    int i;

    _DBUS_LOCK (message_cache);

    i = 0;
    while (i < MAX_MESSAGE_CACHE_SIZE)
    {
        if (message_cache[i])

```



```

        dbus_message_finalize (message_cache[i]);

        ++i;
    }

    message_cache_count = 0;
    message_cache_shutdown_registered = FALSE;

    _DBUS_UNLOCK (message_cache);
}

/**
 * Tries to get a message from the message cache.  The retrieved
 * message will have junk in it, so it still needs to be cleared out
 * in dbus_message_new_empty_header()
 *
 * @returns the message, or #NULL if none cached
 */
static DBusMessage*
dbus_message_get_cached (void)
{
    DBusMessage *message;
    int i;

    message = NULL;

    _DBUS_LOCK (message_cache);

    _dbus_assert (message_cache_count >= 0);

    if (message_cache_count == 0)
    {
        _DBUS_UNLOCK (message_cache);
        return NULL;
    }

    /* This is not necessarily true unless count > 0, and
     * message_cache is uninitialized until the shutdown is
     * registered
     */
    _dbus_assert (message_cache_shutdown_registered);

    i = 0;
    while (i < MAX_MESSAGE_CACHE_SIZE)
    {
        if (message_cache[i])
        {
            message = message_cache[i];
            message_cache[i] = NULL;
            message_cache_count -= 1;
            break;
        }
    }
}

```

```

        ++i;
    }
    _dbus_assert (message_cache_count >= 0);
    _dbus_assert (i < MAX_MESSAGE_CACHE_SIZE);
    _dbus_assert (message != NULL);

    _dbus_assert (_dbus_atomic_get (&message->refcount) == 0);

    _dbus_assert (message->counters == NULL);

    _DBUS_UNLOCK (message_cache);

    return message;
}

#ifdef HAVE_UNIX_FD_PASSING
static void
close_unix_fds(int *fds, unsigned *n_fds)
{
    DBusError e;
    int i;

    if (*n_fds <= 0)
        return;

    dbus_error_init(&e);

    for (i = 0; i < *n_fds; i++)
    {
        if (!_dbus_close(fds[i], &e))
        {
            _dbus_warn("Failed to close file descriptor: %s\n",
e.message);
            dbus_error_free(&e);
        }
    }

    *n_fds = 0;

    /* We don't free the array here, in case we can recycle it later */
}
#endif

static void
free_counter (void *element,
              void *data)
{
    DBusCounter *counter = element;
    DBusMessage *message = data;

    _dbus_counter_adjust_size (counter, - message->size_counter_delta);
#ifdef HAVE_UNIX_FD_PASSING

```

```

    _dbus_counter_adjust_unix_fd (counter, - message-
>unix_fd_counter_delta);
#endif

    _dbus_counter_notify (counter);
    _dbus_counter_unref (counter);
}

/**
 * Tries to cache a message, otherwise finalize it.
 *
 * @param message the message
 */
static void
dbus_message_cache_or_finalize (DBusMessage *message)
{
    dbus_bool_t was_cached;
    int i;

    _dbus_assert (_dbus_atomic_get (&message->refcount) == 0);

    /* This calls application code and has to be done first thing
     * without holding the lock
     */
    _dbus_data_slot_list_clear (&message->slot_list);

    _dbus_list_foreach (&message->counters,
                        free_counter, message);
    _dbus_list_clear (&message->counters);

#ifdef HAVE_UNIX_FD_PASSING
    close_unix_fds(message->unix_fds, &message->n_unix_fds);
#endif

    was_cached = FALSE;

    _DBUS_LOCK (message_cache);

    if (!message_cache_shutdown_registered)
    {
        _dbus_assert (message_cache_count == 0);

        if (!_dbus_register_shutdown_func (dbus_message_cache_shutdown,
NULL))
            goto out;

        i = 0;
        while (i < MAX_MESSAGE_CACHE_SIZE)
        {
            message_cache[i] = NULL;
            ++i;
        }
    }
}

```

```

    message_cache_shutdown_registered = TRUE;
}

_dbus_assert (message_cache_count >= 0);

if (!_dbus_enable_message_cache ())
    goto out;

if (( _dbus_string_get_length (&message->header.data) +
     _dbus_string_get_length (&message->body)) >
     MAX_MESSAGE_SIZE_TO_CACHE)
    goto out;

if (message_cache_count >= MAX_MESSAGE_CACHE_SIZE)
    goto out;

/* Find empty slot */
i = 0;
while (message_cache[i] != NULL)
    ++i;

_dbus_assert (i < MAX_MESSAGE_CACHE_SIZE);

_dbus_assert (message_cache[i] == NULL);
message_cache[i] = message;
message_cache_count += 1;
was_cached = TRUE;
#ifdef DBUS_DISABLE_CHECKS
    message->in_cache = TRUE;
#endif

out:
_dbus_assert (_dbus_atomic_get (&message->refcount) == 0);

_DBUS_UNLOCK (message_cache);

if (!was_cached)
    dbus_message_finalize (message);
}

#ifdef DBUS_DISABLE_CHECKS
static dbus_bool_t
_dbus_message_iter_check (DBusMessageRealIter *iter)
{
    char byte_order;

    if (iter == NULL)
    {
        _dbus_warn_check_failed ("dbus message iterator is NULL\n");
        return FALSE;
    }
}

```

```

byte_order = _dbus_header_get_byte_order (&iter->message->header);

if (iter->iter_type == DBUS_MESSAGE_ITER_TYPE_READER)
{
    if (iter->u.reader.byte_order != byte_order)
    {
        _dbus_warn_check_failed ("dbus message changed byte order
since iterator was created\n");
        return FALSE;
    }
    /* because we swap the message into compiler order when you init
an iter */
    _dbus_assert (iter->u.reader.byte_order ==
DBUS_COMPILER_BYTE_ORDER);
}
else if (iter->iter_type == DBUS_MESSAGE_ITER_TYPE_WRITER)
{
    if (iter->u.writer.byte_order != byte_order)
    {
        _dbus_warn_check_failed ("dbus message changed byte order
since append iterator was created\n");
        return FALSE;
    }
    /* because we swap the message into compiler order when you init
an iter */
    _dbus_assert (iter->u.writer.byte_order ==
DBUS_COMPILER_BYTE_ORDER);
}
else
{
    _dbus_warn_check_failed ("dbus message iterator looks
uninitialized or corrupted\n");
    return FALSE;
}

if (iter->changed_stamp != iter->message->changed_stamp)
{
    _dbus_warn_check_failed ("dbus message iterator invalid because
the message has been modified (or perhaps the iterator is just
uninitialized)\n");
    return FALSE;
}

return TRUE;
}
#endif /* DBUS_DISABLE_CHECKS */

/**
 * Implementation of the varargs arg-getting functions.
 * dbus_message_get_args() is the place to go for complete
 * documentation.

```

```

*
* @todo This may leak memory and file descriptors if parsing fails.
See #21259
*
* @see dbus_message_get_args
* @param iter the message iter
* @param error error to be filled in
* @param first_arg_type type of the first argument
* @param var_args return location for first argument, followed by
list of type/location pairs
* @returns #FALSE if error was set
*/
dbus_bool_t
_dbus_message_iter_get_args_valist (DBusMessageIter *iter,
                                   DBusError *error,
                                   int first_arg_type,
                                   va_list var_args)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    int spec_type, msg_type, i;
    dbus_bool_t retval;

    _dbus_assert (_dbus_message_iter_check (real));

    retval = FALSE;

    spec_type = first_arg_type;
    i = 0;

    while (spec_type != DBUS_TYPE_INVALID)
    {
        msg_type = dbus_message_iter_get_arg_type (iter);

        if (msg_type != spec_type)
        {
            dbus_set_error (error, DBUS_ERROR_INVALID_ARGS,
                           "Argument %d is specified to be of type
%s", but "
                           "is actually of type %s", i,
                           _dbus_type_to_string (spec_type),
                           _dbus_type_to_string (msg_type));

            goto out;
        }

        if (spec_type == DBUS_TYPE_UNIX_FD)
        {
#ifdef HAVE_UNIX_FD_PASSING
            DBusBasicValue idx;
            int *pfd, nfd;

            pfd = va_arg (var_args, int*);

```

```

    _dbus_assert(pfd);

    _dbus_type_reader_read_basic(&real->u.reader, &idx);

    if (idx.u32 >= real->message->n_unix_fds)
    {
        dbus_set_error (error, DBUS_ERROR_INCONSISTENT_MESSAGE,
index %i,"
attached.\n",
                        "Message refers to file descriptor at
                        "but has only %i descriptors
                        idx.u32,
                        real->message->n_unix_fds);
        goto out;
    }

    if ((nfd = _dbus_dup(real->message->unix_fds[idx.u32],
error)) < 0)
        goto out;

    *pfd = nfd;
#else
    dbus_set_error (error, DBUS_ERROR_NOT_SUPPORTED,
passing.\n");
    goto out;
#endif
}
else if (dbus_type_is_basic (spec_type))
{
    DBusBasicValue *ptr;

    ptr = va_arg (var_args, DBusBasicValue*);

    _dbus_assert (ptr != NULL);

    _dbus_type_reader_read_basic (&real->u.reader,
ptr);
}
else if (spec_type == DBUS_TYPE_ARRAY)
{
    int element_type;
    int spec_element_type;
    const DBusBasicValue **ptr;
    int *n_elements_p;
    DBusTypeReader array;

    spec_element_type = va_arg (var_args, int);
    element_type = _dbus_type_reader_get_element_type (&real-
>u.reader);

    if (spec_element_type != element_type)

```

```

        {
            dbus_set_error (error, DBUS_ERROR_INVALID_ARGS,
                "Argument %d is specified to be an array
of \"%s\", but "
                "is actually an array of \"%s\"\n",
                i,
                _dbus_type_to_string
(spec_element_type),
                _dbus_type_to_string (element_type));

            goto out;
        }

    if (dbus_type_is_fixed (spec_element_type) &&
        element_type != DBUS_TYPE_UNIX_FD)
    {
        ptr = va_arg (var_args, const DBusBasicValue**);
        n_elements_p = va_arg (var_args, int*);

        _dbus_assert (ptr != NULL);
        _dbus_assert (n_elements_p != NULL);

        _dbus_type_reader_recurse (&real->u.reader, &array);

        _dbus_type_reader_read_fixed_multi (&array,
            (void *) ptr,
n_elements_p);
    }
    else if (_DBUS_TYPE_IS_STRINGLIKE (spec_element_type))
    {
        char ***str_array_p;
        int n_elements;
        char **str_array;

        str_array_p = va_arg (var_args, char***);
        n_elements_p = va_arg (var_args, int*);

        _dbus_assert (str_array_p != NULL);
        _dbus_assert (n_elements_p != NULL);

        /* Count elements in the array */
        _dbus_type_reader_recurse (&real->u.reader, &array);

        n_elements = 0;
        while (_dbus_type_reader_get_current_type (&array) !=
DBUS_TYPE_INVALID)
        {
            ++n_elements;
            _dbus_type_reader_next (&array);
        }

        str_array = dbus_new0 (char*, n_elements + 1);
    }

```



```

if (str_array == NULL)
{
    _DBUS_SET_OOM (error);
    goto out;
}

/* Now go through and dup each string */
_dbus_type_reader_recurse (&real->u.reader, &array);

i = 0;
while (i < n_elements)
{
    const char *s;
    _dbus_type_reader_read_basic (&array,
                                  (void *) &s);

    str_array[i] = _dbus_strdup (s);
    if (str_array[i] == NULL)
    {
        dbus_free_string_array (str_array);
        _DBUS_SET_OOM (error);
        goto out;
    }

    ++i;

    if (!_dbus_type_reader_next (&array))
        _dbus_assert (i == n_elements);
}

_dbus_assert (_dbus_type_reader_get_current_type
(&array) == DBUS_TYPE_INVALID);
_dbus_assert (i == n_elements);
_dbus_assert (str_array[i] == NULL);

*str_array_p = str_array;
*n_elements_p = n_elements;
}
#endif DBUS_DISABLE_CHECKS
else
{
    _dbus_warn ("you can't read arrays of container types
(struct, variant, array) with %s for now\n",
                _DBUS_FUNCTION_NAME);
    goto out;
}
#endif
}
#endif DBUS_DISABLE_CHECKS
else
{

```

```

        _dbus_warn ("you can only read arrays and basic types with
%s for now\n",
                    _DBUS_FUNCTION_NAME);
        goto out;
    }
#endif

    spec_type = va_arg (var_args, int);
    if (!_dbus_type_reader_next (&real->u.reader) && spec_type !=
DBUS_TYPE_INVALID)
    {
        dbus_set_error (error, DBUS_ERROR_INVALID_ARGS,
                        "Message has only %d arguments, but more
were expected", i);
        goto out;
    }

    i++;
}

retval = TRUE;

out:

return retval;
}

/** @} */

/**
 * @defgroup DbusMessage DbusMessage
 * @ingroup Dbus
 * @brief Message to be sent or received over a #DBusConnection.
 *
 * A DbusMessage is the most basic unit of communication over a
 * DbusConnection. A DbusConnection represents a stream of messages
 * received from a remote application, and a stream of messages
 * sent to a remote application.
 *
 * A message has a message type, returned from
 * dbus_message_get_type(). This indicates whether the message is a
 * method call, a reply to a method call, a signal, or an error reply.
 *
 * A message has header fields such as the sender, destination, method
 * or signal name, and so forth. DbusMessage has accessor functions
for
 * these, such as dbus_message_get_member().
 *
 * Convenience functions dbus_message_is_method_call(),
dbus_message_is_signal(),
 * and dbus_message_is_error() check several header fields at once and
are

```

```

* slightly more efficient than checking the header fields with
individual
* accessor functions.
*
* Finally, a message has arguments. The number and types of arguments
* are in the message's signature header field (accessed with
* dbus_message_get_signature()). Simple argument values are usually
* retrieved with dbus_message_get_args() but more complex values such
* as structs may require the use of #DBusMessageIter.
*
* The D-Bus specification goes into some more detail about header
fields and
* message types.
*
* @{
*/

/**
* @typedef DBusMessage
*
* Opaque data type representing a message received from or to be
* sent to another application.
*/

/**
* Returns the serial of a message or 0 if none has been specified.
* The message's serial number is provided by the application sending
* the message and is used to identify replies to this message.
*
* All messages received on a connection will have a serial provided
* by the remote application.
*
* For messages you're sending, dbus_connection_send() will assign a
* serial and return it to you.
*
* @param message the message
* @returns the serial
*/
dbus_uint32_t
dbus_message_get_serial (DBusMessage *message)
{
    _dbus_return_val_if_fail (message != NULL, 0);

    return _dbus_header_get_serial (&message->header);
}

/**
* Sets the reply serial of a message (the serial of the message this
* is a reply to).
*
* @param message the message
* @param reply_serial the serial we're replying to

```

```

* @returns #FALSE if not enough memory
*/
dbus_bool_t
dbus_message_set_reply_serial (DBusMessage *message,
                              dbus_uint32_t reply_serial)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (reply_serial != 0, FALSE); /* 0 is invalid
*/

    return _dbus_header_set_field_basic (&message->header,
                                         DBUS_HEADER_FIELD_REPLY_SERIAL,
                                         DBUS_TYPE_UINT32,
                                         &reply_serial);
}

/**
 * Returns the serial that the message is a reply to or 0 if none.
 *
 * @param message the message
 * @returns the reply serial
 */
dbus_uint32_t
dbus_message_get_reply_serial (DBusMessage *message)
{
    dbus_uint32_t v_UINT32;

    _dbus_return_val_if_fail (message != NULL, 0);

    if (_dbus_header_get_field_basic (&message->header,
                                       DBUS_HEADER_FIELD_REPLY_SERIAL,
                                       DBUS_TYPE_UINT32,
                                       &v_UINT32))

        return v_UINT32;
    else
        return 0;
}

static void
dbus_message_finalize (DBusMessage *message)
{
    _dbus_assert (_dbus_atomic_get (&message->refcount) == 0);

    /* This calls application callbacks! */
    _dbus_data_slot_list_free (&message->slot_list);

    _dbus_list_foreach (&message->counters,
                       free_counter, message);
    _dbus_list_clear (&message->counters);

    _dbus_header_free (&message->header);
}

```

```

    _dbus_string_free (&message->body);

#ifdef HAVE_UNIX_FD_PASSING
    close_unix_fds(message->unix_fds, &message->n_unix_fds);
    dbus_free(message->unix_fds);
#endif

    _dbus_assert (_dbus_atomic_get (&message->refcount) == 0);

    dbus_free (message);
}

static DBusMessage*
dbus_message_new_empty_header (void)
{
    DBusMessage *message;
    dbus_bool_t from_cache;

    message = dbus_message_get_cached ();

    if (message != NULL)
    {
        from_cache = TRUE;
    }
    else
    {
        from_cache = FALSE;
        message = dbus_new0 (DBusMessage, 1);
        if (message == NULL)
            return NULL;
#ifdef DBUS_DISABLE_CHECKS
        message->generation = _dbus_current_generation;
#endif
#ifdef HAVE_UNIX_FD_PASSING
        message->unix_fds = NULL;
        message->n_unix_fds_allocated = 0;
#endif
    }

    _dbus_atomic_inc (&message->refcount);

    _dbus_message_trace_ref (message, 0, 1, "new_empty_header");

    message->locked = FALSE;
#ifdef DBUS_DISABLE_CHECKS
    message->in_cache = FALSE;
#endif
    message->counters = NULL;
    message->size_counter_delta = 0;
    message->changed_stamp = 0;
}

```

```

#ifdef HAVE_UNIX_FD_PASSING
    message->n_unix_fds = 0;
    message->n_unix_fds_allocated = 0;
    message->unix_fd_counter_delta = 0;
#endif

    if (!from_cache)
        _dbus_data_slot_list_init (&message->slot_list);

    if (from_cache)
    {
        _dbus_header_reinit (&message->header);
        _dbus_string_set_length (&message->body, 0);
    }
    else
    {
        if (!_dbus_header_init (&message->header))
        {
            dbus_free (message);
            return NULL;
        }

        if (!_dbus_string_init_preallocated (&message->body, 32))
        {
            _dbus_header_free (&message->header);
            dbus_free (message);
            return NULL;
        }
    }

    return message;
}

/**
 * Constructs a new message of the given message type.
 * Types include #DBUS_MESSAGE_TYPE_METHOD_CALL,
 * #DBUS_MESSAGE_TYPE_SIGNAL, and so forth.
 *
 * Usually you want to use dbus_message_new_method_call(),
 * dbus_message_new_method_return(), dbus_message_new_signal(),
 * or dbus_message_new_error() instead.
 *
 * @param message_type type of message
 * @returns new message or #NULL if no memory
 */
DBusMessage*
dbus_message_new (int message_type)
{
    DBusMessage *message;

    _dbus_return_val_if_fail (message_type != DBUS_MESSAGE_TYPE_INVALID,
    NULL);
}

```

```

message = dbus_message_new_empty_header ();
if (message == NULL)
    return NULL;

if (!_dbus_header_create (&message->header,
                        DBUS_COMPILER_BYTE_ORDER,
                        message_type,
                        NULL, NULL, NULL, NULL, NULL))
{
    dbus_message_unref (message);
    return NULL;
}

return message;
}

/**
 * Constructs a new message to invoke a method on a remote
 * object. Returns #NULL if memory can't be allocated for the
 * message. The destination may be #NULL in which case no destination
 * is set; this is appropriate when using D-Bus in a peer-to-peer
 * context (no message bus). The interface may be #NULL, which means
 * that if multiple methods with the given name exist it is undefined
 * which one will be invoked.
 *
 * The path and method names may not be #NULL.
 *
 * Destination, path, interface, and method name can't contain
 * any invalid characters (see the D-Bus specification).
 *
 * @param destination name that the message should be sent to or #NULL
 * @param path object path the message should be sent to
 * @param interface interface to invoke method on, or #NULL
 * @param method method to invoke
 *
 * @returns a new DBusMessage, free with dbus_message_unref()
 */
DBusMessage*
dbus_message_new_method_call (const char *destination,
                             const char *path,
                             const char *interface,
                             const char *method)
{
    DBusMessage *message;

    _dbus_return_val_if_fail (path != NULL, NULL);
    _dbus_return_val_if_fail (method != NULL, NULL);
    _dbus_return_val_if_fail (destination == NULL ||
                              _dbus_check_is_valid_bus_name
(destination), NULL);
    _dbus_return_val_if_fail (_dbus_check_is_valid_path (path), NULL);

```

```

    _dbus_return_val_if_fail (interface == NULL ||
                              _dbus_check_is_valid_interface
(interface), NULL);
    _dbus_return_val_if_fail (_dbus_check_is_valid_member (method),
NULL);

    message = dbus_message_new_empty_header ();
    if (message == NULL)
        return NULL;

    if (!_dbus_header_create (&message->header,
                              DBUS_COMPILER_BYTE_ORDER,
                              DBUS_MESSAGE_TYPE_METHOD_CALL,
destination, path, interface, method,
NULL))
    {
        dbus_message_unref (message);
        return NULL;
    }

    return message;
}

/**
 * Constructs a message that is a reply to a method call. Returns
 * #NULL if memory can't be allocated for the message.
 *
 * @param method_call the message being replied to
 * @returns a new DbusMessage, free with dbus_message_unref()
 */
DBusMessage*
dbus_message_new_method_return (DBusMessage *method_call)
{
    DbusMessage *message;
    const char *sender;

    _dbus_return_val_if_fail (method_call != NULL, NULL);

    sender = dbus_message_get_sender (method_call);

    /* sender is allowed to be null here in peer-to-peer case */

    message = dbus_message_new_empty_header ();
    if (message == NULL)
        return NULL;

    if (!_dbus_header_create (&message->header,
                              DBUS_COMPILER_BYTE_ORDER,
                              DBUS_MESSAGE_TYPE_METHOD_RETURN,
sender, NULL, NULL, NULL, NULL))
    {
        dbus_message_unref (message);

```



```

        DBUS_MESSAGE_TYPE_ERROR,
        sender, NULL, NULL, NULL, error_name))
    {
        dbus_message_unref (message);
        return NULL;
    }

    dbus_message_set_no_reply (message, TRUE);

    if (!dbus_message_set_reply_serial (message,
                                        dbus_message_get_serial
(reply_to)))
    {
        dbus_message_unref (message);
        return NULL;
    }

    if (error_message != NULL)
    {
        dbus_message_iter_init_append (message, &iter);
        if (!dbus_message_iter_append_basic (&iter,
                                            DBUS_TYPE_STRING,
                                            &error_message))
        {
            dbus_message_unref (message);
            return NULL;
        }
    }

    return message;
}

/**
 * Creates a new message that is an error reply to another message,
 * allowing
 * you to use printf formatting.
 *
 * See dbus_message_new_error() for details - this function is the
 * same
 * aside from the printf formatting.
 *
 * @todo add _DBUS_GNUC_PRINTF to this (requires moving
_DBUS_GNUC_PRINTF to
 * public header, see DBUS_DEPRECATED for an example)
 *
 * @param reply_to the original message
 * @param error_name the error name
 * @param error_format the error message format as with printf
 * @param ... format string arguments
 * @returns a new error message
 */
DBusMessage*

```

```

dbus_message_new_error_printf (DBusMessage *reply_to,
                               const char *error_name,
                               const char *error_format,
                               ...)
{
    va_list args;
    DBusString str;
    DBusMessage *message;

    _dbus_return_val_if_fail (reply_to != NULL, NULL);
    _dbus_return_val_if_fail (error_name != NULL, NULL);
    _dbus_return_val_if_fail (_dbus_check_is_valid_error_name
                              (error_name), NULL);

    if (!_dbus_string_init (&str))
        return NULL;

    va_start (args, error_format);

    if (_dbus_string_append_printf_valist (&str, error_format, args))
        message = dbus_message_new_error (reply_to, error_name,
                                           _dbus_string_get_const_data (&str));
    else
        message = NULL;

    _dbus_string_free (&str);

    va_end (args);

    return message;
}

/**
 * Creates a new message that is an exact replica of the message
 * specified, except that its refcount is set to 1, its message serial
 * is reset to 0, and if the original message was "locked" (in the
 * outgoing message queue and thus not modifiable) the new message
 * will not be locked.
 *
 * @todo This function can't be used in programs that try to recover
 * from OOM errors.
 *
 * @param message the message
 * @returns the new message.or #NULL if not enough memory or Unix file
 * descriptors (in case the message to copy includes Unix file
 * descriptors) can be allocated.
 */
DBusMessage *
dbus_message_copy (const DBusMessage *message)
{
    DBusMessage *retval;

```

```

_dbus_return_val_if_fail (message != NULL, NULL);

retval = dbus_new0 (DBusMessage, 1);
if (retval == NULL)
    return NULL;

_dbus_atomic_inc (&retval->refcount);

retval->locked = FALSE;
#ifdef DBUS_DISABLE_CHECKS
    retval->generation = message->generation;
#endif

if (!_dbus_header_copy (&message->header, &retval->header))
{
    dbus_free (retval);
    return NULL;
}

if (!_dbus_string_init_preallocated (&retval->body,
                                   _dbus_string_get_length
(&message->body))
{
    _dbus_header_free (&retval->header);
    dbus_free (retval);
    return NULL;
}

if (!_dbus_string_copy (&message->body, 0,
                       &retval->body, 0))
    goto failed_copy;

#ifdef HAVE_UNIX_FD_PASSING
    retval->unix_fds = dbus_new(int, message->n_unix_fds);
    if (retval->unix_fds == NULL && message->n_unix_fds > 0)
        goto failed_copy;

    retval->n_unix_fds_allocated = message->n_unix_fds;

    for (retval->n_unix_fds = 0;
         retval->n_unix_fds < message->n_unix_fds;
         retval->n_unix_fds++)
    {
        retval->unix_fds[retval->n_unix_fds] = _dbus_dup(message-
>unix_fds[retval->n_unix_fds], NULL);

        if (retval->unix_fds[retval->n_unix_fds] < 0)
            goto failed_copy;
    }
#endif

#endif

```

```

    _dbus_message_trace_ref (retval, 0, 1, "copy");
    return retval;

failed_copy:
    _dbus_header_free (&retval->header);
    _dbus_string_free (&retval->body);

#ifdef HAVE_UNIX_FD_PASSING
    close_unix_fds(retval->unix_fds, &retval->n_unix_fds);
    dbus_free(retval->unix_fds);
#endif

    dbus_free (retval);

    return NULL;
}

/**
 * Increments the reference count of a DBusMessage.
 *
 * @param message the message
 * @returns the message
 * @see dbus_message_unref
 */
DBusMessage *
dbus_message_ref (DBusMessage *message)
{
    dbus_int32_t old_refcount;

    _dbus_return_val_if_fail (message != NULL, NULL);
    _dbus_return_val_if_fail (message->generation ==
        _dbus_current_generation, NULL);
    _dbus_return_val_if_fail (!message->in_cache, NULL);

    old_refcount = _dbus_atomic_inc (&message->refcount);
    _dbus_assert (old_refcount >= 1);
    _dbus_message_trace_ref (message, old_refcount, old_refcount + 1,
        "ref");

    return message;
}

/**
 * Decrements the reference count of a DBusMessage, freeing the
 * message if the count reaches 0.
 *
 * @param message the message
 * @see dbus_message_ref
 */
void

```

```

dbus_message_unref (DBusMessage *message)
{
    dbus_int32_t old_refcount;

    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (message->generation ==
_dbus_current_generation);
    _dbus_return_if_fail (!message->in_cache);

    old_refcount = _dbus_atomic_dec (&message->refcount);

    _dbus_assert (old_refcount >= 1);

    _dbus_message_trace_ref (message, old_refcount, old_refcount - 1,
"unref");

    if (old_refcount == 1)
    {
        /* Calls application callbacks! */
        dbus_message_cache_or_finalize (message);
    }
}

/**
 * Gets the type of a message. Types include
 * #DBUS_MESSAGE_TYPE_METHOD_CALL, #DBUS_MESSAGE_TYPE_METHOD_RETURN,
 * #DBUS_MESSAGE_TYPE_ERROR, #DBUS_MESSAGE_TYPE_SIGNAL, but other
 * types are allowed and all code must silently ignore messages of
 * unknown type. #DBUS_MESSAGE_TYPE_INVALID will never be returned.
 *
 * @param message the message
 * @returns the type of the message
 */
int
dbus_message_get_type (DBusMessage *message)
{
    _dbus_return_val_if_fail (message != NULL,
DBUS_MESSAGE_TYPE_INVALID);

    return _dbus_header_get_message_type (&message->header);
}

/**
 * Appends fields to a message given a variable argument list. The
 * variable argument list should contain the type of each argument
 * followed by the value to append. Appendable types are basic types,
 * and arrays of fixed-length basic types (except arrays of Unix file
 * descriptors). To append variable-length basic types, or any more
 * complex value, you have to use an iterator rather than this
 * function.
 *
 * To append a basic type, specify its type code followed by the

```

```

* address of the value. For example:
*
* @code
*
* dbus_int32_t v_INT32 = 42;
* const char *v_STRING = "Hello World";
* dbus_message_append_args (message,
*                             DBUS_TYPE_INT32, &v_INT32,
*                             DBUS_TYPE_STRING, &v_STRING,
*                             DBUS_TYPE_INVALID);
* @endcode
*
* To append an array of fixed-length basic types (except Unix file
* descriptors), pass in the DBUS_TYPE_ARRAY typecode, the element
* typecode, the address of the array pointer, and a 32-bit integer
* giving the number of elements in the array. So for example: @code
* const dbus_int32_t array[] = { 1, 2, 3 }; const dbus_int32_t
* *v_ARRAY = array; dbus_message_append_args (message,
* DBUS_TYPE_ARRAY, DBUS_TYPE_INT32, &v_ARRAY, 3, DBUS_TYPE_INVALID);
* @endcode
*
* This function does not support arrays of Unix file descriptors. If
* you need those you need to manually recurse into the array.
*
* For Unix file descriptors this function will internally duplicate
* the descriptor you passed in. Hence you may close the descriptor
* immediately after this call.
*
* @warning in C, given "int array[]", "&array == array" (the
* comp.lang.c FAQ says otherwise, but gcc and the FAQ don't agree).
* So if you're using an array instead of a pointer you have to create
* a pointer variable, assign the array to it, then take the address
* of the pointer variable. For strings it works to write
* const char *array = "Hello" and then use &array though.
*
* The last argument to this function must be #DBUS_TYPE_INVALID,
* marking the end of the argument list. If you don't do this
* then libdbus won't know to stop and will read invalid memory.
*
* String/signature/path arrays should be passed in as "const char***
* address_of_array" and "int n_elements"
*
* @todo support DBUS_TYPE_STRUCT and DBUS_TYPE_VARIANT and complex
arrays
*
* @todo If this fails due to lack of memory, the message is hosed and
* you have to start over building the whole message.
*
* @param message the message
* @param first_arg_type type of the first argument
* @param ... value of first argument, list of additional type-value
pairs

```



```

    * @returns #TRUE on success
    */
dbus_bool_t
dbus_message_append_args (DBusMessage *message,
                          int          first_arg_type,
                          ...)
{
    dbus_bool_t retval;
    va_list var_args;

    _dbus_return_val_if_fail (message != NULL, FALSE);

    va_start (var_args, first_arg_type);
    retval = dbus_message_append_args_valist (message,
                                              first_arg_type,
                                              var_args);

    va_end (var_args);

    return retval;
}

/**
 * Like dbus_message_append_args() but takes a va_list for use by
 * language bindings.
 *
 * * @todo for now, if this function fails due to OOM it will leave
 * * the message half-written and you have to discard the message
 * * and start over.
 *
 * * @see dbus_message_append_args.
 * * @param message the message
 * * @param first_arg_type type of first argument
 * * @param var_args value of first argument, then list of type/value
 * pairs
 * * @returns #TRUE on success
 */
dbus_bool_t
dbus_message_append_args_valist (DBusMessage *message,
                                int          first_arg_type,
                                va_list     var_args)
{
    int type;
    DBusMessageIter iter;

    _dbus_return_val_if_fail (message != NULL, FALSE);

    type = first_arg_type;

    dbus_message_iter_init_append (message, &iter);

    while (type != DBUS_TYPE_INVALID)
    {

```

```

if (dbus_type_is_basic (type))
{
    const DBusBasicValue *value;
    value = va_arg (var_args, const DBusBasicValue*);

    if (!dbus_message_iter_append_basic (&iter,
                                        type,
                                        value))

        goto failed;
}
else if (type == DBUS_TYPE_ARRAY)
{
    int element_type;
    DBusMessageIter array;
    char buf[2];

    element_type = va_arg (var_args, int);

    buf[0] = element_type;
    buf[1] = '\\0';
    if (!dbus_message_iter_open_container (&iter,
                                          DBUS_TYPE_ARRAY,
                                          buf,
                                          &array))

        goto failed;

    if (dbus_type_is_fixed (element_type) &&
        element_type != DBUS_TYPE_UNIX_FD)
    {
        const DBusBasicValue **value;
        int n_elements;

        value = va_arg (var_args, const DBusBasicValue**);
        n_elements = va_arg (var_args, int);

        if (!dbus_message_iter_append_fixed_array (&array,
                                                  element_type,
                                                  value,
                                                  n_elements))

            dbus_message_iter_abandon_container (&iter, &array);
        goto failed;
    }
}
else if (_DBUS_TYPE_IS_STRINGLIKE (element_type))
{
    const char ***value_p;
    const char **value;
    int n_elements;
    int i;

    value_p = va_arg (var_args, const char***);

```

```

        n_elements = va_arg (var_args, int);

        value = *value_p;

        i = 0;
        while (i < n_elements)
        {
            if (!dbus_message_iter_append_basic (&array,
                                                element_type,
                                                &value[i])) {
                dbus_message_iter_abandon_container (&iter,
&array);
                goto failed;
            }
            ++i;
        }
    else
    {
        _dbus_warn ("arrays of %s can't be appended with %s for
now\n",
                    _dbus_type_to_string (element_type),
                    _DBUS_FUNCTION_NAME);
        goto failed;
    }

    if (!dbus_message_iter_close_container (&iter, &array))
        goto failed;
}
#ifdef DBUS_DISABLE_CHECKS
else
{
    _dbus_warn ("type %s isn't supported yet in %s\n",
                _dbus_type_to_string (type),
                _DBUS_FUNCTION_NAME);
    goto failed;
}
#endif

    type = va_arg (var_args, int);
}

return TRUE;

failed:
return FALSE;
}

/**
 * Gets arguments from a message given a variable argument list. The
 * supported types include those supported by
 * dbus_message_append_args(); that is, basic types and arrays of

```

```

* fixed-length basic types. The arguments are the same as they would
* be for dbus_message_iter_get_basic() or
* dbus_message_iter_get_fixed_array().
*
* In addition to those types, arrays of string, object path, and
* signature are supported; but these are returned as allocated memory
* and must be freed with dbus_free_string_array(), while the other
* types are returned as const references. To get a string array
* pass in "char ***array_location" and "int *n_elements".
*
* Similar to dbus_message_get_fixed_array() this function does not
* support arrays of type DBUS_TYPE_UNIX_FD. If you need to parse
* messages with arrays of Unix file descriptors you need to recurse
* into the array manually.
*
* Unix file descriptors that are read with this function will have
* the FD_CLOEXEC flag set. If you need them without this flag set,
* make sure to unset it with fcntl().
*
* The variable argument list should contain the type of the argument
* followed by a pointer to where the value should be stored. The list
* is terminated with #DBUS_TYPE_INVALID.
*
* Except for string arrays, the returned values are constant; do not
* free them. They point into the #DBusMessage.
*
* If the requested arguments are not present, or do not have the
* requested types, then an error will be set.
*
* If more arguments than requested are present, the requested
* arguments are returned and the extra arguments are ignored.
*
* @todo support DBUS_TYPE_STRUCT and DBUS_TYPE_VARIANT and complex
arrays
*
* @param message the message
* @param error error to be filled in on failure
* @param first_arg_type the first argument type
* @param ... location for first argument value, then list of type-
location pairs
* @returns #FALSE if the error was set
*/
dbus_bool_t
dbus_message_get_args (DBusMessage      *message,
                      DBusError        *error,
                      int               first_arg_type,
                      ...)
{
    dbus_bool_t retval;
    va_list var_args;

    _dbus_return_val_if_fail (message != NULL, FALSE);

```

```

    _dbus_return_val_if_error_is_set (error, FALSE);

    va_start (var_args, first_arg_type);
    retval = dbus_message_get_args_valist (message, error,
first_arg_type, var_args);
    va_end (var_args);

    return retval;
}

/**
 * Like dbus_message_get_args but takes a va_list for use by language
bindings.
 *
 * @see dbus_message_get_args
 * @param message the message
 * @param error error to be filled in
 * @param first_arg_type type of the first argument
 * @param var_args return location for first argument, followed by
list of type/location pairs
 * @returns #FALSE if error was set
 */
dbus_bool_t
dbus_message_get_args_valist (DBusMessage      *message,
                             DBusError        *error,
                             int               first_arg_type,
                             va_list          var_args)
{
    DBusMessageIter iter;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_error_is_set (error, FALSE);

    dbus_message_iter_init (message, &iter);
    return _dbus_message_iter_get_args_valist (&iter, error,
first_arg_type, var_args);
}

static void
_dbus_message_iter_init_common (DBusMessage      *message,
                               DBusMessageRealIter *real,
                               int               iter_type)
{
    _dbus_assert (sizeof (DBusMessageRealIter) <= sizeof
(DBusMessageIter));

    /* Since the iterator will read or write who-knows-what from the
 * message, we need to get in the right byte order
 */
    ensure_byte_order (message);

    real->message = message;

```

```

    real->changed_stamp = message->changed_stamp;
    real->iter_type = iter_type;
    real->sig_refcount = 0;
}

/**
 * Initializes a #DBusMessageIter for reading the arguments of the
 * message passed in.
 *
 * When possible, dbus_message_get_args() is much more convenient.
 * Some types of argument can only be read with #DBusMessageIter
 * however.
 *
 * The easiest way to iterate is like this:
 * @code
 * dbus_message_iter_init (message, &iter);
 * while ((current_type = dbus_message_iter_get_arg_type (&iter)) !=
DBUS_TYPE_INVALID)
 *     dbus_message_iter_next (&iter);
 * @endcode
 *
 * #DBusMessageIter contains no allocated memory; it need not be
 * freed, and can be copied by assignment or memcpy().
 *
 * @param message the message
 * @param iter pointer to an iterator to initialize
 * @returns #FALSE if the message has no arguments
 */
dbus_bool_t
dbus_message_iter_init (DBusMessage      *message,
                       DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    const DBusString *type_str;
    int type_pos;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (iter != NULL, FALSE);

    get_const_signature (&message->header, &type_str, &type_pos);

    _dbus_message_iter_init_common (message, real,
                                   DBUS_MESSAGE_ITER_TYPE_READER);

    _dbus_type_reader_init (&real->u.reader,
                           _dbus_header_get_byte_order (&message-
>header),
                           type_str, type_pos,
                           &message->body,
                           0);
}

```

```

    return _dbus_type_reader_get_current_type (&real->u.reader) !=
    DBUS_TYPE_INVALID;
}

/**
 * Checks if an iterator has any more fields.
 *
 * @param iter the message iter
 * @returns #TRUE if there are more fields following
 */
dbus_bool_t
dbus_message_iter_has_next (DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real), FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
    DBUS_MESSAGE_ITER_TYPE_READER, FALSE);

    return _dbus_type_reader_has_next (&real->u.reader);
}

/**
 * Moves the iterator to the next field, if any. If there's no next
 * field, returns #FALSE. If the iterator moves forward, returns
 * #TRUE.
 *
 * @param iter the message iter
 * @returns #TRUE if the iterator was moved to the next field
 */
dbus_bool_t
dbus_message_iter_next (DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real), FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
    DBUS_MESSAGE_ITER_TYPE_READER, FALSE);

    return _dbus_type_reader_next (&real->u.reader);
}

/**
 * Returns the argument type of the argument that the message iterator
 * points to. If the iterator is at the end of the message, returns
 * #DBUS_TYPE_INVALID. You can thus write a loop as follows:
 *
 * @code
 * dbus_message_iter_init (message, &iter);
 * while ((current_type = dbus_message_iter_get_arg_type (&iter)) !=
    DBUS_TYPE_INVALID)
 *     dbus_message_iter_next (&iter);

```

```

* @endcode
*
* @param iter the message iter
* @returns the argument type
*/
int
dbus_message_iter_get_arg_type (DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real),
DBUS_TYPE_INVALID);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_READER, FALSE);

    return _dbus_type_reader_get_current_type (&real->u.reader);
}

/**
* Returns the element type of the array that the message iterator
* points to. Note that you need to check that the iterator points to
* an array prior to using this function.
*
* @param iter the message iter
* @returns the array element type
*/
int
dbus_message_iter_get_element_type (DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real),
DBUS_TYPE_INVALID);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_READER, DBUS_TYPE_INVALID);
    _dbus_return_val_if_fail (dbus_message_iter_get_arg_type (iter) ==
DBUS_TYPE_ARRAY, DBUS_TYPE_INVALID);

    return _dbus_type_reader_get_element_type (&real->u.reader);
}

/**
* Recurses into a container value when reading values from a message,
* initializing a sub-iterator to use for traversing the child values
* of the container.
*
* Note that this recurses into a value, not a type, so you can only
* recurse if the value exists. The main implication of this is that
* if you have for example an empty array of array of int32, you can
* recurse into the outermost array, but it will have no values, so
* you won't be able to recurse further. There's no array of int32 to
* recurse into.

```



```

*
* If a container is an array of fixed-length types (except Unix file
* descriptors), it is much more efficient to use
* dbus_message_iter_get_fixed_array() to get the whole array in one
* shot, rather than individually walking over the array elements.
*
* Be sure you have somehow checked that
* dbus_message_iter_get_arg_type() matches the type you are expecting
* to recurse into. Results of this function are undefined if there is
* no container to recurse into at the current iterator position.
*
* @param iter the message iterator
* @param sub the sub-iterator to initialize
*/
void
dbus_message_iter_recurse (DBusMessageIter *iter,
                          DBusMessageIter *sub)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    DBusMessageRealIter *real_sub = (DBusMessageRealIter *)sub;

    _dbus_return_if_fail (_dbus_message_iter_check (real));
    _dbus_return_if_fail (sub != NULL);

    *real_sub = *real;
    _dbus_type_reader_recurse (&real->u.reader, &real_sub->u.reader);
}

/**
 * Returns the current signature of a message iterator. This
 * is useful primarily for dealing with variants; one can
 * recurse into a variant and determine the signature of
 * the variant's value.
 *
 * The returned string must be freed with dbus_free().
 *
 * @param iter the message iterator
 * @returns the contained signature, or NULL if out of memory
 */
char *
dbus_message_iter_get_signature (DBusMessageIter *iter)
{
    const DBusString *sig;
    DBusString retstr;
    char *ret;
    int start, len;
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real), NULL);

    if (!_dbus_string_init (&retstr))
        return NULL;

```

```

    _dbus_type_reader_get_signature (&real->u.reader, &sig,
                                     &start, &len);
    if (!_dbus_string_append_len (&retstr,
                                 _dbus_string_get_const_data (sig) + start,
                                 len))
        return NULL;
    if (!_dbus_string_steal_data (&retstr, &ret))
        return NULL;
    _dbus_string_free (&retstr);
    return ret;
}

/**
 * Reads a basic-typed value from the message iterator.
 * Basic types are the non-containers such as integer and string.
 *
 * The value argument should be the address of a location to store
 * the returned value. So for int32 it should be a "dbus_int32_t*"
 * and for string a "const char**". The returned value is
 * by reference and should not be freed.
 *
 * This call duplicates Unix file descriptors when reading them. It is
 * your job to close them when you don't need them anymore.
 *
 * Unix file descriptors that are read with this function will have
 * the FD_CLOEXEC flag set. If you need them without this flag set,
 * make sure to unset it with fcntl().
 *
 * Be sure you have somehow checked that
 * dbus_message_iter_get_arg_type() matches the type you are
 * expecting, or you'll crash when you try to use an integer as a
 * string or something.
 *
 * To read any container type (array, struct, dict) you will need to
 * recurse into the container with dbus_message_iter_recurse(). If
 * the container is an array of fixed-length values (except Unix file
 * descriptors), you can get all the array elements at once with
 * dbus_message_iter_get_fixed_array(). Otherwise, you have to iterate
 * over the container's contents one value at a time.
 *
 * All basic-typed values are guaranteed to fit in a #DBusBasicValue,
 * so in versions of libdbus that have that type, you can write code
like this:
 *
 * @code
 * DBusBasicValue value;
 * int type;
 * dbus_message_iter_get_basic (&read_iter, &value);
 * type = dbus_message_iter_get_arg_type (&read_iter);
 * dbus_message_iter_append_basic (&write_iter, type, &value);
 * @endcode

```

```

*
* (All D-Bus basic types are either numeric and 8 bytes or smaller,
or
* behave like a string; so in older versions of libdbus,
DBusBasicValue
* can be replaced with union { char *string; unsigned char bytes[8];
},
* for instance.)
*
* @param iter the iterator
* @param value location to store the value
*/
void
dbus_message_iter_get_basic (DBusMessageIter *iter,
                             void *value)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_if_fail (_dbus_message_iter_check (real));
    _dbus_return_if_fail (value != NULL);

    if (dbus_message_iter_get_arg_type (iter) == DBUS_TYPE_UNIX_FD)
    {
#ifdef HAVE_UNIX_FD_PASSING
        DBusBasicValue idx;

        _dbus_type_reader_read_basic(&real->u.reader, &idx);

        if (idx.u32 >= real->message->n_unix_fds) {
            /* Hmm, we cannot really signal an error here, so let's make
               sure to return an invalid fd. */
            *((int*) value) = -1;
            return;
        }

        *((int*) value) = _dbus_dup(real->message->unix_fds[idx.u32],
NULL);
#else
        *((int*) value) = -1;
#endif
    }
    else
    {
        _dbus_type_reader_read_basic (&real->u.reader,
                                     value);
    }
}

/**
* Returns the number of bytes in the array as marshaled in the wire
* protocol. The iterator must currently be inside an array-typed
* value.

```

```

*
* This function is deprecated on the grounds that it is stupid. Why
* would you want to know how many bytes are in the array as marshaled
* in the wire protocol? For now, use the n_elements returned from
* dbus_message_iter_get_fixed_array() instead, or iterate over the
* array values and count them.
*
* @todo introduce a variant of this get_n_elements that returns
* the number of elements, though with a non-fixed array it will not
* be very efficient, so maybe it's not good.
*
* @param iter the iterator
* @returns the number of bytes in the array
*/
int
dbus_message_iter_get_array_len (DBusMessageIter *iter)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

    _dbus_return_val_if_fail (_dbus_message_iter_check (real), 0);

    return _dbus_type_reader_get_array_length (&real->u.reader);
}

/**
* Reads a block of fixed-length values from the message iterator.
* Fixed-length values are those basic types that are not string-like,
* such as integers, bool, double. The returned block will be from the
* current position in the array until the end of the array.
*
* There is one exception here: although DBUS_TYPE_UNIX_FD is
* considered a 'fixed' type arrays of this type may not be read with
* this function.
*
* The message iter should be "in" the array (that is, you recurse
into the
* array, and then you call dbus_message_iter_get_fixed_array() on the
* "sub-iterator" created by dbus_message_iter_recurse()).
*
* The value argument should be the address of a location to store the
* returned array. So for int32 it should be a "const dbus_int32_t**"
* The returned value is by reference and should not be freed.
*
* This function should only be used if dbus_type_is_fixed() returns
* #TRUE for the element type.
*
* If an array's elements are not fixed in size, you have to recurse
* into the array with dbus_message_iter_recurse() and read the
* elements one by one.
*
* Because the array is not copied, this function runs in constant
* time and is fast; it's much preferred over walking the entire array

```



```

    /* We create the signature string and point iterators at it "on
demand"
    * when a value is actually appended. That means that init() never
fails
    * due to OOM.
    */
    _dbus_type_writer_init_types_delayed (&real->u.writer,
                                         _dbus_header_get_byte_order
(&message->header),
                                         &message->body,
                                         _dbus_string_get_length
(&message->body));
}

/**
 * Creates a temporary signature string containing the current
 * signature, stores it in the iterator, and points the iterator to
 * the end of it. Used any time we write to the message.
 *
 * @param real an iterator without a type_str
 * @returns #FALSE if no memory
 */
static dbus_bool_t
_dbus_message_iter_open_signature (DBusMessageRealIter *real)
{
    DBusString *str;
    const DBusString *current_sig;
    int current_sig_pos;

    _dbus_assert (real->iter_type == DBUS_MESSAGE_ITER_TYPE_WRITER);

    if (real->u.writer.type_str != NULL)
    {
        _dbus_assert (real->sig_refcount > 0);
        real->sig_refcount += 1;
        return TRUE;
    }

    str = dbus_new (DBusString, 1);
    if (str == NULL)
        return FALSE;

    if (!_dbus_header_get_field_raw (&real->message->header,
                                    DBUS_HEADER_FIELD_SIGNATURE,
                                    &current_sig, &current_sig_pos))
        current_sig = NULL;

    if (current_sig)
    {
        int current_len;

```

```

        current_len = _dbus_string_get_byte (current_sig,
current_sig_pos);
        current_sig_pos += 1; /* move on to sig data */

        if (!_dbus_string_init_preallocated (str, current_len + 4))
        {
            dbus_free (str);
            return FALSE;
        }

        if (!_dbus_string_copy_len (current_sig, current_sig_pos,
current_len,
                                str, 0))
        {
            _dbus_string_free (str);
            dbus_free (str);
            return FALSE;
        }
    }
else
    {
        if (!_dbus_string_init_preallocated (str, 4))
        {
            dbus_free (str);
            return FALSE;
        }
    }

    real->sig_refcount = 1;

    _dbus_type_writer_add_types (&real->u.writer,
                                str, _dbus_string_get_length (str));
    return TRUE;
}

/**
 * Sets the new signature as the message signature, frees the
 * signature string, and marks the iterator as not having a type_str
 * anymore. Frees the signature even if it fails, so you can't
 * really recover from failure. Kinda busted.
 *
 * @param real an iterator without a type_str
 * @returns #FALSE if no memory
 */
static dbus_bool_t
_dbus_message_iter_close_signature (DBusMessageRealIter *real)
{
    DBusString *str;
    const char *v_STRING;
    dbus_bool_t retval;

    _dbus_assert (real->iter_type == DBUS_MESSAGE_ITER_TYPE_WRITER);

```

```

_dbus_assert (real->u.writer.type_str != NULL);
_dbus_assert (real->sig_refcount > 0);

real->sig_refcount -= 1;

if (real->sig_refcount > 0)
    return TRUE;
_dbus_assert (real->sig_refcount == 0);

retval = TRUE;

str = real->u.writer.type_str;

v_STRING = _dbus_string_get_const_data (str);
if (!_dbus_header_set_field_basic (&real->message->header,
                                  DBUS_HEADER_FIELD_SIGNATURE,
                                  DBUS_TYPE_SIGNATURE,
                                  &v_STRING))

    retval = FALSE;

_dbus_type_writer_remove_types (&real->u.writer);
_dbus_string_free (str);
dbus_free (str);

return retval;
}

/**
 * Frees the signature string and marks the iterator as not having a
 * type_str anymore.  Since the new signature is not set, the message
 * will generally be hosed after this is called.
 *
 * @param real an iterator without a type_str
 */
static void
_dbus_message_iter_abandon_signature (DBusMessageRealIter *real)
{
    DBusString *str;

    _dbus_assert (real->iter_type == DBUS_MESSAGE_ITER_TYPE_WRITER);
    _dbus_assert (real->u.writer.type_str != NULL);
    _dbus_assert (real->sig_refcount > 0);

    real->sig_refcount -= 1;

    if (real->sig_refcount > 0)
        return;
    _dbus_assert (real->sig_refcount == 0);

    str = real->u.writer.type_str;

    _dbus_type_writer_remove_types (&real->u.writer);

```



```

    _dbus_string_free (str);
    dbus_free (str);
}

#ifdef DBUS_DISABLE_CHECKS
static dbus_bool_t
_dbus_message_iter_append_check (DBusMessageRealIter *iter)
{
    if (!_dbus_message_iter_check (iter))
        return FALSE;

    if (iter->message->locked)
    {
        _dbus_warn_check_failed ("dbus append iterator can't be used:
message is locked (has already been sent)\n");
        return FALSE;
    }

    return TRUE;
}
#endif /* DBUS_DISABLE_CHECKS */

#ifdef HAVE_UNIX_FD_PASSING
static int *
expand_fd_array(DBusMessage *m,
                unsigned n)
{
    _dbus_assert(m);

    /* This makes space for adding n new fds to the array and returns a
    pointer to the place where the first fd should be put. */

    if (m->n_unix_fds + n > m->n_unix_fds_allocated)
    {
        unsigned k;
        int *p;

        /* Make twice as much space as necessary */
        k = (m->n_unix_fds + n) * 2;

        /* Allocate at least four */
        if (k < 4)
            k = 4;

        p = dbus_realloc(m->unix_fds, k * sizeof(int));
        if (p == NULL)
            return NULL;

        m->unix_fds = p;
        m->n_unix_fds_allocated = k;
    }
}

```

```

    return m->unix_fds + m->n_unix_fds;
}
#endif

/**
 * Appends a basic-typed value to the message. The basic types are the
 * non-container types such as integer and string.
 *
 * The "value" argument should be the address of a basic-typed value.
 * So for string, const char**. For integer, dbus_int32_t*.
 *
 * For Unix file descriptors this function will internally duplicate
 * the descriptor you passed in. Hence you may close the descriptor
 * immediately after this call.
 *
 * @todo If this fails due to lack of memory, the message is hosed and
 * you have to start over building the whole message.
 *
 * @param iter the append iterator
 * @param type the type of the value
 * @param value the address of the value
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_message_iter_append_basic (DBusMessageIter *iter,
                               int type,
                               const void *value)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    dbus_bool_t ret;

    _dbus_return_val_if_fail (_dbus_message_iter_append_check (real),
FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER, FALSE);
    _dbus_return_val_if_fail (dbus_type_is_basic (type), FALSE);
    _dbus_return_val_if_fail (value != NULL, FALSE);

#ifdef DBUS_DISABLE_CHECKS
    switch (type)
    {
        const char * const *string_p;
        const dbus_bool_t *bool_p;

        case DBUS_TYPE_STRING:
            string_p = value;
            _dbus_return_val_if_fail (!_dbus_check_is_valid_utf8
(*string_p), FALSE);
            break;

        case DBUS_TYPE_OBJECT_PATH:
            string_p = value;

```

```

        _dbus_return_val_if_fail (_dbus_check_is_valid_path
(*string_p), FALSE);
        break;

        case DBUS_TYPE_SIGNATURE:
            string_p = value;
            _dbus_return_val_if_fail (_dbus_check_is_valid_signature
(*string_p), FALSE);
            break;

        case DBUS_TYPE_BOOLEAN:
            bool_p = value;
            _dbus_return_val_if_fail (*bool_p == 0 || *bool_p == 1,
FALSE);
            break;

        default:
            {
                /* nothing to check, all possible values are allowed */
            }
    }
#endif

    if (!_dbus_message_iter_open_signature (real))
        return FALSE;

    if (type == DBUS_TYPE_UNIX_FD)
    {
#ifdef HAVE_UNIX_FD_PASSING
        int *fds;
        dbus_uint32_t u;

        /* First step, include the fd in the fd list of this message */
        if (!(fds = expand_fd_array(real->message, 1)))
            return FALSE;

        *fds = _dbus_dup(*(int*) value, NULL);
        if (*fds < 0)
            return FALSE;

        u = real->message->n_unix_fds;

        /* Second step, write the index to the fd */
        if (!(ret = _dbus_type_writer_write_basic (&real->u.writer,
DBUS_TYPE_UNIX_FD, &u))) {
            _dbus_close(*fds, NULL);
            return FALSE;
        }

        real->message->n_unix_fds += 1;
        u += 1;

```

```

/* Final step, update the header accordingly */
ret = _dbus_header_set_field_basic (&real->message->header,
                                   DBUS_HEADER_FIELD_UNIX_FDS,
                                   DBUS_TYPE_UINT32,
                                   &u);

/* If any of these operations fail the message is
   hosed. However, no memory or fds should be leaked since what
   has been added to message has been added to the message, and
   can hence be accounted for when the message is being
   freed. */
#else
    ret = FALSE;
#endif
    }
    else
    {
        ret = _dbus_type_writer_write_basic (&real->u.writer, type,
value);
    }

    if (!_dbus_message_iter_close_signature (real))
        ret = FALSE;

    return ret;
}

/**
 * Appends a block of fixed-length values to an array. The
 * fixed-length types are all basic types that are not string-like. So
 * int32, double, bool, etc. (Unix file descriptors however are not
 * supported.) You must call dbus_message_iter_open_container() to
 * open an array of values before calling this function. You may call
 * this function multiple times (and intermixed with calls to
 * dbus_message_iter_append_basic()) for the same array.
 *
 * The "value" argument should be the address of the array. So for
 * integer, "dbus_int32_t**" is expected for example.
 *
 * @warning in C, given "int array[]", "&array == array" (the
 * comp.lang.c FAQ says otherwise, but gcc and the FAQ don't agree).
 * So if you're using an array instead of a pointer you have to create
 * a pointer variable, assign the array to it, then take the address
 * of the pointer variable.
 * @code
 * const dbus_int32_t array[] = { 1, 2, 3 };
 * const dbus_int32_t *v_ARRAY = array;
 * if (!dbus_message_iter_append_fixed_array (&iter, DBUS_TYPE_INT32,
&v_ARRAY, 3))
 *     fprintf (stderr, "No memory!\n");
 * @endcode
 * For strings it works to write const char *array = "Hello" and then

```

```

* use &array though.
*
* @todo If this fails due to lack of memory, the message is hosed and
* you have to start over building the whole message.
*
* @param iter the append iterator
* @param element_type the type of the array elements
* @param value the address of the array
* @param n_elements the number of elements to append
* @returns #FALSE if not enough memory
*/
dbus_bool_t
dbus_message_iter_append_fixed_array (DBusMessageIter *iter,
                                     int element_type,
                                     const void *value,
                                     int n_elements)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    dbus_bool_t ret;

    _dbus_return_val_if_fail (_dbus_message_iter_append_check (real),
FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER, FALSE);
    _dbus_return_val_if_fail (dbus_type_is_fixed (element_type) &&
element_type != DBUS_TYPE_UNIX_FD, FALSE);
    _dbus_return_val_if_fail (real->u.writer.container_type ==
DBUS_TYPE_ARRAY, FALSE);
    _dbus_return_val_if_fail (value != NULL, FALSE);
    _dbus_return_val_if_fail (n_elements >= 0, FALSE);
    _dbus_return_val_if_fail (n_elements <=
DBUS_MAXIMUM_ARRAY_LENGTH /
_dbus_type_get_alignment (element_type),
FALSE);

#ifdef DBUS_DISABLE_CHECKS
    if (element_type == DBUS_TYPE_BOOLEAN)
    {
        const dbus_bool_t * const *bools = value;
        int i;

        for (i = 0; i < n_elements; i++)
        {
            _dbus_return_val_if_fail ((*bools)[i] == 0 || (*bools)[i] ==
1, FALSE);
        }
    }
#endif

    ret = _dbus_type_writer_write_fixed_multi (&real->u.writer,
element_type, value, n_elements);

```

```

    return ret;
}

/**
 * Appends a container-typed value to the message; you are required to
 * append the contents of the container using the returned
 * sub-iterator, and then call
 * dbus_message_iter_close_container(). Container types are for
 * example struct, variant, and array. For variants, the
 * contained_signature should be the type of the single value inside
 * the variant. For structs and dict entries, contained_signature
 * should be #NULL; it will be set to whatever types you write into
 * the struct. For arrays, contained_signature should be the type of
 * the array elements.
 *
 * @todo If this fails due to lack of memory, the message is hosed and
 * you have to start over building the whole message.
 *
 * @param iter the append iterator
 * @param type the type of the value
 * @param contained_signature the type of container contents
 * @param sub sub-iterator to initialize
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_message_iter_open_container (DBusMessageIter *iter,
                                int type,
                                const char
*contained_signature,
                                DBusMessageIter *sub)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
    DBusMessageRealIter *real_sub = (DBusMessageRealIter *)sub;
    DBusString contained_str;

    _dbus_return_val_if_fail (_dbus_message_iter_append_check (real),
FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER, FALSE);
    _dbus_return_val_if_fail (dbus_type_is_container (type), FALSE);
    _dbus_return_val_if_fail (sub != NULL, FALSE);
    _dbus_return_val_if_fail ((type == DBUS_TYPE_STRUCT &&
contained_signature == NULL) ||
(type == DBUS_TYPE_DICT_ENTRY &&
contained_signature == NULL) ||
(type == DBUS_TYPE_VARIANT &&
contained_signature != NULL) ||
(type == DBUS_TYPE_ARRAY &&
contained_signature != NULL), FALSE);

    /* this would fail if the contained_signature is a dict entry, since
 * dict entries are invalid signatures standalone (they must be in

```

```

    * an array)
    */
    _dbus_return_val_if_fail ((type == DBUS_TYPE_ARRAY &&
contained_signature && *contained_signature ==
DBUS_DICT_ENTRY_BEGIN_CHAR) ||
        (contained_signature == NULL ||
        _dbus_check_is_valid_signature
(contained_signature)),
        FALSE);

    if (!_dbus_message_iter_open_signature (real))
        return FALSE;

    *real_sub = *real;

    if (contained_signature != NULL)
    {
        _dbus_string_init_const (&contained_str, contained_signature);

        return _dbus_type_writer_recurse (&real->u.writer,
            type,
            &contained_str, 0,
            &real_sub->u.writer);
    }
    else
    {
        return _dbus_type_writer_recurse (&real->u.writer,
            type,
            NULL, 0,
            &real_sub->u.writer);
    }
}

/**
 * Closes a container-typed value appended to the message; may write
 * out more information to the message known only after the entire
 * container is written, and may free resources created by
 * dbus_message_iter_open_container().
 *
 * @todo If this fails due to lack of memory, the message is hosed and
 * you have to start over building the whole message.
 *
 * @param iter the append iterator
 * @param sub sub-iterator to close
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_message_iter_close_container (DBusMessageIter *iter,
        DBusMessageIter *sub)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;

```

```

DBusMessageRealIter *real_sub = (DBusMessageRealIter *)sub;
dbus_bool_t ret;

    _dbus_return_val_if_fail (_dbus_message_iter_append_check (real),
FALSE);
    _dbus_return_val_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER, FALSE);
    _dbus_return_val_if_fail (_dbus_message_iter_append_check
(real_sub), FALSE);
    _dbus_return_val_if_fail (real_sub->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER, FALSE);

    ret = _dbus_type_writer_unrecurse (&real->u.writer,
&real_sub->u.writer);

    if (!_dbus_message_iter_close_signature (real))
        ret = FALSE;

    return ret;
}

/**
 * Abandons creation of a contained-typed value and frees resources
created
 * by dbus_message_iter_open_container(). Once this returns, the
message
 * is hosed and you have to start over building the whole message.
 *
 * This should only be used to abandon creation of a message when you
have
 * open containers.
 *
 * @param iter the append iterator
 * @param sub sub-iterator to close
 */
void
dbus_message_iter_abandon_container (DBusMessageIter *iter,
DBusMessageIter *sub)
{
    DBusMessageRealIter *real = (DBusMessageRealIter *)iter;
#ifdef DBUS_DISABLE_CHECKS
    DBusMessageRealIter *real_sub = (DBusMessageRealIter *)sub;

    _dbus_return_if_fail (_dbus_message_iter_append_check (real));
    _dbus_return_if_fail (real->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER);
    _dbus_return_if_fail (_dbus_message_iter_append_check (real_sub));
    _dbus_return_if_fail (real_sub->iter_type ==
DBUS_MESSAGE_ITER_TYPE_WRITER);
#endif

    _dbus_message_iter_abandon_signature (real);

```



```

}

/**
 * Sets a flag indicating that the message does not want a reply; if
 * this flag is set, the other end of the connection may (but is not
 * required to) optimize by not sending method return or error
 * replies. If this flag is set, there is no way to know whether the
 * message successfully arrived at the remote end. Normally you know a
 * message was received when you receive the reply to it.
 *
 * The flag is #FALSE by default, that is by default the other end is
 * required to reply.
 *
 * On the protocol level this toggles
 #DBUS_HEADER_FLAG_NO_REPLY_EXPECTED
 *
 * @param message the message
 * @param no_reply #TRUE if no reply is desired
 */
void
dbus_message_set_no_reply (DBusMessage *message,
                          dbus_bool_t no_reply)
{
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (!message->locked);

    _dbus_header_toggle_flag (&message->header,
                              DBUS_HEADER_FLAG_NO_REPLY_EXPECTED,
                              no_reply);
}

/**
 * Returns #TRUE if the message does not expect
 * a reply.
 *
 * @param message the message
 * @returns #TRUE if the message sender isn't waiting for a reply
 */
dbus_bool_t
dbus_message_get_no_reply (DBusMessage *message)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);

    return _dbus_header_get_flag (&message->header,
                                  DBUS_HEADER_FLAG_NO_REPLY_EXPECTED);
}

/**
 * Sets a flag indicating that an owner for the destination name will
 * be automatically started before the message is delivered. When this
 * flag is set, the message is held until a name owner finishes
 * starting up, or fails to start up. In case of failure, the reply

```

```

* will be an error.
*
* The flag is set to #TRUE by default, i.e. auto starting is the
default.
*
* On the protocol level this toggles #DBUS_HEADER_FLAG_NO_AUTO_START
*
* @param message the message
* @param auto_start #TRUE if auto-starting is desired
*/
void
dbus_message_set_auto_start (DBusMessage *message,
                             dbus_bool_t auto_start)
{
    _dbus_return_if_fail (message != NULL);
    _dbus_return_if_fail (!message->locked);

    _dbus_header_toggle_flag (&message->header,
                              DBUS_HEADER_FLAG_NO_AUTO_START,
                              !auto_start);
}

/**
 * Returns #TRUE if the message will cause an owner for
 * destination name to be auto-started.
 *
 * @param message the message
 * @returns #TRUE if the message will use auto-start
 */
dbus_bool_t
dbus_message_get_auto_start (DBusMessage *message)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);

    return !_dbus_header_get_flag (&message->header,
                                   DBUS_HEADER_FLAG_NO_AUTO_START);
}

/**
 * Sets the object path this message is being sent to (for
 * DBUS_MESSAGE_TYPE_METHOD_CALL) or the one a signal is being
 * emitted from (for DBUS_MESSAGE_TYPE_SIGNAL).
 *
 * The path must contain only valid characters as defined
 * in the D-Bus specification.
 *
 * @param message the message
 * @param object_path the path or #NULL to unset
 * @returns #FALSE if not enough memory
 */
dbus_bool_t

```

```

dbus_message_set_path (DBusMessage  *message,
                      const char    *object_path)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (object_path == NULL ||
                              _dbus_check_is_valid_path (object_path),
                              FALSE);

    return set_or_delete_string_field (message,
                                       DBUS_HEADER_FIELD_PATH,
                                       DBUS_TYPE_OBJECT_PATH,
                                       object_path);
}

/**
 * Gets the object path this message is being sent to (for
 * DBUS_MESSAGE_TYPE_METHOD_CALL) or being emitted from (for
 * DBUS_MESSAGE_TYPE_SIGNAL). Returns #NULL if none.
 *
 * See also dbus_message_get_path_decomposed().
 *
 * The returned string becomes invalid if the message is
 * modified, since it points into the wire-marshaled message data.
 *
 * @param message the message
 * @returns the path (should not be freed) or #NULL
 */
const char*
dbus_message_get_path (DBusMessage  *message)
{
    const char *v;

    _dbus_return_val_if_fail (message != NULL, NULL);

    v = NULL; /* in case field doesn't exist */
    _dbus_header_get_field_basic (&message->header,
                                  DBUS_HEADER_FIELD_PATH,
                                  DBUS_TYPE_OBJECT_PATH,
                                  (void *) &v);

    return v;
}

/**
 * Checks if the message has a particular object path. The object
 * path is the destination object for a method call or the emitting
 * object for a signal.
 *
 * @param message the message
 * @param path the path name
 * @returns #TRUE if there is a path field in the header
 */

```

```

dbus_bool_t
dbus_message_has_path (DBusMessage *message,
                      const char *path)
{
    const char *msg_path;
    msg_path = dbus_message_get_path (message);

    if (msg_path == NULL)
    {
        if (path == NULL)
            return TRUE;
        else
            return FALSE;
    }

    if (path == NULL)
        return FALSE;

    if (strcmp (msg_path, path) == 0)
        return TRUE;

    return FALSE;
}

/**
 * Gets the object path this message is being sent to
 * (for DBUS_MESSAGE_TYPE_METHOD_CALL) or being emitted
 * from (for DBUS_MESSAGE_TYPE_SIGNAL) in a decomposed
 * format (one array element per path component).
 * Free the returned array with dbus_free_string_array().
 *
 * An empty but non-NULL path array means the path "/".
 * So the path "/foo/bar" becomes { "foo", "bar", NULL }
 * and the path "/" becomes { NULL }.
 *
 * See also dbus_message_get_path().
 *
 * @todo this could be optimized by using the len from the message
 * instead of calling strlen() again
 *
 * @param message the message
 * @param path place to store allocated array of path components;
#NULL set here if no path field exists
 * @returns #FALSE if no memory to allocate the array
 */
dbus_bool_t
dbus_message_get_path_decomposed (DBusMessage *message,
                                 char ***path)
{
    const char *v;

    _dbus_return_val_if_fail (message != NULL, FALSE);
}

```

```

_dbus_return_val_if_fail (path != NULL, FALSE);

*path = NULL;

v = dbus_message_get_path (message);
if (v != NULL)
{
    if (!_dbus_decompose_path (v, strlen (v),
                              path, NULL))
        return FALSE;
}
return TRUE;
}

/**
 * Sets the interface this message is being sent to
 * (for DBUS_MESSAGE_TYPE_METHOD_CALL) or
 * the interface a signal is being emitted from
 * (for DBUS_MESSAGE_TYPE_SIGNAL).
 *
 * The interface name must contain only valid characters as defined
 * in the D-Bus specification.
 *
 * @param message the message
 * @param interface the interface or #NULL to unset
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_message_set_interface (DBusMessage *message,
                           const char *interface)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (interface == NULL ||
                              _dbus_check_is_valid_interface
(interface),
                              FALSE);

    return set_or_delete_string_field (message,
                                      DBUS_HEADER_FIELD_INTERFACE,
                                      DBUS_TYPE_STRING,
                                      interface);
}

/**
 * Gets the interface this message is being sent to
 * (for DBUS_MESSAGE_TYPE_METHOD_CALL) or being emitted
 * from (for DBUS_MESSAGE_TYPE_SIGNAL).
 * The interface name is fully-qualified (namespaced).
 * Returns #NULL if none.
 *
 * The returned string becomes invalid if the message is

```

```

* modified, since it points into the wire-marshaled message data.
*
* @param message the message
* @returns the message interface (should not be freed) or #NULL
*/
const char*
dbus_message_get_interface (DBusMessage *message)
{
    const char *v;

    _dbus_return_val_if_fail (message != NULL, NULL);

    v = NULL; /* in case field doesn't exist */
    _dbus_header_get_field_basic (&message->header,
                                  DBUS_HEADER_FIELD_INTERFACE,
                                  DBUS_TYPE_STRING,
                                  (void *) &v);

    return v;
}

/**
* Checks if the message has an interface
*
* @param message the message
* @param interface the interface name
* @returns #TRUE if the interface field in the header matches
*/
dbus_bool_t
dbus_message_has_interface (DBusMessage *message,
                            const char *interface)
{
    const char *msg_interface;
    msg_interface = dbus_message_get_interface (message);

    if (msg_interface == NULL)
    {
        if (interface == NULL)
            return TRUE;
        else
            return FALSE;
    }

    if (interface == NULL)
        return FALSE;

    if (strcmp (msg_interface, interface) == 0)
        return TRUE;

    return FALSE;
}

```



```

}

/**
 * Checks if the message has an interface member
 *
 * @param message the message
 * @param member the member name
 * @returns #TRUE if there is a member field in the header
 */
dbus_bool_t
dbus_message_has_member (DBusMessage *message,
                        const char *member)
{
    const char *msg_member;
    msg_member = dbus_message_get_member (message);

    if (msg_member == NULL)
    {
        if (member == NULL)
            return TRUE;
        else
            return FALSE;
    }

    if (member == NULL)
        return FALSE;

    if (strcmp (msg_member, member) == 0)
        return TRUE;

    return FALSE;
}

/**
 * Sets the name of the error (DBUS_MESSAGE_TYPE_ERROR).
 * The name is fully-qualified (namespaced).
 *
 * The error name must contain only valid characters as defined
 * in the D-Bus specification.
 *
 * @param message the message
 * @param error_name the name or #NULL to unset
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_message_set_error_name (DBusMessage *message,
                            const char *error_name)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (error_name == NULL ||

```



```

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (destination == NULL ||
        _dbus_check_is_valid_bus_name
(destination),
        FALSE);

    return set_or_delete_string_field (message,
        DBUS_HEADER_FIELD_DESTINATION,
        DBUS_TYPE_STRING,
        destination);
}

/**
 * Gets the destination of a message or #NULL if there is none set.
 *
 * The returned string becomes invalid if the message is
 * modified, since it points into the wire-marshaled message data.
 *
 * @param message the message
 * @returns the message destination (should not be freed) or #NULL
 */
const char*
dbus_message_get_destination (DBusMessage *message)
{
    const char *v;

    _dbus_return_val_if_fail (message != NULL, NULL);

    v = NULL; /* in case field doesn't exist */
    _dbus_header_get_field_basic (&message->header,
        DBUS_HEADER_FIELD_DESTINATION,
        DBUS_TYPE_STRING,
        (void *) &v);

    return v;
}

/**
 * Sets the message sender.
 *
 * The sender must be a valid bus name as defined in the D-Bus
 * specification.
 *
 * Usually you don't want to call this. The message bus daemon will
 * call it to set the origin of each message. If you aren't
implementing
 * a message bus daemon you shouldn't need to set the sender.
 *
 * @param message the message
 * @param sender the sender or #NULL to unset
 * @returns #FALSE if not enough memory
 */

```

```

dbus_bool_t
dbus_message_set_sender (DBusMessage *message,
                        const char *sender)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (!message->locked, FALSE);
    _dbus_return_val_if_fail (sender == NULL ||
                              _dbus_check_is_valid_bus_name (sender),
                              FALSE);

    return set_or_delete_string_field (message,
                                      DBUS_HEADER_FIELD_SENDER,
                                      DBUS_TYPE_STRING,
                                      sender);
}

/**
 * Gets the unique name of the connection which originated this
 * message, or #NULL if unknown or inapplicable. The sender is filled
 * in by the message bus.
 *
 * Note, the returned sender is always the unique bus name.
 * Connections may own multiple other bus names, but those
 * are not found in the sender field.
 *
 * The returned string becomes invalid if the message is
 * modified, since it points into the wire-marshaled message data.
 *
 * @param message the message
 * @returns the unique name of the sender or #NULL
 */
const char*
dbus_message_get_sender (DBusMessage *message)
{
    const char *v;

    _dbus_return_val_if_fail (message != NULL, NULL);

    v = NULL; /* in case field doesn't exist */
    _dbus_header_get_field_basic (&message->header,
                                  DBUS_HEADER_FIELD_SENDER,
                                  DBUS_TYPE_STRING,
                                  (void *) &v);

    return v;
}

/**
 * Gets the type signature of the message, i.e. the arguments in the
 * message payload. The signature includes only "in" arguments for
 * #DBUS_MESSAGE_TYPE_METHOD_CALL and only "out" arguments for
 * #DBUS_MESSAGE_TYPE_METHOD_RETURN, so is slightly different from

```

```

* what you might expect (that is, it does not include the signature
of the
* entire C++-style method).
*
* The signature is a string made up of type codes such as
* #DBUS_TYPE_INT32. The string is terminated with nul (nul is also
* the value of #DBUS_TYPE_INVALID).
*
* The returned string becomes invalid if the message is
* modified, since it points into the wire-marshaled message data.
*
* @param message the message
* @returns the type signature
*/
const char*
dbus_message_get_signature (DBusMessage *message)
{
    const DBusString *type_str;
    int type_pos;

    _dbus_return_val_if_fail (message != NULL, NULL);

    get_const_signature (&message->header, &type_str, &type_pos);

    return _dbus_string_get_const_data_len (type_str, type_pos, 0);
}

static dbus_bool_t
_dbus_message_has_type_interface_member (DBusMessage *message,
                                         int type,
                                         const char *interface,
                                         const char *member)
{
    const char *n;

    _dbus_assert (message != NULL);
    _dbus_assert (interface != NULL);
    _dbus_assert (member != NULL);

    if (dbus_message_get_type (message) != type)
        return FALSE;

    /* Optimize by checking the short member name first
     * instead of the longer interface name
     */

    n = dbus_message_get_member (message);

    if (n && strcmp (n, member) == 0)
        {
            n = dbus_message_get_interface (message);

```

```

        if (n == NULL || strcmp (n, interface) == 0)
            return TRUE;
    }

    return FALSE;
}

/**
 * Checks whether the message is a method call with the given
 * interface and member fields.  If the message is not
 * #DBUS_MESSAGE_TYPE_METHOD_CALL, or has a different interface or
 * member field, returns #FALSE.  If the interface field is missing,
 * then it will be assumed equal to the provided interface.  The D-Bus
 * protocol allows method callers to leave out the interface name.
 *
 * @param message the message
 * @param interface the name to check (must not be #NULL)
 * @param method the name to check (must not be #NULL)
 *
 * @returns #TRUE if the message is the specified method call
 */
dbus_bool_t
dbus_message_is_method_call (DBusMessage *message,
                             const char *interface,
                             const char *method)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (interface != NULL, FALSE);
    _dbus_return_val_if_fail (method != NULL, FALSE);
    /* don't check that interface/method are valid since it would be
     * expensive, and not catch many common errors
     */

    return _dbus_message_has_type_interface_member (message,
                                                    DBUS_MESSAGE_TYPE_METHOD_CALL,
                                                    interface, method);
}

/**
 * Checks whether the message is a signal with the given interface and
 * member fields.  If the message is not #DBUS_MESSAGE_TYPE_SIGNAL, or
 * has a different interface or member field, returns #FALSE.
 *
 * @param message the message
 * @param interface the name to check (must not be #NULL)
 * @param signal_name the name to check (must not be #NULL)
 *
 * @returns #TRUE if the message is the specified signal
 */
dbus_bool_t
dbus_message_is_signal (DBusMessage *message,

```

```

        const char *interface,
        const char *signal_name)
{
    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (interface != NULL, FALSE);
    _dbus_return_val_if_fail (signal_name != NULL, FALSE);
    /* don't check that interface/name are valid since it would be
     * expensive, and not catch many common errors
     */

    return _dbus_message_has_type_interface_member (message,
DBUS_MESSAGE_TYPE_SIGNAL,
                                                    interface,
signal_name);
}

/**
 * Checks whether the message is an error reply with the given error
 * name.  If the message is not #DBUS_MESSAGE_TYPE_ERROR, or has a
 * different name, returns #FALSE.
 *
 * @param message the message
 * @param error_name the name to check (must not be #NULL)
 *
 * @returns #TRUE if the message is the specified error
 */
dbus_bool_t
dbus_message_is_error (DBusMessage *message,
                      const char *error_name)
{
    const char *n;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (error_name != NULL, FALSE);
    /* don't check that error_name is valid since it would be expensive,
     * and not catch many common errors
     */

    if (dbus_message_get_type (message) != DBUS_MESSAGE_TYPE_ERROR)
        return FALSE;

    n = dbus_message_get_error_name (message);

    if (n && strcmp (n, error_name) == 0)
        return TRUE;
    else
        return FALSE;
}

/**
 * Checks whether the message was sent to the given name.  If the

```

```

* message has no destination specified or has a different
* destination, returns #FALSE.
*
* @param message the message
* @param name the name to check (must not be #NULL)
*
* @returns #TRUE if the message has the given destination name
*/
dbus_bool_t
dbus_message_has_destination (DBusMessage *message,
                             const char *name)
{
    const char *s;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (name != NULL, FALSE);
    /* don't check that name is valid since it would be expensive, and
     * not catch many common errors
     */

    s = dbus_message_get_destination (message);

    if (s && strcmp (s, name) == 0)
        return TRUE;
    else
        return FALSE;
}

/**
 * Checks whether the message has the given unique name as its sender.
 * If the message has no sender specified or has a different sender,
 * returns #FALSE. Note that a peer application will always have the
 * unique name of the connection as the sender. So you can't use this
 * function to see whether a sender owned a well-known name.
 *
 * Messages from the bus itself will have #DBUS_SERVICE_DBUS
 * as the sender.
 *
 * @param message the message
 * @param name the name to check (must not be #NULL)
 *
 * @returns #TRUE if the message has the given sender
 */
dbus_bool_t
dbus_message_has_sender (DBusMessage *message,
                        const char *name)
{
    const char *s;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (name != NULL, FALSE);
    /* don't check that name is valid since it would be expensive, and

```

```

    * not catch many common errors
    */

    s = dbus_message_get_sender (message);

    if (s && strcmp (s, name) == 0)
        return TRUE;
    else
        return FALSE;
}

/**
 * Checks whether the message has the given signature; see
 * dbus_message_get_signature() for more details on what the signature
 * looks like.
 *
 * @param message the message
 * @param signature typecode array
 * @returns #TRUE if message has the given signature
 */
dbus_bool_t
dbus_message_has_signature (DBusMessage *message,
                           const char *signature)
{
    const char *s;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (signature != NULL, FALSE);
    /* don't check that signature is valid since it would be expensive,
     * and not catch many common errors
     */

    s = dbus_message_get_signature (message);

    if (s && strcmp (s, signature) == 0)
        return TRUE;
    else
        return FALSE;
}

/**
 * Sets a #DBusError based on the contents of the given
 * message. The error is only set if the message
 * is an error message, as in #DBUS_MESSAGE_TYPE_ERROR.
 * The name of the error is set to the name of the message,
 * and the error message is set to the first argument
 * if the argument exists and is a string.
 *
 * The return value indicates whether the error was set (the error is
 * set if and only if the message is an error message). So you can
 * check for an error reply and convert it to DBusError in one go:
 * @code

```



```

*   if (dbus_set_error_from_message (error, reply))
*       return error;
*   else
*       process reply;
* @endcode
*
* @param error the error to set
* @param message the message to set it from
* @returns #TRUE if the message had type #DBUS_MESSAGE_TYPE_ERROR
*/
dbus_bool_t
dbus_set_error_from_message (DBusError  *error,
                             DBusMessage *message)
{
    const char *str;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_error_is_set (error, FALSE);

    if (dbus_message_get_type (message) != DBUS_MESSAGE_TYPE_ERROR)
        return FALSE;

    str = NULL;
    dbus_message_get_args (message, NULL,
                           DBUS_TYPE_STRING, &str,
                           DBUS_TYPE_INVALID);

    dbus_set_error (error, dbus_message_get_error_name (message),
                   str ? "%s" : NULL, str);

    return TRUE;
}

/**
 * Checks whether a message contains unix fds
 *
 * @param message the message
 * @returns #TRUE if the message contains unix fds
 */
dbus_bool_t
dbus_message_contains_unix_fds (DBusMessage *message)
{
#ifdef HAVE_UNIX_FD_PASSING
    _dbus_assert (message);

    return message->n_unix_fds > 0;
#else
    return FALSE;
#endif
}

/** @} */

```

```

/**
 * @addtogroup DBusMessageInternals
 *
 * @{
 */

/**
 * The initial buffer size of the message loader.
 *
 * @todo this should be based on min header size plus some average
 * body size, or something. Or rather, the min header size only, if we
 * want to try to read only the header, store that in a DBusMessage,
 * then read only the body and store that, etc., depends on
 * how we optimize _dbus_message_loader_get_buffer() and what
 * the exact message format is.
 */
#define INITIAL_LOADER_DATA_LEN 32

/**
 * Creates a new message loader. Returns #NULL if memory can't
 * be allocated.
 *
 * @returns new loader, or #NULL.
 */
DBusMessageLoader*
_dbus_message_loader_new (void)
{
    DBusMessageLoader *loader;

    loader = dbus_new0 (DBusMessageLoader, 1);
    if (loader == NULL)
        return NULL;

    loader->refcount = 1;

    loader->corrupted = FALSE;
    loader->corruption_reason = DBUS_VALID;

    /* this can be configured by the app, but defaults to the protocol
max */
    loader->max_message_size = DBUS_MAXIMUM_MESSAGE_LENGTH;

    /* We set a very relatively conservative default here since due to
how
SCM_RIGHTS works we need to preallocate an fd array of the maximum
number of unix fds we want to receive in advance. A
try-and-reallocate loop is not possible. */
    loader->max_message_unix_fds = 1024;

    if (!_dbus_string_init (&loader->data))
        {

```

```

        dbus_free (loader);
        return NULL;
    }

    /* preallocate the buffer for speed, ignore failure */
    _dbus_string_set_length (&loader->data, INITIAL_LOADER_DATA_LEN);
    _dbus_string_set_length (&loader->data, 0);

#ifdef HAVE_UNIX_FD_PASSING
    loader->unix_fds = NULL;
    loader->n_unix_fds = loader->n_unix_fds_allocated = 0;
    loader->unix_fds_outstanding = FALSE;
#endif

    return loader;
}

/**
 * Increments the reference count of the loader.
 *
 * @param loader the loader.
 * @returns the loader
 */
DBusMessageLoader *
_dbus_message_loader_ref (DBusMessageLoader *loader)
{
    loader->refcount += 1;

    return loader;
}

/**
 * Decrements the reference count of the loader and finalizes the
 * loader when the count reaches zero.
 *
 * @param loader the loader.
 */
void
_dbus_message_loader_unref (DBusMessageLoader *loader)
{
    loader->refcount -= 1;
    if (loader->refcount == 0)
    {
#ifdef HAVE_UNIX_FD_PASSING
        close_unix_fds(loader->unix_fds, &loader->n_unix_fds);
        dbus_free(loader->unix_fds);
#endif
        _dbus_list_foreach (&loader->messages,
            (DBusForeachFunction) dbus_message_unref,
            NULL);
        _dbus_list_clear (&loader->messages);
        _dbus_string_free (&loader->data);
    }
}

```

```

        dbus_free (loader);
    }
}

/**
 * Gets the buffer to use for reading data from the network. Network
 * data is read directly into an allocated buffer, which is then used
 * in the DBusMessage, to avoid as many extra memcpy's as possible.
 * The buffer must always be returned immediately using
 * _dbus_message_loader_return_buffer(), even if no bytes are
 * successfully read.
 *
 * @todo this function can be a lot more clever. For example
 * it can probably always return a buffer size to read exactly
 * the body of the next message, thus avoiding any memory wastage
 * or reallocs.
 *
 * @todo we need to enforce a max length on strings in header fields.
 *
 * @param loader the message loader.
 * @param buffer the buffer
 */
void
_dbus_message_loader_get_buffer (DBusMessageLoader *loader,
                                DBusString **buffer)
{
    _dbus_assert (!loader->buffer_outstanding);

    *buffer = &loader->data;

    loader->buffer_outstanding = TRUE;
}

/**
 * Returns a buffer obtained from _dbus_message_loader_get_buffer(),
 * indicating to the loader how many bytes of the buffer were filled
 * in. This function must always be called, even if no bytes were
 * successfully read.
 *
 * @param loader the loader.
 * @param buffer the buffer.
 * @param bytes_read number of bytes that were read into the buffer.
 */
void
_dbus_message_loader_return_buffer (DBusMessageLoader *loader,
                                   DBusString *buffer,
                                   int bytes_read)
{
    _dbus_assert (loader->buffer_outstanding);
    _dbus_assert (buffer == &loader->data);

    loader->buffer_outstanding = FALSE;
}

```

```

}

/**
 * Gets the buffer to use for reading unix fds from the network.
 *
 * This works similar to _dbus_message_loader_get_buffer()
 *
 * @param loader the message loader.
 * @param fds the array to read fds into
 * @param max_n_fds how many fds to read at most
 * @return TRUE on success, FALSE on OOM
 */
dbus_bool_t
_dbus_message_loader_get_unix_fds(DBusMessageLoader *loader,
                                  int **fds,
                                  unsigned *max_n_fds)
{
#ifdef HAVE_UNIX_FD_PASSING
    _dbus_assert (!loader->unix_fds_outstanding);

    /* Allocate space where we can put the fds we read. We allocate
       space for max_message_unix_fds since this is an
       upper limit how many fds can be received within a single
       message. Since SCM_RIGHTS doesn't allow a reallocate+retry logic
       we are allocating the maximum possible array size right from the
       beginning. This sucks a bit, however unless SCM_RIGHTS is fixed
       there is no better way. */

    if (loader->n_unix_fds_allocated < loader->max_message_unix_fds)
    {
        int *a = dbus_realloc(loader->unix_fds,
                              loader->max_message_unix_fds *
sizeof(loader->unix_fds[0]));

        if (!a)
            return FALSE;

        loader->unix_fds = a;
        loader->n_unix_fds_allocated = loader->max_message_unix_fds;
    }

    *fds = loader->unix_fds + loader->n_unix_fds;
    *max_n_fds = loader->n_unix_fds_allocated - loader->n_unix_fds;

    loader->unix_fds_outstanding = TRUE;
    return TRUE;
#else
    _dbus_assert_not_reached("Platform doesn't support unix fd
passing");
    return FALSE;
#endif
}

```

```

/**
 * Returns a buffer obtained from _dbus_message_loader_get_unix_fds().
 *
 * This works similar to _dbus_message_loader_return_buffer()
 *
 * @param loader the message loader.
 * @param fds the array fds were read into
 * @param max_n_fds how many fds were read
 */

void
_dbus_message_loader_return_unix_fds(DBusMessageLoader *loader,
                                     int *fds,
                                     unsigned n_fds)
{
#ifdef HAVE_UNIX_FD_PASSING
    _dbus_assert(loader->unix_fds_outstanding);
    _dbus_assert(loader->unix_fds + loader->n_unix_fds == fds);
    _dbus_assert(loader->n_unix_fds + n_fds <= loader->n_unix_fds_allocated);

    loader->n_unix_fds += n_fds;
    loader->unix_fds_outstanding = FALSE;
#else
    _dbus_assert_not_reached("Platform doesn't support unix fd
passing");
#endif
}

/*
 * FIXME when we move the header out of the buffer, that memmoves all
 * buffered messages. Kind of crappy.
 *
 * Also we copy the header and body, which is kind of crappy. To
 * avoid this, we have to allow header and body to be in a single
 * memory block, which is good for messages we read and bad for
 * messages we are creating. But we could move_len() the buffer into
 * this single memory block, and move_len() will just swap the buffers
 * if you're moving the entire buffer replacing the dest string.
 *
 * We could also have the message loader tell the transport how many
 * bytes to read; so it would first ask for some arbitrary number like
 * 256, then if the message was incomplete it would use the
 * header/body len to ask for exactly the size of the message (or
 * blocks the size of a typical kernel buffer for the socket). That
 * way we don't get trailing bytes in the buffer that have to be
 * memmoved. Though I suppose we also don't have a chance of reading a
 * bunch of small messages at once, so the optimization may be stupid.
 *
 * Another approach would be to keep a "start" index into
 * loader->data and only delete it occasionally, instead of after

```

```

* each message is loaded.
*
* load_message() returns FALSE if not enough memory OR the loader was
corrupted
*/
static dbus_bool_t
load_message (DBusMessageLoader *loader,
             DBusMessage        *message,
             int                 byte_order,
             int                 fields_array_len,
             int                 header_len,
             int                 body_len)
{
    dbus_bool_t oom;
    DBusValidity validity;
    const DBusString *type_str;
    int type_pos;
    DBusValidationMode mode;
    dbus_uint32_t n_unix_fds = 0;

    mode = DBUS_VALIDATION_MODE_DATA_IS_UNTRUSTED;

    oom = FALSE;

#ifdef 0
    _dbus_verbose_bytes_of_string (&loader->data, 0, header_len /* +
body_len */);
#endif

    /* 1. VALIDATE AND COPY OVER HEADER */
    _dbus_assert (_dbus_string_get_length (&message->header.data) == 0);
    _dbus_assert ((header_len + body_len) <= _dbus_string_get_length
(&loader->data));

    if (!_dbus_header_load (&message->header,
                          mode,
                          &validity,
                          byte_order,
                          fields_array_len,
                          header_len,
                          body_len,
                          &loader->data, 0,
                          _dbus_string_get_length (&loader->data)))
    {
        _dbus_verbose ("Failed to load header for new message code
%d\n", validity);

        /* assert here so we can catch any code that still uses
DBUS_VALID to indicate
oom errors.  They should use DBUS_VALIDITY_UNKNOWN_OOM_ERROR
instead */
        _dbus_assert (validity != DBUS_VALID);
    }
}

```

```

    if (validity == DBUS_VALIDITY_UNKNOWN_OOM_ERROR)
        oom = TRUE;
    else
        {
            loader->corrupted = TRUE;
            loader->corruption_reason = validity;
        }
    goto failed;
}

_dbus_assert (validity == DBUS_VALID);

/* 2. VALIDATE BODY */
if (mode != DBUS_VALIDATION_MODE_WE_TRUST_THIS_DATA_ABSOLUTELY)
{
    get_const_signature (&message->header, &type_str, &type_pos);

    /* Because the bytes_remaining arg is NULL, this validates that
the
    * body is the right length
    */
    validity = _dbus_validate_body_with_reason (type_str,
                                                type_pos,
                                                byte_order,
                                                NULL,
                                                &loader->data,
                                                header_len,
                                                body_len);

    if (validity != DBUS_VALID)
    {
        _dbus_verbose ("Failed to validate message body code %d\n",
validity);

        loader->corrupted = TRUE;
        loader->corruption_reason = validity;

        goto failed;
    }
}

/* 3. COPY OVER UNIX FDS */
_dbus_header_get_field_basic (&message->header,
                              DBUS_HEADER_FIELD_UNIX_FDS,
                              DBUS_TYPE_UINT32,
                              &n_unix_fds);

#ifdef HAVE_UNIX_FD_PASSING

    if (n_unix_fds > loader->n_unix_fds)
        {

```



```

        _dbus_verbose("Message contains references to more unix fds than
were sent %u != %u\n",
                    n_unix_fds, loader->n_unix_fds);

        loader->corrupted = TRUE;
        loader->corruption_reason = DBUS_INVALID_MISSING_UNIX_FDS;
        goto failed;
    }

    /* If this was a recycled message there might still be
    some memory allocated for the fds */
    dbus_free(message->unix_fds);

    if (n_unix_fds > 0)
    {
        message->unix_fds = _dbus_memdup(loader->unix_fds, n_unix_fds *
sizeof(message->unix_fds[0]));
        if (message->unix_fds == NULL)
        {
            _dbus_verbose ("Failed to allocate file descriptor
array\n");
            oom = TRUE;
            goto failed;
        }

        message->n_unix_fds_allocated = message->n_unix_fds =
n_unix_fds;
        loader->n_unix_fds -= n_unix_fds;
        memmove(loader->unix_fds + n_unix_fds, loader->unix_fds, loader-
>n_unix_fds);
    }
    else
        message->unix_fds = NULL;

#else

    if (n_unix_fds > 0)
    {
        _dbus_verbose ("Hmm, message claims to come with file
descriptors "
                    "but that's not supported on our platform,
disconnecting.\n");

        loader->corrupted = TRUE;
        loader->corruption_reason = DBUS_INVALID_MISSING_UNIX_FDS;
        goto failed;
    }

#endif

    /* 3. COPY OVER BODY AND QUEUE MESSAGE */

```

```

if (!_dbus_list_append (&loader->messages, message))
{
    _dbus_verbose ("Failed to append new message to loader
queue\n");
    oom = TRUE;
    goto failed;
}

_dbus_assert (_dbus_string_get_length (&message->body) == 0);
_dbus_assert (_dbus_string_get_length (&loader->data) >=
(header_len + body_len));

if (!_dbus_string_copy_len (&loader->data, header_len, body_len,
&message->body, 0))
{
    _dbus_verbose ("Failed to move body into new message\n");
    oom = TRUE;
    goto failed;
}

_dbus_string_delete (&loader->data, 0, header_len + body_len);

/* don't waste more than 2k of memory */
_dbus_string_compact (&loader->data, 2048);

_dbus_assert (_dbus_string_get_length (&message->header.data) ==
header_len);
_dbus_assert (_dbus_string_get_length (&message->body) == body_len);

_dbus_verbose ("Loaded message %p\n", message);

_dbus_assert (!oom);
_dbus_assert (!loader->corrupted);
_dbus_assert (loader->messages != NULL);
_dbus_assert (_dbus_list_find_last (&loader->messages, message) !=
NULL);

return TRUE;

failed:

/* Clean up */

/* does nothing if the message isn't in the list */
_dbus_list_remove_last (&loader->messages, message);

if (oom)
    _dbus_assert (!loader->corrupted);
else
    _dbus_assert (loader->corrupted);

```

```

    _dbus_verbose_bytes_of_string (&loader->data, 0,
    _dbus_string_get_length (&loader->data));

    return FALSE;
}

/**
 * Converts buffered data into messages, if we have enough data. If
 * we don't have enough data, does nothing.
 *
 * @todo we need to check that the proper named header fields exist
 * for each message type.
 *
 * @todo If a message has unknown type, we should probably eat it
 * right here rather than passing it out to applications. However
 * it's not an error to see messages of unknown type.
 *
 * @param loader the loader.
 * @returns #TRUE if we had enough memory to finish.
 */
dbus_bool_t
_dbus_message_loader_queue_messages (DBusMessageLoader *loader)
{
    while (!loader->corrupted &&
           _dbus_string_get_length (&loader->data) >=
           DBUS_MINIMUM_HEADER_SIZE)
    {
        DBusValidity validity;
        int byte_order, fields_array_len, header_len, body_len;

        if (_dbus_header_have_message_untrusted (loader->
        >max_message_size,
                                                &validity,
                                                &byte_order,
                                                &fields_array_len,
                                                &header_len,
                                                &body_len,
                                                &loader->data, 0,
                                                _dbus_string_get_length
        (&loader->data)))
        {
            DBusMessage *message;

            _dbus_assert (validity == DBUS_VALID);

            message = dbus_message_new_empty_header ();
            if (message == NULL)
                return FALSE;

            if (!load_message (loader, message,
                              byte_order, fields_array_len,
                              header_len, body_len))

```

```

        {
            dbus_message_unref (message);
            /* load_message() returns false if corrupted or OOM; if
             * corrupted then return TRUE for not OOM
             */
            return loader->corrupted;
        }

        _dbus_assert (loader->messages != NULL);
        _dbus_assert (_dbus_list_find_last (&loader->messages,
message) != NULL);
    }
    else
    {
        _dbus_verbose ("Initial peek at header says we don't have a
whole message yet, or data broken with invalid code %d\n",
        validity);
        if (validity != DBUS_VALID)
        {
            loader->corrupted = TRUE;
            loader->corruption_reason = validity;
        }
        return TRUE;
    }
}

return TRUE;
}

/**
 * Peeks at first loaded message, returns #NULL if no messages have
 * been queued.
 *
 * @param loader the loader.
 * @returns the next message, or #NULL if none.
 */
DBusMessage*
_dbus_message_loader_peek_message (DBusMessageLoader *loader)
{
    if (loader->messages)
        return loader->messages->data;
    else
        return NULL;
}

/**
 * Pops a loaded message (passing ownership of the message
 * to the caller). Returns #NULL if no messages have been
 * queued.
 *
 * @param loader the loader.
 * @returns the next message, or #NULL if none.

```

```

*/
DBusMessage*
_dbus_message_loader_pop_message (DBusMessageLoader *loader)
{
    return _dbus_list_pop_first (&loader->messages);
}

/**
 * Pops a loaded message inside a list link (passing ownership of the
 * message and link to the caller). Returns #NULL if no messages have
 * been loaded.
 *
 * @param loader the loader.
 * @returns the next message link, or #NULL if none.
 */
DBusList*
_dbus_message_loader_pop_message_link (DBusMessageLoader *loader)
{
    return _dbus_list_pop_first_link (&loader->messages);
}

/**
 * Returns a popped message link, used to undo a pop.
 *
 * @param loader the loader
 * @param link the link with a message in it
 */
void
_dbus_message_loader_putback_message_link (DBusMessageLoader *loader,
                                           DBusList *link)
{
    _dbus_list_prepend_link (&loader->messages, link);
}

/**
 * Checks whether the loader is confused due to bad data.
 * If messages are received that are invalid, the
 * loader gets confused and gives up permanently.
 * This state is called "corrupted."
 *
 * @param loader the loader
 * @returns #TRUE if the loader is hosed.
 */
dbus_bool_t
_dbus_message_loader_get_is_corrupted (DBusMessageLoader *loader)
{
    _dbus_assert ((loader->corrupted && loader->corruption_reason !=
DBUS_VALID) ||
                (!loader->corrupted && loader->corruption_reason ==
DBUS_VALID));
    return loader->corrupted;
}

```

```

/**
 * Checks what kind of bad data confused the loader.
 *
 * @param loader the loader
 * @returns why the loader is hosed, or DBUS_VALID if it isn't.
 */
DBusValidity
_dbus_message_loader_get_corruption_reason (DBusMessageLoader *loader)
{
    _dbus_assert ((loader->corrupted && loader->corruption_reason !=
DBUS_VALID) ||
                (!loader->corrupted && loader->corruption_reason ==
DBUS_VALID));

    return loader->corruption_reason;
}

/**
 * Sets the maximum size message we allow.
 *
 * @param loader the loader
 * @param size the max message size in bytes
 */
void
_dbus_message_loader_set_max_message_size (DBusMessageLoader *loader,
                                           long size)
{
    if (size > DBUS_MAXIMUM_MESSAGE_LENGTH)
    {
        _dbus_verbose ("clamping requested max message size %ld to
%d\n",
                    size, DBUS_MAXIMUM_MESSAGE_LENGTH);
        size = DBUS_MAXIMUM_MESSAGE_LENGTH;
    }
    loader->max_message_size = size;
}

/**
 * Gets the maximum allowed message size in bytes.
 *
 * @param loader the loader
 * @returns max size in bytes
 */
long
_dbus_message_loader_get_max_message_size (DBusMessageLoader *loader)
{
    return loader->max_message_size;
}

/**
 * Sets the maximum unix fds per message we allow.

```

```

*
* @param loader the loader
* @param size the max number of unix fds in a message
*/
void
_dbus_message_loader_set_max_message_unix_fds (DBusMessageLoader
*loader,
                                                long
                                                n)
{
    if (n > DBUS_MAXIMUM_MESSAGE_UNIX_FDS)
    {
        _dbus_verbose ("clamping requested max message unix_fds %ld to
%d\n",
                      n, DBUS_MAXIMUM_MESSAGE_UNIX_FDS);
        n = DBUS_MAXIMUM_MESSAGE_UNIX_FDS;
    }
    loader->max_message_unix_fds = n;
}

/**
 * Gets the maximum allowed number of unix fds per message
 *
 * @param loader the loader
 * @returns max unix fds
 */
long
_dbus_message_loader_get_max_message_unix_fds (DBusMessageLoader
*loader)
{
    return loader->max_message_unix_fds;
}

static DBusDataSlotAllocator slot_allocator;
_DBUS_DEFINE_GLOBAL_LOCK (message_slots);

/**
 * Allocates an integer ID to be used for storing application-specific
 * data on any DBusMessage. The allocated ID may then be used
 * with dbus_message_set_data() and dbus_message_get_data().
 * The passed-in slot must be initialized to -1, and is filled in
 * with the slot ID. If the passed-in slot is not -1, it's assumed
 * to be already allocated, and its refcount is incremented.
 *
 * The allocated slot is global, i.e. all DBusMessage objects will
 * have a slot with the given integer ID reserved.
 *
 * @param slot_p address of a global variable storing the slot
 * @returns #FALSE on failure (no memory)
 */
dbus_bool_t
dbus_message_allocate_data_slot (dbus_int32_t *slot_p)
{

```

```

    return _dbus_data_slot_allocator_alloc (&slot_allocator,
                                           &_DBUS_LOCK_NAME
(message_slots),
                                           slot_p);
}

/**
 * Deallocates a global ID for message data slots.
 * dbus_message_get_data() and dbus_message_set_data() may no
 * longer be used with this slot. Existing data stored on existing
 * DbusMessage objects will be freed when the message is
 * finalized, but may not be retrieved (and may only be replaced if
 * someone else reallocates the slot). When the refcount on the
 * passed-in slot reaches 0, it is set to -1.
 *
 * @param slot_p address storing the slot to deallocate
 */
void
dbus_message_free_data_slot (dbus_int32_t *slot_p)
{
    _dbus_return_if_fail (*slot_p >= 0);

    _dbus_data_slot_allocator_free (&slot_allocator, slot_p);
}

/**
 * Stores a pointer on a DbusMessage, along
 * with an optional function to be used for freeing
 * the data when the data is set again, or when
 * the message is finalized. The slot number
 * must have been allocated with dbus_message_allocate_data_slot().
 *
 * @param message the message
 * @param slot the slot number
 * @param data the data to store
 * @param free_data_func finalizer function for the data
 * @returns #TRUE if there was enough memory to store the data
 */
dbus_bool_t
dbus_message_set_data (DBusMessage *message,
                      dbus_int32_t slot,
                      void *data,
                      DbusFreeFunction free_data_func)
{
    DbusFreeFunction old_free_func;
    void *old_data;
    dbus_bool_t retval;

    _dbus_return_val_if_fail (message != NULL, FALSE);
    _dbus_return_val_if_fail (slot >= 0, FALSE);

    retval = _dbus_data_slot_list_set (&slot_allocator,

```



```

        &message->slot_list,
        slot, data, free_data_func,
        &old_free_func, &old_data);

    if (retval)
    {
        /* Do the actual free outside the message lock */
        if (old_free_func)
            (* old_free_func) (old_data);
    }

    return retval;
}

/**
 * Retrieves data previously set with dbus_message_set_data().
 * The slot must still be allocated (must not have been freed).
 *
 * @param message the message
 * @param slot the slot to get data from
 * @returns the data, or #NULL if not found
 */
void*
dbus_message_get_data (DBusMessage *message,
                      dbus_int32_t slot)
{
    void *res;

    _dbus_return_val_if_fail (message != NULL, NULL);

    res = _dbus_data_slot_list_get (&slot_allocator,
                                    &message->slot_list,
                                    slot);

    return res;
}

/**
 * Utility function to convert a machine-readable (not translated)
 * string into a D-Bus message type.
 *
 * @code
 * "method_call"    -> DBUS_MESSAGE_TYPE_METHOD_CALL
 * "method_return" -> DBUS_MESSAGE_TYPE_METHOD_RETURN
 * "signal"         -> DBUS_MESSAGE_TYPE_SIGNAL
 * "error"          -> DBUS_MESSAGE_TYPE_ERROR
 * anything else   -> DBUS_MESSAGE_TYPE_INVALID
 * @endcode
 */
int
dbus_message_type_from_string (const char *type_str)

```

```

{
    if (strcmp (type_str, "method_call") == 0)
        return DBUS_MESSAGE_TYPE_METHOD_CALL;
    if (strcmp (type_str, "method_return") == 0)
        return DBUS_MESSAGE_TYPE_METHOD_RETURN;
    else if (strcmp (type_str, "signal") == 0)
        return DBUS_MESSAGE_TYPE_SIGNAL;
    else if (strcmp (type_str, "error") == 0)
        return DBUS_MESSAGE_TYPE_ERROR;
    else
        return DBUS_MESSAGE_TYPE_INVALID;
}

/**
 * Utility function to convert a D-Bus message type into a
 * machine-readable string (not translated).
 *
 * @code
 *     DBUS_MESSAGE_TYPE_METHOD_CALL      -> "method_call"
 *     DBUS_MESSAGE_TYPE_METHOD_RETURN   -> "method_return"
 *     DBUS_MESSAGE_TYPE_SIGNAL          -> "signal"
 *     DBUS_MESSAGE_TYPE_ERROR           -> "error"
 *     DBUS_MESSAGE_TYPE_INVALID         -> "invalid"
 * @endcode
 */
const char *
dbus_message_type_to_string (int type)
{
    switch (type)
    {
        case DBUS_MESSAGE_TYPE_METHOD_CALL:
            return "method_call";
        case DBUS_MESSAGE_TYPE_METHOD_RETURN:
            return "method_return";
        case DBUS_MESSAGE_TYPE_SIGNAL:
            return "signal";
        case DBUS_MESSAGE_TYPE_ERROR:
            return "error";
        default:
            return "invalid";
    }
}

/**
 * Turn a D-BusMessage into the marshalled form as described in the D-
 * Bus
 * specification.
 *
 * Generally, this function is only useful for encapsulating D-Bus
 * messages in
 * a different protocol.

```

```

*
* @param msg the DBusMessage
* @param marshalled_data_p the location to save the marshalled form
to
* @param len_p the location to save the length of the marshalled form
to
* @returns #FALSE if there was not enough memory
*/
dbus_bool_t
dbus_message_marshall (DBusMessage *msg,
                      char **marshalled_data_p,
                      int *len_p)
{
    DBusString tmp;
    dbus_bool_t was_locked;

    _dbus_return_val_if_fail (msg != NULL, FALSE);
    _dbus_return_val_if_fail (marshalled_data_p != NULL, FALSE);
    _dbus_return_val_if_fail (len_p != NULL, FALSE);

    if (!_dbus_string_init (&tmp))
        return FALSE;

    /* Ensure the message is locked, to ensure the length header is
filled in. */
    was_locked = msg->locked;

    if (!was_locked)
        dbus_message_lock (msg);

    if (!_dbus_string_copy (&(msg->header.data), 0, &tmp, 0))
        goto fail;

    *len_p = _dbus_string_get_length (&tmp);

    if (!_dbus_string_copy (&(msg->body), 0, &tmp, *len_p))
        goto fail;

    *len_p = _dbus_string_get_length (&tmp);

    if (!_dbus_string_steal_data (&tmp, marshalled_data_p))
        goto fail;

    _dbus_string_free (&tmp);

    if (!was_locked)
        msg->locked = FALSE;

    return TRUE;

fail:
    _dbus_string_free (&tmp);

```

```

    if (!was_locked)
        msg->locked = FALSE;

    return FALSE;
}

/**
 * Demarshal a D-Bus message from the format described in the D-Bus
 * specification.
 *
 * Generally, this function is only useful for encapsulating D-Bus
messages in
 * a different protocol.
 *
 * @param str the marshalled DBusMessage
 * @param len the length of str
 * @param error the location to save errors to
 * @returns #NULL if there was an error
 */
DBusMessage *
dbus_message_demarshal (const char *str,
                       int          len,
                       DBusError  *error)
{
    DBusMessageLoader *loader;
    DBusString *buffer;
    DBusMessage *msg;

    _dbus_return_val_if_fail (str != NULL, NULL);

    loader = _dbus_message_loader_new ();

    if (loader == NULL)
        return NULL;

    _dbus_message_loader_get_buffer (loader, &buffer);
    _dbus_string_append_len (buffer, str, len);
    _dbus_message_loader_return_buffer (loader, buffer, len);

    if (!_dbus_message_loader_queue_messages (loader))
        goto fail_oom;

    if (_dbus_message_loader_get_is_corrupted (loader))
        goto fail_corrupt;

    msg = _dbus_message_loader_pop_message (loader);

    if (!msg)
        goto fail_oom;

    _dbus_message_loader_unref (loader);

```

```

return msg;

fail_corrupt:
    dbus_set_error (error, DBUS_ERROR_INVALID_ARGS, "Message is
corrupted (%s)",
                    _dbus_validity_to_error_message (loader-
>corruption_reason));
    _dbus_message_loader_unref (loader);
    return NULL;

fail_oom:
    _DBUS_SET_OOM (error);
    _dbus_message_loader_unref (loader);
    return NULL;
}

/**
 * Returns the number of bytes required to be in the buffer to
demarshal a
 * D-Bus message.
 *
 * Generally, this function is only useful for encapsulating D-Bus
messages in
 * a different protocol.
 *
 * @param str data to be marshalled
 * @param len the length of str
 * @param error the location to save errors to
 * @returns -1 if there was no valid data to be demarshalled, 0 if
there wasn't enough data to determine how much should be demarshalled.
Otherwise returns the number of bytes to be demarshalled
 *
 */
int
dbus_message_demarshal_bytes_needed(const char *buf,
                                   int          len)
{
    DBusString str;
    int byte_order, fields_array_len, header_len, body_len;
    DBusValidity validity = DBUS_VALID;
    int have_message;

    if (!buf || len < DBUS_MINIMUM_HEADER_SIZE)
        return 0;

    if (len > DBUS_MAXIMUM_MESSAGE_LENGTH)
        len = DBUS_MAXIMUM_MESSAGE_LENGTH;
    _dbus_string_init_const_len (&str, buf, len);

    validity = DBUS_VALID;
    have_message
        = _dbus_header_have_message_untrusted(DBUS_MAXIMUM_MESSAGE_LENGTH,

```

```

        &validity, &byte_order,
        &fields_array_len,
        &header_len,
        &body_len,
        &str, 0,
        len);

_dbus_string_free (&str);

if (validity == DBUS_VALID)
{
    _dbus_assert (have_message || (header_len + body_len) > len);
    (void) have_message; /* unused unless asserting */
    return header_len + body_len;
}
else
{
    return -1; /* broken! */
}
}

/** @} */

/* tests in dbus-message-util.c */

File = dbus-message.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-message.h DbusMessage object
 *
 * Copyright (C) 2002, 2003, 2005 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 *

```

```

 */
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_MESSAGE_H
#define DBUS_MESSAGE_H

#include <dbus/dbus-macros.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-arch-deps.h>
#include <dbus/dbus-memory.h>
#include <dbus/dbus-errors.h>
#include <stdarg.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DbusMessage
 * @{
 */

typedef struct DbusMessage DbusMessage;
/** Opaque type representing a message iterator. Can be copied by
value, and contains no allocated memory so never needs to be freed and
can be allocated on the stack. */
typedef struct DbusMessageIter DbusMessageIter;

/**
 * DbusMessageIter struct; contains no public fields.
 */
struct DbusMessageIter
{
    void *dummy1;          /**< Don't use this */
    void *dummy2;          /**< Don't use this */
    dbus_uint32_t dummy3; /**< Don't use this */
    int dummy4;           /**< Don't use this */
    int dummy5;           /**< Don't use this */
    int dummy6;           /**< Don't use this */
    int dummy7;           /**< Don't use this */
    int dummy8;           /**< Don't use this */
    int dummy9;           /**< Don't use this */
    int dummy10;          /**< Don't use this */
    int dummy11;          /**< Don't use this */
    int pad1;             /**< Don't use this */
    int pad2;             /**< Don't use this */
    void *pad3;           /**< Don't use this */
};

DBUS_EXPORT

```

```

DBusMessage* dbus_message_new          (int
message_type);
DBUS_EXPORT
DBusMessage* dbus_message_new_method_call (const char *bus_name,
                                           const char *path,
                                           const char *interface,
                                           const char *method);

DBUS_EXPORT
DBusMessage* dbus_message_new_method_return (DBusMessage
*method_call);
DBUS_EXPORT
DBusMessage* dbus_message_new_signal     (const char *path,
                                           const char *interface,
                                           const char *name);

DBUS_EXPORT
DBusMessage* dbus_message_new_error     (DBusMessage *reply_to,
                                           const char *error_name,
                                           const char
*error_message);
DBUS_EXPORT
DBusMessage* dbus_message_new_error_printf (DBusMessage *reply_to,
                                           const char *error_name,
                                           const char
*error_format,
                                           ...);

DBUS_EXPORT
DBusMessage* dbus_message_copy          (const DBusMessage
*message);

DBUS_EXPORT
DBusMessage* dbus_message_ref           (DBusMessage *message);
DBUS_EXPORT
void         dbus_message_unref         (DBusMessage *message);
DBUS_EXPORT
int         dbus_message_get_type       (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t dbus_message_set_path       (DBusMessage *message,
                                           const char
*object_path);
DBUS_EXPORT
const char* dbus_message_get_path       (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t dbus_message_has_path       (DBusMessage *message,
                                           const char
*object_path);
DBUS_EXPORT
dbus_bool_t dbus_message_set_interface  (DBusMessage *message,
                                           const char
*interface);
DBUS_EXPORT
const char* dbus_message_get_interface  (DBusMessage *message);

```



```

DBUS_EXPORT
dbus_bool_t    dbus_message_has_interface    (DBusMessage *message,
const char
*interface);
DBUS_EXPORT
dbus_bool_t    dbus_message_set_member      (DBusMessage *message,
const char *member);
DBUS_EXPORT
const char*    dbus_message_get_member      (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t    dbus_message_has_member      (DBusMessage *message,
const char *member);
DBUS_EXPORT
dbus_bool_t    dbus_message_set_error_name   (DBusMessage *message,
const char *name);
DBUS_EXPORT
const char*    dbus_message_get_error_name   (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t    dbus_message_set_destination (DBusMessage *message,
const char *destination);
DBUS_EXPORT
const char*    dbus_message_get_destination (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t    dbus_message_set_sender      (DBusMessage *message,
const char *sender);
DBUS_EXPORT
const char*    dbus_message_get_sender      (DBusMessage *message);
DBUS_EXPORT
const char*    dbus_message_get_signature   (DBusMessage *message);
DBUS_EXPORT
void           dbus_message_set_no_reply     (DBusMessage *message,
dbus_bool_t no_reply);
DBUS_EXPORT
dbus_bool_t    dbus_message_get_no_reply     (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t    dbus_message_is_method_call   (DBusMessage *message,
const char *interface,
const char *method);
DBUS_EXPORT
dbus_bool_t    dbus_message_is_signal        (DBusMessage *message,
const char *interface,
const char *signal_name);
DBUS_EXPORT
dbus_bool_t    dbus_message_is_error         (DBusMessage *message,
const char *error_name);
DBUS_EXPORT
dbus_bool_t    dbus_message_has_destination (DBusMessage *message,
const char *bus_name);
DBUS_EXPORT

```



```

                                va_list          var_args);

DBUS_EXPORT
dbus_bool_t dbus_message_contains_unix_fds (DBusMessage *message);

DBUS_EXPORT
dbus_bool_t dbus_message_iter_init        (DBusMessage
*message,
                                           DbusMessageIter
*iter);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_has_next    (DBusMessageIter
*iter);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_next        (DBusMessageIter
*iter);
DBUS_EXPORT
char*       dbus_message_iter_get_signature (DBusMessageIter
*iter);
DBUS_EXPORT
int         dbus_message_iter_get_arg_type (DBusMessageIter
*iter);
DBUS_EXPORT
int         dbus_message_iter_get_element_type (DBusMessageIter
*iter);
DBUS_EXPORT
void        dbus_message_iter_recurse     (DBusMessageIter *iter,
                                           DbusMessageIter *sub);
DBUS_EXPORT
void        dbus_message_iter_get_basic   (DBusMessageIter *iter,
                                           void
*value);
#ifdef DBUS_DISABLE_DEPRECATED
/* This function returns the wire protocol size of the array in bytes,
 * you do not want to know that probably
 */
DBUS_EXPORT
DBUS_DEPRECATED int         dbus_message_iter_get_array_len
(DBusMessageIter *iter);
#endif
DBUS_EXPORT
void        dbus_message_iter_get_fixed_array (DBusMessageIter *iter,
void
*value,
                                           int
*n_elements);

DBUS_EXPORT
void        dbus_message_iter_init_append   (DBusMessage
*message,

```

```

                                                                    DbusMessageIter
*iter);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_append_basic (DBusMessageIter
*iter,
                                                                    int
type,
                                                                    const void
*value);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_append_fixed_array (DBusMessageIter
*iter,
                                                                    int
element_type,
                                                                    const void
*value,
                                                                    int
n_elements);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_open_container (DBusMessageIter
*iter,
                                                                    int
type,
                                                                    const char
*contained_signature,
                                                                    DbusMessageIter
*sub);
DBUS_EXPORT
dbus_bool_t dbus_message_iter_close_container (DBusMessageIter
*iter,
                                                                    DbusMessageIter
*sub);
DBUS_EXPORT
void          dbus_message_iter_abandon_container (DBusMessageIter
*iter,
                                                                    DbusMessageIter
*sub);
DBUS_EXPORT
void dbus_message_lock      (DBusMessage *message);
DBUS_EXPORT
dbus_bool_t  dbus_set_error_from_message (DBusError *error,
                                                                    DbusMessage *message);
DBUS_EXPORT
dbus_bool_t  dbus_message_allocate_data_slot (dbus_int32_t
*slot_p);
DBUS_EXPORT
void          dbus_message_free_data_slot      (dbus_int32_t
*slot_p);

```

```

DBUS_EXPORT
dbus_bool_t dbus_message_set_data      (DBusMessage
*message,
                                         dbus_int32_t      slot,
                                         void             *data,
                                         DBusFreeFunction

free_data_func);
DBUS_EXPORT
void*        dbus_message_get_data     (DBusMessage
*message,
                                         dbus_int32_t      slot);

DBUS_EXPORT
int          dbus_message_type_from_string (const char *type_str);
DBUS_EXPORT
const char* dbus_message_type_to_string   (int type);

DBUS_EXPORT
dbus_bool_t  dbus_message_marshall      (DBusMessage *msg,
                                         char         **marshalled_data_p,
                                         int          *len_p);

DBUS_EXPORT
DBusMessage* dbus_message_demarshal     (const char *str,
                                         int         len,
                                         DBusError  *error);

DBUS_EXPORT
int          dbus_message_demarshal_bytes_needed (const char *str,
                                                  int len);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_MESSAGE_H */

```

File = dbus-misc.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-misc.c A few assorted public functions that don't fit
elsewhere
*
* Copyright (C) 2006 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by

```

```
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-misc.h"
#include "dbus-internals.h"
#include "dbus-string.h"
```

```
/**
 * @defgroup DBusMisc Miscellaneous
 * @ingroup DBus
 * @brief Miscellaneous API that doesn't cleanly fit anywhere else
 *
 * @{
 */
/**
 * Obtains the machine UUID of the machine this process is running on.
 *
 * The returned string must be freed with dbus_free().
 *
 * This UUID is guaranteed to remain the same until the next reboot
 * (unless the sysadmin foolishly changes it and screws themselves).
 * It will usually remain the same across reboots also, but hardware
 * configuration changes or rebuilding the machine could break that.
 *
 * The idea is that two processes with the same machine ID should be
 * able to use shared memory, UNIX domain sockets, process IDs, and
other
 * features of the OS that require both processes to be running
 * on the same OS kernel instance.
 *
 * The machine ID can also be used to create unique per-machine
 * instances. For example, you could use it in bus names or
 * X selection names.
 *
 * The machine ID is preferred over the machine hostname, because
 * the hostname is frequently set to "localhost.localdomain" and
 * may also change at runtime.
 */
```

```

* You can get the machine ID of a remote application by invoking the
* method GetMachineId from interface org.freedesktop.DBus.Peer.
*
* If the remote application has the same machine ID as the one
* returned by this function, then the remote application is on the
* same machine as your application.
*
* The UUID is not a UUID in the sense of RFC4122; the details
* are explained in the D-Bus specification.
*
* @returns a 32-byte-long hex-encoded UUID string, or #NULL if
insufficient memory
*/
char*
dbus_get_local_machine_id (void)
{
    DBusString uuid;
    char *s;

    s = NULL;

    if (!_dbus_string_init (&uuid))
        return NULL;

    if (!_dbus_get_local_machine_uuid_encoded (&uuid) ||
        !_dbus_string_steal_data (&uuid, &s))
    {
        _dbus_string_free (&uuid);
        return NULL;
    }
    else
    {
        _dbus_string_free (&uuid);
        return s;
    }
}

/**
 * @def DBUS_MAJOR_VERSION
 *
 * The COMPILER TIME major version of libdbus, that is, the "X" in
 "X.Y.Z",
 * as an integer literal. Consider carefully whether to use this or
 the
 * runtime version from dbus_get_version().
 */

/**
 * @def DBUS_MINOR_VERSION
 *

```

```

* The COMPILER TIME minor version of libdbus, that is, the "Y" in
"X.Y.Z",
* as an integer literal. Consider carefully whether to use this or
the
* runtime version from dbus_get_version().
*/

/**
* @def DBUS_MICRO_VERSION
*
* The COMPILER TIME micro version of libdbus, that is, the "Z" in
"X.Y.Z",
* as an integer literal. Consider carefully whether to use this or
the
* runtime version from dbus_get_version().
*/

/**
* @def DBUS_VERSION
*
* The COMPILER TIME version of libdbus, as a single integer that has 0
in the most
* significant byte, the major version in the next most significant
byte,
* the minor version in the third most significant, and the micro
version in the
* least significant byte. This means two DBUS_VERSION can be compared
to see
* which is higher.
*
* Consider carefully whether to use this or the runtime version from
* dbus_get_version().
*/

/**
* @def DBUS_VERSION_STRING
*
* The COMPILER TIME version of libdbus, as a string "X.Y.Z".
*
* Consider carefully whether to use this or the runtime version from
* dbus_get_version().
*/

/**
* Gets the DYNAMICALLY LINKED version of libdbus. Alternatively,
there
* are macros #DBUS_MAJOR_VERSION, #DBUS_MINOR_VERSION,
#DBUS_MICRO_VERSION,
* and #DBUS_VERSION which allow you to test the VERSION YOU ARE
COMPILED AGAINST.
* In other words, you can get either the runtime or the compile-time
version.

```



```

* Think carefully about which of these you want in a given case.
*
* The libdbus full version number is "MAJOR.MINOR.MICRO" where the
* MINOR changes if API is added, and the MICRO changes with each
* release of a MAJOR.MINOR series. The MINOR is an odd number for
* development releases and an even number for stable releases.
*
* @param major_version_p pointer to return the major version, or
#NULL
* @param minor_version_p pointer to return the minor version, or
#NULL
* @param micro_version_p pointer to return the micro version, or
#NULL
*
*/
void
dbus_get_version (int *major_version_p,
                 int *minor_version_p,
                 int *micro_version_p)
{
    if (major_version_p)
        *major_version_p = DBUS_MAJOR_VERSION;
    if (minor_version_p)
        *minor_version_p = DBUS_MINOR_VERSION;
    if (micro_version_p)
        *micro_version_p = DBUS_MICRO_VERSION;
}

/** @} */ /* End of public API */

#ifdef DBUS_BUILD_TESTS

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include "dbus-test.h"
#include <stdlib.h>

dbus_bool_t
_dbus_misc_test (void)
{
    int major, minor, micro;
    DBusString str;

    /* make sure we don't crash on NULL */
    dbus_get_version (NULL, NULL, NULL);

    /* Now verify that all the compile-time version stuff
     * is right and matches the runtime. These tests
     * are mostly intended to catch various kinds of
     * typo (mixing up major and minor, that sort of thing).

```

```

    */
    dbus_get_version (&major, &minor, &micro);

    _dbus_assert (major == DBUS_MAJOR_VERSION);
    _dbus_assert (minor == DBUS_MINOR_VERSION);
    _dbus_assert (micro == DBUS_MICRO_VERSION);

#define MAKE_VERSION(x, y, z) (((x) << 16) | ((y) << 8) | (z))

    /* check that MAKE_VERSION works and produces the intended ordering
    */
    _dbus_assert (MAKE_VERSION (1, 0, 0) > MAKE_VERSION (0, 0, 0));
    _dbus_assert (MAKE_VERSION (1, 1, 0) > MAKE_VERSION (1, 0, 0));
    _dbus_assert (MAKE_VERSION (1, 1, 1) > MAKE_VERSION (1, 1, 0));

    _dbus_assert (MAKE_VERSION (2, 0, 0) > MAKE_VERSION (1, 1, 1));
    _dbus_assert (MAKE_VERSION (2, 1, 0) > MAKE_VERSION (1, 1, 1));
    _dbus_assert (MAKE_VERSION (2, 1, 1) > MAKE_VERSION (1, 1, 1));

    /* check DBUS_VERSION */
    _dbus_assert (MAKE_VERSION (major, minor, micro) == DBUS_VERSION);

    /* check that ordering works with DBUS_VERSION */
    _dbus_assert (MAKE_VERSION (major - 1, minor, micro) <
DBUS_VERSION);
    _dbus_assert (MAKE_VERSION (major, minor - 1, micro) <
DBUS_VERSION);
    _dbus_assert (MAKE_VERSION (major, minor, micro - 1) <
DBUS_VERSION);

    _dbus_assert (MAKE_VERSION (major + 1, minor, micro) >
DBUS_VERSION);
    _dbus_assert (MAKE_VERSION (major, minor + 1, micro) >
DBUS_VERSION);
    _dbus_assert (MAKE_VERSION (major, minor, micro + 1) >
DBUS_VERSION);

    /* Check DBUS_VERSION_STRING */

    if (!_dbus_string_init (&str))
        _dbus_assert_not_reached ("no memory");

    if (!(_dbus_string_append_int (&str, major) &&
        _dbus_string_append_byte (&str, '.') &&
        _dbus_string_append_int (&str, minor) &&
        _dbus_string_append_byte (&str, '.') &&
        _dbus_string_append_int (&str, micro)))
        _dbus_assert_not_reached ("no memory");

    _dbus_assert (_dbus_string_equal_c_str (&str, DBUS_VERSION_STRING));

    _dbus_string_free (&str);

```

```

    return TRUE;
}

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif

File = dbus-misc.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-misc.h A few assorted public functions that don't fit
elsewhere
*
* Copyright (C) 2006 Red Hat, Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_MISC_H
#define DBUS_MISC_H

#include <dbus/dbus-types.h>
#include <dbus/dbus-errors.h>

DBUS_BEGIN_DECLS

/**

```

```

* @addtogroup DBusMisc
* @{
*/
DBUS_EXPORT
char*      dbus_get_local_machine_id  (void);

DBUS_EXPORT
void       dbus_get_version           (int *major_version_p,
                                       int *minor_version_p,
                                       int *micro_version_p);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_MISC_H */

```

File = dbus-monitor.1

```

.\"
.\" dbus\--monitor manual page.
.\" Copyright (C) 2003 Red Hat, Inc.
.\"
.TH dbus\--monitor 1
.SH NAME
dbus\--monitor \-- debug probe to print message bus messages
.SH SYNOPSIS
.PP
.B dbus\--monitor
[\--system | \--session | \--address ADDRESS] [\--profile | \--
monitor]
[watch expressions]

.SH DESCRIPTION

The \fIdbus\--monitor\fP command is used to monitor messages going
through a D\--Bus message bus. See
http://www.freedesktop.org/software/dbus/ for more information about
the big picture.

.PP
There are two well\--known message buses: the systemwide message bus
(installed on many systems as the "messagebus" service) and the
per\--user\--login\--session message bus (started each time a user logs
in).
The \--system and \--session options direct \fIdbus\--monitor\fP to
monitor the system or session buses respectively. If neither is
specified, \fIdbus\--monitor\fP monitors the session bus.

```

.PP
\fIdbus\monitor\fP has two different output modes, the 'classic'\-
style
monitoring mode and profiling mode. The profiling format is a compact
format with a single line per message and microsecond\resolution
timing
information. The \-\-profile and \-\-monitor options select the
profiling
and monitoring output format respectively. If neither is specified,
\fIdbus\monitor\fP uses the monitoring output format.

.PP
In order to get \fIdbus\monitor\fP to see the messages you are
interested
in, you should specify a set of watch expressions as you would expect
to
be passed to the \fIdbus_bus_add_match\fP function.

.PP
The message bus configuration may keep \fIdbus\monitor\fP from seeing
all messages, especially if you run the monitor as a non\root user.

.SH OPTIONS

.TP

.I "\-\-system"

Monitor the system message bus.

.TP

.I "\-\-session"

Monitor the session message bus. (This is the default.)

.TP

.I "\-\-address ADDRESS"

Monitor an arbitrary message bus given at ADDRESS.

.TP

.I "\-\-profile"

Use the profiling output format.

.TP

.I "\-\-monitor"

Use the monitoring output format. (This is the default.)

.SH EXAMPLE

Here is an example of using dbus\monitor to watch for the gnome
typing

monitor to say things

.nf

```
dbus\monitor  
"type='signal',sender='org.gnome.TypingMonitor',interface='org.gnome.T  
ypingMonitor'"
```

.fi

.SH AUTHOR

dbus\-monitor was written by Philip Blundell.
The profiling output mode was added by Olli Salli.

.SH BUGS

Please send bug reports to the D\ -Bus mailing list or bug tracker,
see <http://www.freedesktop.org/software/dbus/>

File = dbus-monitor.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://www.oasis-
open.org/docbook/xml/4.1.2/docbookx.dtd">
<!-- lifted from troff+man by doclifter -->
<refentry id='dbusmonitor1'>
<!-- -->
<!-- dbus\-monitor manual page. -->
<!-- Copyright (C) 2003 Red Hat, Inc. -->

<refmeta>
<refentrytitle>dbus-monitor</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>
<refnamediv id='name'>
<refname>dbus-monitor</refname>
<refpurpose>debug probe to print message bus messages</refpurpose>
</refnamediv>
<!-- body begins here -->
<refsynopsisdiv id='synopsis'>
<cmdsynopsis>
  <command>dbus-monitor</command>
  <group choice='opt'><arg choice='plain'>--system </arg><arg
choice='plain'>--session </arg><arg choice='plain'>--address
<replaceable>ADDRESS</replaceable></arg></group>
  <group choice='opt'><arg choice='plain'>--profile </arg><arg
choice='plain'>--monitor </arg></group>
  <arg choice='opt'><arg
choice='plain'><replaceable>watch</replaceable></arg><arg
choice='plain'><replaceable>expressions</replaceable></arg></arg>
  <sbr/>
</cmdsynopsis>
</refsynopsisdiv>

<refsect1 id='description'><title>DESCRIPTION</title>
<para>The <command>dbus-monitor</command> command is used to monitor
messages going
through a D-Bus message bus. See
```

<http://www.freedesktop.org/software/dbus/> for more information about the big picture.

There are two well-known message buses: the systemwide message bus (installed on many systems as the "messagebus" service) and the per-user-login-session message bus (started each time a user logs in). The `--system` and `--session` options direct `dbus-monitor` to monitor the system or session buses respectively. If neither is specified, `dbus-monitor` monitors the session bus.

`dbus-monitor` has two different output modes, the 'classic'-style monitoring mode and profiling mode. The profiling format is a compact format with a single line per message and microsecond-resolution timing information. The `--profile` and `--monitor` options select the profiling and monitoring output format respectively. If neither is specified, `dbus-monitor` uses the monitoring output format.

In order to get `dbus-monitor` to see the messages you are interested in, you should specify a set of watch expressions as you would expect to be passed to the `dbus_bus_add_match` function.

The message bus configuration may keep `dbus-monitor` from seeing all messages, especially if you run the monitor as a non-root user.

OPTIONS

`--system`

Monitor the system message bus.

```
<term><option>--session</option></term>
<listitem>
<para>Monitor the session message bus. (This is the default.)</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--address ADDRESS</option></term>
<listitem>
<para>Monitor an arbitrary message bus given at ADDRESS.</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--profile</option></term>
<listitem>
<para>Use the profiling output format.</para>
</listitem>
</varlistentry>
<varlistentry>
<term><option>--monitor</option></term>
<listitem>
<para>Use the monitoring output format. (This is the default.)</para>
</listitem>
</varlistentry>
</variablelist>
</refsect1>
```

```
<refsect1 id='example'><title>EXAMPLE</title>
<para>Here is an example of using dbus-monitor to watch for the gnome
typing
monitor to say things</para>
<literallayout remap='.nf'>
```

```
dbus-monitor
"type='signal',sender='org.gnome.TypingMonitor',interface='org.gnome.T
ypingMonitor'"
```

```
</literallayout> <!-- .fi -->
```

```
</refsect1>
```

```
<refsect1 id='author'><title>AUTHOR</title>
<para>dbus-monitor was written by Philip Blundell.
The profiling output mode was added by Olli Salli.</para>
```

```
</refsect1>
```

```
<refsect1 id='bugs'><title>BUGS</title>
<para>Please send bug reports to the D-Bus mailing list or bug
tracker,
```



```
see <ulink
url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop
.org/software/dbus/</ulink></para>
</refsect1>
</refentry>
```

File = dbus-nonce.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-nonce.c Nonce handling functions used by nonce-tcp (internal
to D-Bus implementation)
*
* Copyright (C) 2009 Klaralvdalens Datakonsult AB, a KDAB Group
company, info@kdab.net
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-
1307 USA
*
*/

#include <config.h>
// major sections of this file are modified code from libassuan, (C)
FSF
#include "dbus-nonce.h"
#include "dbus-internals.h"
#include "dbus-protocol.h"
#include "dbus-sysdeps.h"

#include <stdio.h>

static dbus_bool_t
do_check_nonce (int fd, const DBusString *nonce, DBusError *error)
{
```

```

DBusString buffer;
DBusString p;
size_t nleft;
dbus_bool_t result;
int n;

_DBUS_ASSERT_ERROR_IS_CLEAR (error);

nleft = 16;

if ( !_dbus_string_init (&buffer)
    || !_dbus_string_init (&p) ) {
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&p);
    _dbus_string_free (&buffer);
    return FALSE;
}

while (nleft)
{
    n = _dbus_read_socket (fd, &p, nleft);
    if (n == -1 && _dbus_get_is_errno_eintr())
        ;
    else if (n == -1 && _dbus_get_is_errno_eagain_or_ewouldblock())
        _dbus_sleep_milliseconds (100);
    else if (n == -1)
    {
        dbus_set_error (error, DBUS_ERROR_IO_ERROR, "Could not read
nonce from socket (fd=%d)", fd );
        _dbus_string_free (&p);
        _dbus_string_free (&buffer);
        return FALSE;
    }
    else if (!n)
    {
        _dbus_string_free (&p);
        _dbus_string_free (&buffer);
        dbus_set_error (error, DBUS_ERROR_IO_ERROR, "Could not read
nonce from socket (fd=%d)", fd );
        return FALSE;
    }
    else
    {
        _dbus_string_append_len(&buffer, _dbus_string_get_const_data
(&p), n);
        nleft -= n;
    }
}

result = _dbus_string_equal_len (&buffer, nonce, 16);
if (!result)

```

```

    dbus_set_error (error, DBUS_ERROR_ACCESS_DENIED, "Nonces do not
match, access denied (fd=%d)", fd );

    _dbus_string_free (&p);
    _dbus_string_free (&buffer);

    return result;
}

/**
 * reads the nonce from the nonce file and stores it in a string
 *
 * @param fname the file to read the nonce from
 * @param nonce returns the nonce. Must be an initialized string, the
nonce will be appended.
 * @param error error object to report possible errors
 * @return FALSE iff reading the nonce fails (error is set then)
 */
dbus_bool_t
_dbus_read_nonce (const DBusString *fname, DBusString *nonce,
DBusError* error)
{
    FILE *fp;
    char buffer[17];
    size_t nread;

    buffer[sizeof buffer - 1] = '\0';

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_verbose ("reading nonce from file: %s\n",
_dbus_string_get_const_data (fname));

    fp = fopen (_dbus_string_get_const_data (fname), "rb");
    if (!fp)
        return FALSE;
    nread = fread (buffer, 1, sizeof buffer - 1, fp);
    fclose (fp);
    if (!nread)
    {
        dbus_set_error (error, DBUS_ERROR_FILE_NOT_FOUND, "Could not
read nonce from file %s", _dbus_string_get_const_data (fname));
        return FALSE;
    }

    if (!_dbus_string_append_len (nonce, buffer, sizeof buffer - 1 ))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return FALSE;
    }
    return TRUE;
}

```

```

}

int
_dbus_accept_with_noncefile (int listen_fd, const DBusNonceFile
*noncefile)
{
    int fd;
    DBusString nonce;

    _dbus_assert (noncefile != NULL);
    if (!_dbus_string_init (&nonce))
        return -1;
    //PENDING(kdab): set better errors
    if (_dbus_read_nonce (_dbus_noncefile_get_path(noncefile), &nonce,
NULL) != TRUE)
        return -1;
    fd = _dbus_accept (listen_fd);
    if (_dbus_socket_is_invalid (fd))
        return fd;
    if (do_check_nonce(fd, &nonce, NULL) != TRUE) {
        _dbus_verbose ("nonce check failed. Closing socket.\n");
        _dbus_close_socket(fd, NULL);
        return -1;
    }

    return fd;
}

static dbus_bool_t
generate_and_write_nonce (const DBusString *filename, DBusError
*error)
{
    DBusString nonce;
    dbus_bool_t ret;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (!_dbus_string_init (&nonce))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return FALSE;
    }

    if (!_dbus_generate_random_bytes (&nonce, 16))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        _dbus_string_free (&nonce);
        return FALSE;
    }

    ret = _dbus_string_save_to_file (&nonce, filename, FALSE, error);

```

```

    _dbus_string_free (&nonce);

    return ret;
}

/**
 * sends the nonce over a given socket. Blocks while doing so.
 *
 * @param fd the file descriptor to write the nonce data to (usually a
socket)
 * @param noncefile the noncefile location to read the nonce from
 * @param error contains error details if FALSE is returned
 * @return TRUE iff the nonce was successfully sent. Note that this
does not
 * indicate whether the server accepted the nonce.
 */
dbus_bool_t
_dbus_send_nonce (int fd, const DBusString *noncefile, DBusError
*error)
{
    dbus_bool_t read_result;
    int send_result;
    DBusString nonce;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (_dbus_string_get_length (noncefile) == 0)
        return FALSE;

    if (!_dbus_string_init (&nonce))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return FALSE;
    }

    read_result = _dbus_read_nonce (noncefile, &nonce, error);
    if (!read_result)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        _dbus_string_free (&nonce);
        return FALSE;
    }
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    send_result = _dbus_write_socket (fd, &nonce, 0,
_dbus_string_get_length (&nonce));

    _dbus_string_free (&nonce);

    if (send_result == -1)
    {
        dbus_set_error (error,

```

```

        _dbus_error_from_system_errno (),
        "Failed to send nonce (fd=%d): %s",
        fd, _dbus_strerror_from_errno ());
    return FALSE;
}

return TRUE;
}

static dbus_bool_t
do_noncefile_create (DBusNonceFile *noncefile,
                    DBusError *error,
                    dbus_bool_t use_subdir)
{
    DBusString randomStr;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_assert (noncefile);

    if (!_dbus_string_init (&randomStr))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto on_error;
    }

    if (!_dbus_generate_random_ascii (&randomStr, 8))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto on_error;
    }

    if (!_dbus_string_init (&noncefile->dir)
        || !_dbus_string_append (&noncefile->dir, _dbus_get_tmpdir()))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto on_error;
    }
    if (use_subdir)
    {
        if (!_dbus_string_append (&noncefile->dir, "/dbus_nonce-")
            || !_dbus_string_append (&noncefile->dir,
                _dbus_string_get_const_data (&randomStr)) )
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            goto on_error;
        }
        if (!_dbus_string_init (&noncefile->path)
            || !_dbus_string_copy (&noncefile->dir, 0, &noncefile-
                >path, 0)
            || !_dbus_string_append (&noncefile->path, "/nonce"))
        {

```

```

        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto on_error;
    }
    if (!_dbus_create_directory (&noncefile->dir, error))
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto on_error;
    }
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

}
else
{
    if (!_dbus_string_init (&noncefile->path)
        || !_dbus_string_copy (&noncefile->dir, 0, &noncefile-
>path, 0)
        || !_dbus_string_append (&noncefile->path, "/dbus_nonce-")
        || !_dbus_string_append (&noncefile->path,
_dbus_string_get_const_data (&randomStr))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto on_error;
    }

}

if (!generate_and_write_nonce (&noncefile->path, error))
{
    _DBUS_ASSERT_ERROR_IS_SET (error);
    if (use_subdir)
        _dbus_delete_directory (&noncefile->dir, NULL); //we ignore
possible errors deleting the dir and return the write error instead
        goto on_error;
}
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_string_free (&randomStr);

    return TRUE;
on_error:
    if (use_subdir)
        _dbus_delete_directory (&noncefile->dir, NULL);
    _dbus_string_free (&noncefile->dir);
    _dbus_string_free (&noncefile->path);
    _dbus_string_free (&randomStr);
    return FALSE;
}

#ifdef DBUS_WIN
/**
 * creates a nonce file in a user-readable location and writes a
generated nonce to it

```

```

*
* @param noncefile returns the nonce file location
* @param error error details if creating the nonce file fails
* @return TRUE iff the nonce file was successfully created
*/
dbus_bool_t
_dbus_noncefile_create (DBusNonceFile *noncefile,
                        DBusError *error)
{
    return do_noncefile_create (noncefile, error,
/*use_subdir=*/FALSE);
}

/**
* deletes the noncefile and frees the DBusNonceFile object.
*
* @param noncefile the nonce file to delete. Contents will be freed.
* @param error error details if the nonce file could not be deleted
* @return TRUE
*/
dbus_bool_t
_dbus_noncefile_delete (DBusNonceFile *noncefile,
                        DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_delete_file (&noncefile->path, error);
    _dbus_string_free (&noncefile->dir);
    _dbus_string_free (&noncefile->path);
    return TRUE;
}

#else
/**
* creates a nonce file in a user-readable location and writes a
generated nonce to it.
* Initializes the noncefile object.
*
* @param noncefile returns the nonce file location
* @param error error details if creating the nonce file fails
* @return TRUE iff the nonce file was successfully created
*/
dbus_bool_t
_dbus_noncefile_create (DBusNonceFile *noncefile,
                        DBusError *error)
{
    return do_noncefile_create (noncefile, error,
/*use_subdir=*/TRUE);
}

/**
* deletes the noncefile and frees the DBusNonceFile object.

```



```

*
* @param noncefile the nonce file to delete. Contents will be freed.
* @param error error details if the nonce file could not be deleted
* @return TRUE
*/
dbus_bool_t
_dbus_noncefile_delete (DBusNonceFile *noncefile,
                        DBusError *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    _dbus_delete_directory (&noncefile->dir, error);
    _dbus_string_free (&noncefile->dir);
    _dbus_string_free (&noncefile->path);
    return TRUE;
}
#endif

/**
 * returns the absolute file path of the nonce file
 *
 * @param noncefile an initialized noncefile object
 * @return the absolute path of the nonce file
 */
const DBusString*
_dbus_noncefile_get_path (const DBusNonceFile *noncefile)
{
    _dbus_assert (noncefile);
    return &noncefile->path;
}

/**
 * reads data from a file descriptor and checks if the received data
matches
 * the data in the given noncefile.
 *
 * @param fd the file descriptor to read the nonce from
 * @param noncefile the nonce file to check the received data against
 * @param error error details on fail
 * @return TRUE iff a nonce could be successfully read from the file
descriptor
 * and matches the nonce from the given nonce file
 */
dbus_bool_t
_dbus_noncefile_check_nonce (int fd,
                             const DBusNonceFile *noncefile,
                             DBusError* error)
{
    return do_check_nonce (fd, _dbus_noncefile_get_path (noncefile),
error);
}

```

```
/** @} end of nonce */
```

```
File = dbus-nonce.h
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* dbus-nonce.h Nonce handling functions used by nonce-tcp (internal  
to D-Bus implementation)
```

```
 *  
 * Copyright (C) 2009 Klaralvdalens Datakonsult AB, a KDAB Group  
company, info@kdab.net  
 *  
 * Licensed under the Academic Free License version 2.1  
 *  
 * This program is free software; you can redistribute it and/or  
modify  
 * it under the terms of the GNU General Public License as published  
by  
 * the Free Software Foundation; either version 2 of the License, or  
 * (at your option) any later version.  
 *  
 * This program is distributed in the hope that it will be useful,  
 * but WITHOUT ANY WARRANTY; without even the implied warranty of  
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
 * GNU General Public License for more details.  
 *  
 * You should have received a copy of the GNU General Public License  
 * along with this program; if not, write to the Free Software  
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-  
1307 USA
```

```
 *  
 */  
#ifndef DBUS_NONCE_H  
#define DBUS_NONCE_H  
  
#include <dbus/dbus-macros.h>  
#include <dbus/dbus-types.h>  
#include <dbus/dbus-errors.h>  
#include <dbus/dbus-string.h>
```

```
DBUS_BEGIN_DECLS
```

```
typedef struct DBusNonceFile DBusNonceFile;
```

```
struct DBusNonceFile  
{  
    DBusString path;  
    DBusString dir;  
};
```

```

// server

dbus_bool_t _dbus_noncefile_create (DBusNonceFile *noncefile,
                                   DBusError *error);

dbus_bool_t _dbus_noncefile_delete (DBusNonceFile *noncefile,
                                   DBusError *error);

dbus_bool_t _dbus_noncefile_check_nonce (int fd,
                                       const DBusNonceFile
*noncefile,
                                       DBusError *error);

const DBusString* _dbus_noncefile_get_path (const DBusNonceFile
*noncefile);

int _dbus_accept_with_noncefile (int listen_fd,
                                const DBusNonceFile *noncefile);

// shared

dbus_bool_t _dbus_read_nonce (const DBusString *fname,
                              DBusString *nonce,
                              DBusError *error);

// client

dbus_bool_t _dbus_send_nonce (int fd,
                              const DBusString *noncefile,
                              DBusError *error);

DBUS_END_DECLS

#endif /* DBUS_NONCE_H */

File = dbus-object-tree.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-object-tree.c DBusObjectTree (internals of DBusConnection)
 *
 * Copyright (C) 2003, 2005 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or

```

```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#include "dbus-object-tree.h"
#include "dbus-connection-internal.h"
#include "dbus-internals.h"
#include "dbus-hash.h"
#include "dbus-protocol.h"
#include "dbus-string.h"
#include <string.h>
#include <stdlib.h>

/**
 * @defgroup DBusObjectTree A hierarchy of objects with container-
contained relationship
 * @ingroup DBusInternals
 * @brief DBusObjectTree is used by DBusConnection to track the object
tree
 *
 * Types and functions related to DBusObjectTree. These
 * are all library-internal.
 *
 * @{
 */

/** Subnode of the object hierarchy */
typedef struct DBusObjectSubtree DBusObjectSubtree;

static DBusObjectSubtree* _dbus_object_subtree_new (const char
*name,
                                                    const
DBusObjectPathVTable *vtable,
                                                    void
*user_data);
static DBusObjectSubtree* _dbus_object_subtree_ref
(DBusObjectSubtree *subtree);
static void _dbus_object_subtree_unref
(DBusObjectSubtree *subtree);

/**

```

```

* Internals of DBusObjectTree
*/
struct DBusObjectTree
{
    int                refcount;    /**< Reference count */
    DBusConnection    *connection; /**< Connection this tree belongs to
*/

    DBusObjectSubtree *root;       /**< Root of the tree ("/" node) */
};

/**
 * Struct representing a single registered subtree handler, or node
 * that's a parent of a registered subtree handler. If
 * message_function != NULL there's actually a handler at this node.
 */
struct DBusObjectSubtree
{
    DBusAtomic                refcount;    /**<
Reference count */
    DBusObjectSubtree        *parent;     /**< Parent
node */
    DBusObjectPathUnregisterFunction unregister_function; /**<
Function to call on unregister */
    DBusObjectPathMessageFunction message_function;    /**<
Function to handle messages */
    void                      *user_data;    /**< Data
for functions */
    DBusObjectSubtree        **subtrees;    /**< Child
nodes */
    int                      n_subtrees;    /**< Number
of child nodes */
    int                      max_subtrees;    /**< Number
of allocated entries in subtrees */
    unsigned int             invoke_as_fallback : 1; /**<
Whether to invoke message_function when child nodes don't handle the
message */
    char                     name[1]; /**< Allocated as large
as necessary */
};

/**
 * Creates a new object tree, representing a mapping from paths
 * to handler vtables.
 *
 * @param connection the connection this tree belongs to
 * @returns the new tree or #NULL if no memory
 */
DBusObjectTree*
_dbus_object_tree_new (DBusConnection *connection)
{
    DBusObjectTree *tree;

```

```

/* the connection passed in here isn't fully constructed,
 * so don't do anything more than store a pointer to
 * it
 */

tree = dbus_new0 (DBusObjectTree, 1);
if (tree == NULL)
    goto oom;

tree->refcount = 1;
tree->connection = connection;
tree->root = _dbus_object_subtree_new ("/", NULL, NULL);
if (tree->root == NULL)
    goto oom;
tree->root->invoke_as_fallback = TRUE;

return tree;

oom:
if (tree)
    {
        dbus_free (tree);
    }

return NULL;
}

/**
 * Increment the reference count
 * @param tree the object tree
 * @returns the object tree
 */
DBusObjectTree *
_dbus_object_tree_ref (DBusObjectTree *tree)
{
    _dbus_assert (tree->refcount > 0);

    tree->refcount += 1;

    return tree;
}

/**
 * Decrement the reference count
 * @param tree the object tree
 */
void
_dbus_object_tree_unref (DBusObjectTree *tree)
{
    _dbus_assert (tree->refcount > 0);

```

```

tree->refcount -= 1;

if (tree->refcount == 0)
{
    _dbus_object_tree_free_all_unlocked (tree);

    dbus_free (tree);
}
}

/** Set to 1 to get a bunch of debug spew about finding the
 * subtree nodes
 */
#define VERBOSE_FIND 0

static DBusObjectSubtree*
find_subtree_recurse (DBusObjectSubtree *subtree,
                     const char **path,
                     dbus_bool_t create_if_not_found,
                     int *index_in_parent,
                     dbus_bool_t *exact_match)
{
    int i, j;
    dbus_bool_t return_deepest_match;

    return_deepest_match = exact_match != NULL;

    _dbus_assert (!(return_deepest_match && create_if_not_found));

    if (path[0] == NULL)
    {
        #if VERBOSE_FIND
            _dbus_verbose (" path exhausted, returning %s\n",
                          subtree->name);
        #endif
        if (exact_match != NULL)
            *exact_match = TRUE;
        return subtree;
    }

    #if VERBOSE_FIND
        _dbus_verbose (" searching children of %s for %s\n",
                      subtree->name, path[0]);
    #endif

    i = 0;
    j = subtree->n_subtrees;
    while (i < j)
    {
        int k, v;

        k = (i + j) / 2;

```

```

        v = strcmp (path[0], subtree->subtrees[k]->name);

#ifdef VERBOSE_FIND
        _dbus_verbose (" %s cmp %s = %d\n",
                      path[0], subtree->subtrees[k]->name,
                      v);
#endif

        if (v == 0)
            {
                if (index_in_parent)
                    {
#ifdef VERBOSE_FIND
                        _dbus_verbose (" storing parent index %d\n", k);
#endif
                        *index_in_parent = k;
                    }

                if (return_deepest_match)
                    {
                        DBusObjectSubtree *next;

                        next = find_subtree_recurse (subtree->subtrees[k],
                                                  &path[1],
                                                  create_if_not_found,
                                                  index_in_parent,
                                                  exact_match);

                        if (next == NULL &&
                            subtree->invoke_as_fallback)
                            {
#ifdef VERBOSE_FIND
                                _dbus_verbose (" no deeper match found, returning
%s\n",
                                              subtree->name);
#endif
                            }
                        if (exact_match != NULL)
                            *exact_match = FALSE;
                        return subtree;
                    }
                else
                    return next;
            }
        else
            return find_subtree_recurse (subtree->subtrees[k],
                                        &path[1],
                                        create_if_not_found,
                                        index_in_parent,
                                        exact_match);
    }
    else if (v < 0)
        {
            j = k;

```



```

        }
    else
    {
        i = k + 1;
    }
}

#ifdef VERBOSE_FIND
    _dbus_verbose (" no match found, current tree %s,
create_if_not_found = %d\n",
                  subtree->name, create_if_not_found);
#endif

    if (create_if_not_found)
    {
        DBusObjectSubtree* child;
        int child_pos, new_n_subtrees;

#ifdef VERBOSE_FIND
        _dbus_verbose (" creating subtree %s\n",
                      path[0]);
#endif

        child = _dbus_object_subtree_new (path[0],
                                         NULL, NULL);

        if (child == NULL)
            return NULL;

        new_n_subtrees = subtree->n_subtrees + 1;
        if (new_n_subtrees > subtree->max_subtrees)
        {
            int new_max_subtrees;
            DBusObjectSubtree **new_subtrees;

            new_max_subtrees = subtree->max_subtrees == 0 ? 1 : 2 *
subtree->max_subtrees;
            new_subtrees = dbus_realloc (subtree->subtrees,
                                       new_max_subtrees * sizeof
(DBusObjectSubtree*));
            if (new_subtrees == NULL)
            {
                _dbus_object_subtree_unref (child);
                return NULL;
            }
            subtree->subtrees = new_subtrees;
            subtree->max_subtrees = new_max_subtrees;
        }

        /* The binary search failed, so i == j points to the
           place the child should be inserted. */
        child_pos = i;
        _dbus_assert (child_pos < new_n_subtrees &&

```

```

        new_n_subtrees <= subtree->max_subtrees);
if (child_pos + 1 < new_n_subtrees)
{
    memmove (&subtree->subtrees[child_pos+1],
            &subtree->subtrees[child_pos],
            (new_n_subtrees - child_pos - 1) *
            sizeof subtree->subtrees[0]);
}
subtree->subtrees[child_pos] = child;

if (index_in_parent)
    *index_in_parent = child_pos;
subtree->n_subtrees = new_n_subtrees;
child->parent = subtree;

return find_subtree_recurse (child,
                            &path[1], create_if_not_found,
                            index_in_parent, exact_match);
}
else
{
    if (exact_match != NULL)
        *exact_match = FALSE;
    return (return_deepest_match && subtree->invoke_as_fallback) ?
subtree : NULL;
}
}

static DBusObjectSubtree*
find_subtree (DBusObjectTree *tree,
             const char    **path,
             int           *index_in_parent)
{
    DBusObjectSubtree *subtree;

#ifdef VERBOSE_FIND
    _dbus_verbose ("Looking for exact registered subtree\n");
#endif

    subtree = find_subtree_recurse (tree->root, path, FALSE,
index_in_parent, NULL);

    if (subtree && subtree->message_function == NULL)
        return NULL;
    else
        return subtree;
}

static DBusObjectSubtree*
lookup_subtree (DBusObjectTree *tree,
              const char    **path)
{

```



```

        void                *user_data,
        DBusError           *error)
{
    DBusObjectSubtree *subtree;

    _dbus_assert (tree != NULL);
    _dbus_assert (vtable->message_function != NULL);
    _dbus_assert (path != NULL);

    subtree = ensure_subtree (tree, path);
    if (subtree == NULL)
    {
        _DBUS_SET_OOM (error);
        return FALSE;
    }

    if (subtree->message_function != NULL)
    {
        if (error != NULL)
        {
            char *complete_path = flatten_path (path);

            dbus_set_error (error, DBUS_ERROR_OBJECT_PATH_IN_USE,
                "A handler is already registered for %s",
                complete_path ? complete_path
                : "(cannot represent path: out
of memory!)");

            dbus_free (complete_path);
        }

        return FALSE;
    }

    subtree->message_function = vtable->message_function;
    subtree->unregister_function = vtable->unregister_function;
    subtree->user_data = user_data;
    subtree->invoke_as_fallback = fallback != FALSE;

    return TRUE;
}

/**
 * Unregisters an object subtree that was registered with the
 * same path.
 *
 * @param tree the global object tree
 * @param path path to the subtree (same as the one passed to
_dbus_object_tree_register())
 */
void

```

```

_dbus_object_tree_unregister_and_unlock (DBusObjectTree
*tree,
                                     const char
**path)
{
    int i;
    DBusObjectSubtree *subtree;
    DBusObjectPathUnregisterFunction unregister_function;
    void *user_data;
    DBusConnection *connection;

    _dbus_assert (path != NULL);

    unregister_function = NULL;
    user_data = NULL;

    subtree = find_subtree (tree, path, &i);

#ifdef DBUS_DISABLE_CHECKS
    if (subtree == NULL)
    {
        _dbus_warn ("Attempted to unregister path (path[0] = %s path[1]
= %s) which isn't registered\n",
                    path[0] ? path[0] : "null",
                    path[1] ? path[1] : "null");
        goto unlock;
    }
#else
    _dbus_assert (subtree != NULL);
#endif

    _dbus_assert (subtree->parent == NULL ||
                  (i >= 0 && subtree->parent->subtrees[i] == subtree));

    subtree->message_function = NULL;

    unregister_function = subtree->unregister_function;
    user_data = subtree->user_data;

    subtree->unregister_function = NULL;
    subtree->user_data = NULL;

    /* If we have no subtrees of our own, remove from
     * our parent (FIXME could also be more aggressive
     * and remove our parent if it becomes empty)
     */
    if (subtree->parent && subtree->n_subtrees == 0)
    {
        /* assumes a 0-byte memmove is OK */
        memmove (&subtree->parent->subtrees[i],
                &subtree->parent->subtrees[i+1],
                (subtree->parent->n_subtrees - i - 1) *

```

```

        sizeof (subtree->parent->subtrees[0]));
    subtree->parent->n_subtrees -= 1;

    subtree->parent = NULL;

    _dbus_object_subtree_unref (subtree);
}
subtree = NULL;

unlock:
    connection = tree->connection;

    /* Unlock and call application code */
#ifdef DBUS_BUILD_TESTS
    if (connection)
#endif
    {
        _dbus_connection_ref_unlocked (connection);
        _dbus_verbose ("unlock\n");
        _dbus_connection_unlock (connection);
    }

    if (unregister_function)
        (* unregister_function) (connection, user_data);

#ifdef DBUS_BUILD_TESTS
    if (connection)
#endif
    dbus_connection_unref (connection);
}

static void
free_subtree_recurse (DBusConnection *connection,
                     DBusObjectSubtree *subtree)
{
    /* Delete them from the end, for slightly
     * more robustness against odd reentrancy.
     */
    while (subtree->n_subtrees > 0)
    {
        DBusObjectSubtree *child;

        child = subtree->subtrees[subtree->n_subtrees - 1];
        subtree->subtrees[subtree->n_subtrees - 1] = NULL;
        subtree->n_subtrees -= 1;
        child->parent = NULL;

        free_subtree_recurse (connection, child);
    }

    /* Call application code */
    if (subtree->unregister_function)

```

```

        (* subtree->unregister_function) (connection,
                                         subtree->user_data);

subtree->message_function = NULL;
subtree->unregister_function = NULL;
subtree->user_data = NULL;

/* Now free ourselves */
_dbus_object_subtree_unref (subtree);
}

/**
 * Free all the handlers in the tree. Lock on tree's connection
 * must not be held.
 *
 * @param tree the object tree
 */
void
_dbus_object_tree_free_all_unlocked (DBusObjectTree *tree)
{
    if (tree->root)
        free_subtree_recurse (tree->connection,
                              tree->root);

    tree->root = NULL;
}

static dbus_bool_t
_dbus_object_tree_list_registered_unlocked (DBusObjectTree *tree,
                                           const char
**parent_path,
                                           char
***child_entries)
{
    DBusObjectSubtree *subtree;
    char **retval;

    _dbus_assert (parent_path != NULL);
    _dbus_assert (child_entries != NULL);

    *child_entries = NULL;

    subtree = lookup_subtree (tree, parent_path);
    if (subtree == NULL)
    {
        retval = dbus_new0 (char *, 1);
    }
    else
    {
        int i;
        retval = dbus_new0 (char*, subtree->n_subtrees + 1);
        if (retval == NULL)
            goto out;
    }
}

```

```

    i = 0;
    while (i < subtree->n_subtrees)
    {
        retval[i] = _dbus_strdup (subtree->subtrees[i]->name);
        if (retval[i] == NULL)
        {
            dbus_free_string_array (retval);
            retval = NULL;
            goto out;
        }
        ++i;
    }
}

out:

    *child_entries = retval;
    return retval != NULL;
}

static DBusHandlerResult
handle_default_introspect_and_unlock (DBusObjectTree      *tree,
                                      DBusMessage          *message,
                                      const char            **path)
{
    DBusString xml;
    DBusHandlerResult result;
    char **children;
    int i;
    DBusMessage *reply;
    DBusMessageIter iter;
    const char *v_STRING;
    dbus_bool_t already_unlocked;

    /* We have the connection lock here */

    already_unlocked = FALSE;

    _dbus_verbose (" considering default Introspect() handler...\n");

    reply = NULL;

    if (!dbus_message_is_method_call (message,
                                      DBUS_INTERFACE_INTROSPECTABLE,
                                      "Introspect"))
    {
#ifdef DBUS_BUILD_TESTS
        if (tree->connection)
#endif
        {
            _dbus_verbose ("unlock\n");

```



```

        _dbus_connection_unlock (tree->connection);
    }

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}

_dbus_verbose (" using default Introspect() handler!\n");

if (!_dbus_string_init (&xml))
{
#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        _dbus_verbose ("unlock\n");
        _dbus_connection_unlock (tree->connection);
    }

    return DBUS_HANDLER_RESULT_NEED_MEMORY;
}

result = DBUS_HANDLER_RESULT_NEED_MEMORY;

children = NULL;
if (!_dbus_object_tree_list_registered_unlocked (tree, path,
&children))
    goto out;

if (!_dbus_string_append (&xml,
DBUS_INTROSPECT_1_0_XML_DOCTYPE_DECL_NODE))
    goto out;

if (!_dbus_string_append (&xml, "<node>\n"))
    goto out;

i = 0;
while (children[i] != NULL)
{
    if (!_dbus_string_append_printf (&xml, " <node
name=\"%s\"/>\n",
                                     children[i]))
        goto out;

    ++i;
}

if (!_dbus_string_append (&xml, "</node>\n"))
    goto out;

reply = dbus_message_new_method_return (message);
if (reply == NULL)
    goto out;

```

```

    dbus_message_iter_init_append (reply, &iter);
    v_STRING = _dbus_string_get_const_data (&xml);
    if (!dbus_message_iter_append_basic (&iter, DBUS_TYPE_STRING,
&v_STRING))
        goto out;

#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        already_unlocked = TRUE;

        if (!_dbus_connection_send_and_unlock (tree->connection, reply,
NULL))
            goto out;
    }

    result = DBUS_HANDLER_RESULT_HANDLED;

out:
#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        if (!already_unlocked)
        {
            _dbus_verbose ("unlock\n");
            _dbus_connection_unlock (tree->connection);
        }
    }

    _dbus_string_free (&xml);
    dbus_free_string_array (children);
    if (reply)
        dbus_message_unref (reply);

    return result;
}

/**
 * Tries to dispatch a message by directing it to handler for the
 * object path listed in the message header, if any. Messages are
 * dispatched first to the registered handler that matches the largest
 * number of path elements; that is, message to /foo/bar/baz would go
 * to the handler for /foo/bar before the one for /foo.
 *
 *
 * @todo thread problems
 *
 * @param tree the global object tree
 * @param message the message to dispatch
 * @returns whether message was handled successfully

```

```

*/
DBusHandlerResult
_dbus_object_tree_dispatch_and_unlock (DBusObjectTree      *tree,
                                       DBusMessage
                                       *message,
                                       dbus_bool_t
                                       *found_object)
{
    char **path;
    dbus_bool_t exact_match;
    DBusList *list;
    DBusList *link;
    DBusHandlerResult result;
    DBusObjectSubtree *subtree;

#ifdef 0
    _dbus_verbose ("Dispatch of message by object path\n");
#endif

    path = NULL;
    if (!dbus_message_get_path_decomposed (message, &path))
    {
#ifdef DBUS_BUILD_TESTS
        if (tree->connection)
#endif
        {
            _dbus_verbose ("unlock\n");
            _dbus_connection_unlock (tree->connection);
        }

        _dbus_verbose ("No memory to get decomposed path\n");

        return DBUS_HANDLER_RESULT_NEED_MEMORY;
    }

    if (path == NULL)
    {
#ifdef DBUS_BUILD_TESTS
        if (tree->connection)
#endif
        {
            _dbus_verbose ("unlock\n");
            _dbus_connection_unlock (tree->connection);
        }

        _dbus_verbose ("No path field in message\n");
        return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
    }

    /* Find the deepest path that covers the path in the message */
    subtree = find_handler (tree, (const char**) path, &exact_match);

```

```

if (found_object)
    *found_object = !!subtree;

/* Build a list of all paths that cover the path in the message */

list = NULL;

while (subtree != NULL)
{
    if (subtree->message_function != NULL && (exact_match ||
subtree->invoke_as_fallback))
    {
        _dbus_object_subtree_ref (subtree);

        /* run deepest paths first */
        if (!_dbus_list_append (&list, subtree))
        {
            result = DBUS_HANDLER_RESULT_NEED_MEMORY;
            _dbus_object_subtree_unref (subtree);
            goto free_and_return;
        }
    }

    exact_match = FALSE;
    subtree = subtree->parent;
}

_dbus_verbose ("%d handlers in the path tree for this message\n",
                _dbus_list_get_length (&list));

/* Invoke each handler in the list */

result = DBUS_HANDLER_RESULT_NOT_YET_HANDLED;

link = _dbus_list_get_first_link (&list);
while (link != NULL)
{
    DBusList *next = _dbus_list_get_next_link (&list, link);
    subtree = link->data;

    /* message_function is NULL if we're unregistered
     * due to reentrancy
     */
    if (subtree->message_function)
    {
        DBusObjectPathMessageFunction message_function;
        void *user_data;

        message_function = subtree->message_function;
        user_data = subtree->user_data;
    }
}
#endif

```

```

        _dbus_verbose (" (invoking a handler)\n");
#endif

#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        _dbus_verbose ("unlock\n");
        _dbus_connection_unlock (tree->connection);
    }

    /* FIXME you could unregister the subtree in another thread
     * before we invoke the callback, and I can't figure out a
     * good way to solve this.
     */

    result = (* message_function) (tree->connection,
                                   message,
                                   user_data);

#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
        _dbus_connection_lock (tree->connection);

    if (result != DBUS_HANDLER_RESULT_NOT_YET_HANDLED)
        goto free_and_return;
}

link = next;
}

free_and_return:

if (result == DBUS_HANDLER_RESULT_NOT_YET_HANDLED)
{
    /* This hardcoded default handler does a minimal Introspect()
     */
    result = handle_default_introspect_and_unlock (tree, message,
                                                    (const char**)
path);
}
else
{
#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        _dbus_verbose ("unlock\n");
        _dbus_connection_unlock (tree->connection);
    }
}
}

```

```

while (list != NULL)
{
    link = _dbus_list_get_first_link (&list);
    _dbus_object_subtree_unref (link->data);
    _dbus_list_remove_link (&list, link);
}

dbus_free_string_array (path);

return result;
}

/**
 * Looks up the data passed to _dbus_object_tree_register() for a
 * handler at the given path.
 *
 * @param tree the global object tree
 * @param path NULL-terminated array of path elements giving path to
 subtree
 * @returns the object's user_data or #NULL if none found
 */
void*
_dbus_object_tree_get_user_data_unlocked (DBusObjectTree *tree,
                                         const char **path)
{
    dbus_bool_t exact_match;
    DBusObjectSubtree *subtree;

    _dbus_assert (tree != NULL);
    _dbus_assert (path != NULL);

    /* Find the deepest path that covers the path in the message */
    subtree = find_handler (tree, (const char**) path, &exact_match);

    if ((subtree == NULL) || !exact_match)
    {
        _dbus_verbose ("No object at specified path found\n");
        return NULL;
    }

    return subtree->user_data;
}

/**
 * Allocates a subtree object.
 *
 * @param name name to duplicate.
 * @returns newly-allocated subtree
 */
static DBusObjectSubtree*
allocate_subtree_object (const char *name)

```

```

{
    int len;
    DBusObjectSubtree *subtree;
    const size_t front_padding = _DBUS_STRUCT_OFFSET (DBusObjectSubtree,
name);

    _dbus_assert (name != NULL);

    len = strlen (name);

    subtree = dbus_malloc0 (MAX (front_padding + (len + 1), sizeof
(DBusObjectSubtree)));

    if (subtree == NULL)
        return NULL;

    memcpy (subtree->name, name, len + 1);

    return subtree;
}

static DBusObjectSubtree*
_dbus_object_subtree_new (const char *name,
                          const DBusObjectPathVTable *vtable,
                          void *user_data)
{
    DBusObjectSubtree *subtree;

    subtree = allocate_subtree_object (name);
    if (subtree == NULL)
        goto oom;

    _dbus_assert (name != NULL);

    subtree->parent = NULL;

    if (vtable)
    {
        subtree->message_function = vtable->message_function;
        subtree->unregister_function = vtable->unregister_function;
    }
    else
    {
        subtree->message_function = NULL;
        subtree->unregister_function = NULL;
    }

    subtree->user_data = user_data;
    _dbus_atomic_inc (&subtree->refcount);
    subtree->subtrees = NULL;
    subtree->n_subtrees = 0;
    subtree->max_subtrees = 0;
}

```

```

    subtree->invoke_as_fallback = FALSE;

    return subtree;

oom:
    return NULL;
}

static DBusObjectSubtree *
_dbus_object_subtree_ref (DBusObjectSubtree *subtree)
{
#ifdef DBUS_DISABLE_ASSERT
    _dbus_atomic_inc (&subtree->refcount);
#else
    dbus_int32_t old_value;

    old_value = _dbus_atomic_inc (&subtree->refcount);
    _dbus_assert (old_value > 0);
#endif

    return subtree;
}

static void
_dbus_object_subtree_unref (DBusObjectSubtree *subtree)
{
    dbus_int32_t old_value;

    old_value = _dbus_atomic_dec (&subtree->refcount);
    _dbus_assert (old_value > 0);

    if (old_value == 1)
    {
        _dbus_assert (subtree->unregister_function == NULL);
        _dbus_assert (subtree->message_function == NULL);

        dbus_free (subtree->subtrees);
        dbus_free (subtree);
    }
}

/**
 * Lists the registered fallback handlers and object path handlers at
 * the given parent_path. The returned array should be freed with
 * dbus_free_string_array().
 *
 * @param tree the object tree
 * @param parent_path the path to list the child handlers of
 * @param child_entries returns #NULL-terminated array of children
 * @returns #FALSE if no memory to allocate the child entries
 */
dbus_bool_t

```



```

_dbus_object_tree_list_registered_and_unlock (DBusObjectTree *tree,
                                              const char
**parent_path,
                                              char
***child_entries)
{
    dbus_bool_t result;

    result = _dbus_object_tree_list_registered_unlocked (tree,
                                                         parent_path,
                                                         child_entries);

#ifdef DBUS_BUILD_TESTS
    if (tree->connection)
#endif
    {
        _dbus_verbose ("unlock\n");
        _dbus_connection_unlock (tree->connection);
    }

    return result;
}

/** Set to 1 to get a bunch of spew about disassembling the path
string */
#define VERBOSE_DECOMPOSE 0

/**
 * Decompose an object path.  A path of just "/" is
 * represented as an empty vector of strings.
 * The path need not be nul terminated.
 *
 * @param data the path data
 * @param len the length of the path string
 * @param path address to store new object path
 * @param path_len length of stored path
 */
dbus_bool_t
_dbus_decompose_path (const char* data,
                    int len,
                    char ***path,
                    int *path_len)
{
    char **retval;
    int n_components;
    int i, j, comp;

    _dbus_assert (data != NULL);
    _dbus_assert (path != NULL);

#ifdef VERBOSE_DECOMPOSE

```

```

    _dbus_verbose ("Decomposing path \"%s\"\n",
                  data);
#endif

n_components = 0;
if (len > 1) /* if path is not just "/" */
{
    i = 0;
    while (i < len)
    {
        _dbus_assert (data[i] != '\0');
        if (data[i] == '/')
            n_components += 1;
        ++i;
    }
}

retval = dbus_new0 (char*, n_components + 1);

if (retval == NULL)
    return FALSE;

comp = 0;
if (n_components == 0)
    i = 1;
else
    i = 0;
while (comp < n_components)
{
    _dbus_assert (i < len);

    if (data[i] == '/')
        ++i;
    j = i;

    while (j < len && data[j] != '/')
        ++j;

    /* Now [i, j) is the path component */
    _dbus_assert (i < j);
    _dbus_assert (data[i] != '/');
    _dbus_assert (j == len || data[j] == '/');

#ifdef VERBOSE_DECOMPOSE
        _dbus_verbose (" (component in [%d,%d])\n",
                      i, j);
#endif

    retval[comp] = _dbus_memdup (&data[i], j - i + 1);
    if (retval[comp] == NULL)
    {
        dbus_free_string_array (retval);
    }
}

```

```

        return FALSE;
    }
    retval[comp][j-i] = '\0';
#ifdef VERBOSE_DECOMPOSE
    _dbus_verbose (" (component %d = \"%s\")\n",
                  comp, retval[comp]);
#endif

    ++comp;
    i = j;
}
_dbus_assert (i == len);

*path = retval;
if (path_len)
    *path_len = n_components;

return TRUE;
}

/** @} */

static char*
flatten_path (const char **path)
{
    DBusString str;
    char *s;

    if (!_dbus_string_init (&str))
        return NULL;

    if (path[0] == NULL)
    {
        if (!_dbus_string_append_byte (&str, '/'))
            goto nomem;
    }
    else
    {
        int i;

        i = 0;
        while (path[i])
        {
            if (!_dbus_string_append_byte (&str, '/'))
                goto nomem;

            if (!_dbus_string_append (&str, path[i]))
                goto nomem;

            ++i;
        }
    }
}

```

```

    if (!_dbus_string_steal_data (&str, &s))
        goto nomem;

    _dbus_string_free (&str);

    return s;

nomem:
    _dbus_string_free (&str);
    return NULL;
}

#ifdef DBUS_BUILD_TESTS

#ifndef DOXYGEN_SHOULD_SKIP_THIS

#include "dbus-test.h"
#include <stdio.h>

typedef enum
{
    STR_EQUAL,
    STR_PREFIX,
    STR_DIFFERENT
} StrComparison;

/* Returns TRUE if container is a parent of child
 */
static StrComparison
path_contains (const char **container,
               const char **child)
{
    int i;

    i = 0;
    while (child[i] != NULL)
    {
        int v;

        if (container[i] == NULL)
            return STR_PREFIX; /* container ran out, child continues;
                               * thus the container is a parent of the
                               * child.
                               */

        _dbus_assert (container[i] != NULL);
        _dbus_assert (child[i] != NULL);

        v = strcmp (container[i], child[i]);

```

```

        if (v != 0)
            return STR_DIFFERENT; /* they overlap until here and then are
different,
                                * not overlapping
                                */

        ++i;
    }

    /* Child ran out; if container also did, they are equal;
    * otherwise, the child is a parent of the container.
    */
    if (container[i] == NULL)
        return STR_EQUAL;
    else
        return STR_DIFFERENT;
}

#ifdef 0
static void
spew_subtree_recurse (DBusObjectSubtree *subtree,
                    int indent)
{
    int i;

    i = 0;
    while (i < indent)
        {
            _dbus_verbose (" ");
            ++i;
        }

    _dbus_verbose ("%s (%d children)\n",
                    subtree->name, subtree->n_subtrees);

    i = 0;
    while (i < subtree->n_subtrees)
        {
            spew_subtree_recurse (subtree->subtrees[i], indent + 2);

            ++i;
        }
}

static void
spew_tree (DBusObjectTree *tree)
{
    spew_subtree_recurse (tree->root, 0);
}
#endif

/**

```

```

* Callback data used in tests
*/
typedef struct
{
    const char **path; /**< Path */
    dbus_bool_t handler_fallback; /**< true if the handler may be called
as fallback */
    dbus_bool_t message_handled; /**< Gets set to true if message
handler called */
    dbus_bool_t handler_unregistered; /**< gets set to true if handler
is unregistered */
} TreeTestData;

static void
test_unregister_function (DBusConnection *connection,
                          void *user_data)
{
    TreeTestData *ttd = user_data;

    ttd->handler_unregistered = TRUE;
}

static DBusHandlerResult
test_message_function (DBusConnection *connection,
                      DBusMessage *message,
                      void *user_data)
{
    TreeTestData *ttd = user_data;

    ttd->message_handled = TRUE;

    return DBUS_HANDLER_RESULT_NOT_YET_HANDLED;
}

static dbus_bool_t
do_register (DBusObjectTree *tree,
            const char **path,
            dbus_bool_t fallback,
            int i,
            TreeTestData *tree_test_data)
{
    DBusObjectPathVTable vtable = { test_unregister_function,
                                    test_message_function, NULL };

    tree_test_data[i].message_handled = FALSE;
    tree_test_data[i].handler_unregistered = FALSE;
    tree_test_data[i].handler_fallback = fallback;
    tree_test_data[i].path = path;

    if (!_dbus_object_tree_register (tree, fallback, path,
                                    &vtable,

```

```

        &tree_test_data[i],
        NULL))

    return FALSE;

==
    _dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path)
                  &tree_test_data[i]);

    return TRUE;
}

static dbus_bool_t
do_test_dispatch (DBusObjectTree *tree,
                 const char **path,
                 int i,
                 TreeTestData *tree_test_data,
                 int n_test_data)
{
    DBusMessage *message;
    int j;
    DBusHandlerResult result;
    char *flat;

    message = NULL;

    flat = flatten_path (path);
    if (flat == NULL)
        goto oom;

    message = dbus_message_new_method_call (NULL,
                                           flat,

"org.freedesktop.TestInterface",

                                           "Foo");

    dbus_free (flat);
    if (message == NULL)
        goto oom;

    j = 0;
    while (j < n_test_data)
    {
        tree_test_data[j].message_handled = FALSE;
        ++j;
    }

    result = _dbus_object_tree_dispatch_and_unlock (tree, message,
    NULL);
    if (result == DBUS_HANDLER_RESULT_NEED_MEMORY)
        goto oom;

    _dbus_assert (tree_test_data[i].message_handled);

```

```

j = 0;
while (j < n_test_data)
{
    if (tree_test_data[j].message_handled)
    {
        if (tree_test_data[j].handler_fallback)
            _dbus_assert (path_contains (tree_test_data[j].path,
                path) != STR_DIFFERENT);
        else
            _dbus_assert (path_contains (tree_test_data[j].path, path) ==
STR_EQUAL);
    }
    else
    {
        if (tree_test_data[j].handler_fallback)
            _dbus_assert (path_contains (tree_test_data[j].path,
                path) == STR_DIFFERENT);
        else
            _dbus_assert (path_contains (tree_test_data[j].path, path) !=
STR_EQUAL);
    }

    ++j;
}

dbus_message_unref (message);

return TRUE;

oom:
if (message)
    dbus_message_unref (message);
return FALSE;
}

static size_t
string_array_length (const char **array)
{
    size_t i;
    for (i = 0; array[i]; i++) ;
    return i;
}

typedef struct
{
    const char *path;
    const char *result[20];
} DecomposePathTest;

static DecomposePathTest decompose_tests[] = {
    { "/foo", { "foo", NULL } },
    { "/foo/bar", { "foo", "bar", NULL } },

```



```

    { "/", { NULL } },
    { "/a/b", { "a", "b", NULL } },
    { "/a/b/c", { "a", "b", "c", NULL } },
    { "/a/b/c/d", { "a", "b", "c", "d", NULL } },
    { "/foo/bar/q", { "foo", "bar", "q", NULL } },
    { "/foo/bar/this/is/longer", { "foo", "bar", "this", "is", "longer",
NULL } }
};

```

```

static dbus_bool_t
run_decompose_tests (void)
{
    int i;

    i = 0;
    while (i < _DBUS_N_ELEMENTS (decompose_tests))
    {
        char **result;
        int result_len;
        int expected_len;

        if (!_dbus_decompose_path (decompose_tests[i].path,
                                strlen (decompose_tests[i].path),
                                &result, &result_len))

            return FALSE;

        expected_len = string_array_length (decompose_tests[i].result);

        if (result_len != (int) string_array_length ((const
char**)result) ||
            expected_len != result_len ||
            path_contains (decompose_tests[i].result,
                          (const char**) result) != STR_EQUAL)
        {
            int real_len = string_array_length ((const char**)result);
            _dbus_warn ("Expected decompose of %s to have len %d,
returned %d, appears to have %d\n",
                      decompose_tests[i].path, expected_len,
result_len,
                      real_len);
            _dbus_warn ("Decompose resulted in elements: { ");
            i = 0;
            while (i < real_len)
            {
                _dbus_warn ("\\"%s\\"%s", result[i],
                          (i + 1) == real_len ? "" : ", ");
                ++i;
            }
            _dbus_warn ("}\n");
            _dbus_assert_not_reached ("path decompose failed\n");
        }
    }
}

```

```

        dbus_free_string_array (result);

        ++i;
    }

    return TRUE;
}

static dbus_bool_t
object_tree_test_iteration (void *data)
{
    const char *path0[] = { NULL };
    const char *path1[] = { "foo", NULL };
    const char *path2[] = { "foo", "bar", NULL };
    const char *path3[] = { "foo", "bar", "baz", NULL };
    const char *path4[] = { "foo", "bar", "boo", NULL };
    const char *path5[] = { "blah", NULL };
    const char *path6[] = { "blah", "boof", NULL };
    const char *path7[] = { "blah", "boof", "this", "is", "really",
"long", NULL };
    const char *path8[] = { "childless", NULL };
    DBusObjectTree *tree;
    TreeTestData tree_test_data[9];
    int i;
    dbus_bool_t exact_match;

    if (!run_decompose_tests ())
        return FALSE;

    tree = NULL;

    tree = _dbus_object_tree_new (NULL);
    if (tree == NULL)
        goto out;

    if (!do_register (tree, path0, TRUE, 0, tree_test_data))
        goto out;

    _dbus_assert (find_subtree (tree, path0, NULL));
    _dbus_assert (!find_subtree (tree, path1, NULL));
    _dbus_assert (!find_subtree (tree, path2, NULL));
    _dbus_assert (!find_subtree (tree, path3, NULL));
    _dbus_assert (!find_subtree (tree, path4, NULL));
    _dbus_assert (!find_subtree (tree, path5, NULL));
    _dbus_assert (!find_subtree (tree, path6, NULL));
    _dbus_assert (!find_subtree (tree, path7, NULL));
    _dbus_assert (!find_subtree (tree, path8, NULL));

    _dbus_assert (find_handler (tree, path0, &exact_match) &&
exact_match);
    _dbus_assert (find_handler (tree, path1, &exact_match) == tree->root
&& !exact_match);

```

```

    _dbus_assert (find_handler (tree, path2, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path3, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path4, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path5, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path6, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path7, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path8, &exact_match) == tree->root
&& !exact_match);

    if (!do_register (tree, path1, TRUE, 1, tree_test_data))
        goto out;

    _dbus_assert (find_subtree (tree, path0, NULL));
    _dbus_assert (find_subtree (tree, path1, NULL));
    _dbus_assert (!find_subtree (tree, path2, NULL));
    _dbus_assert (!find_subtree (tree, path3, NULL));
    _dbus_assert (!find_subtree (tree, path4, NULL));
    _dbus_assert (!find_subtree (tree, path5, NULL));
    _dbus_assert (!find_subtree (tree, path6, NULL));
    _dbus_assert (!find_subtree (tree, path7, NULL));
    _dbus_assert (!find_subtree (tree, path8, NULL));

    _dbus_assert (find_handler (tree, path0, &exact_match) &&
exact_match);
    _dbus_assert (find_handler (tree, path1, &exact_match) &&
exact_match);
    _dbus_assert (find_handler (tree, path2, &exact_match) &&
!exact_match);
    _dbus_assert (find_handler (tree, path3, &exact_match) &&
!exact_match);
    _dbus_assert (find_handler (tree, path4, &exact_match) &&
!exact_match);
    _dbus_assert (find_handler (tree, path5, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path6, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path7, &exact_match) == tree->root
&& !exact_match);
    _dbus_assert (find_handler (tree, path8, &exact_match) == tree->root
&& !exact_match);

    if (!do_register (tree, path2, TRUE, 2, tree_test_data))
        goto out;

    _dbus_assert (find_subtree (tree, path1, NULL));
    _dbus_assert (find_subtree (tree, path2, NULL));

```

```

_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));

if (!do_register (tree, path3, TRUE, 3, tree_test_data))
    goto out;

_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));

if (!do_register (tree, path4, TRUE, 4, tree_test_data))
    goto out;

_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));

if (!do_register (tree, path5, TRUE, 5, tree_test_data))
    goto out;

_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));

_dbus_assert (find_handler (tree, path0, &exact_match) == tree->root
&& exact_match);
_dbus_assert (find_handler (tree, path1, &exact_match) != tree->root
&& exact_match);
_dbus_assert (find_handler (tree, path2, &exact_match) != tree->root
&& exact_match);

```

```
_dbus_assert (find_handler (tree, path3, &exact_match) != tree->root
&& exact_match);
_dbus_assert (find_handler (tree, path4, &exact_match) != tree->root
&& exact_match);
_dbus_assert (find_handler (tree, path5, &exact_match) != tree->root
&& exact_match);
_dbus_assert (find_handler (tree, path6, &exact_match) != tree->root
&& !exact_match);
_dbus_assert (find_handler (tree, path7, &exact_match) != tree->root
&& !exact_match);
_dbus_assert (find_handler (tree, path8, &exact_match) == tree->root
&& !exact_match);
```

```
if (!do_register (tree, path6, TRUE, 6, tree_test_data))
    goto out;
```

```
_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));
```

```
if (!do_register (tree, path7, TRUE, 7, tree_test_data))
    goto out;
```

```
_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));
```

```
if (!do_register (tree, path8, TRUE, 8, tree_test_data))
    goto out;
```

```
_dbus_assert (find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));
```

```

    _dbus_assert (find_handler (tree, path0, &exact_match) == tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path1, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path2, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path3, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path4, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path5, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path6, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path7, &exact_match) != tree->root
&& exact_match);
    _dbus_assert (find_handler (tree, path8, &exact_match) != tree->root
&& exact_match);

    /* test the list_registered function */

    {
        const char *root[] = { NULL };
        char **child_entries;
        int nb;

        _dbus_object_tree_list_registered_unlocked (tree, path1,
&child_entries);
        if (child_entries != NULL)
            {
                nb = string_array_length ((const char**)child_entries);
                _dbus_assert (nb == 1);
                dbus_free_string_array (child_entries);
            }

        _dbus_object_tree_list_registered_unlocked (tree, path2,
&child_entries);
        if (child_entries != NULL)
            {
                nb = string_array_length ((const char**)child_entries);
                _dbus_assert (nb == 2);
                dbus_free_string_array (child_entries);
            }

        _dbus_object_tree_list_registered_unlocked (tree, path8,
&child_entries);
        if (child_entries != NULL)
            {
                nb = string_array_length ((const char**)child_entries);
                _dbus_assert (nb == 0);
                dbus_free_string_array (child_entries);
            }
    }

```

```

    _dbus_object_tree_list_registered_unlocked (tree, root,
&child_entries);
    if (child_entries != NULL)
        {
            nb = string_array_length ((const char**)child_entries);
            _dbus_assert (nb == 3);
            dbus_free_string_array (child_entries);
        }
}

/* Check that destroying tree calls unregister funcs */
_dbus_object_tree_unref (tree);

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (tree_test_data))
    {
        _dbus_assert (tree_test_data[i].handler_unregister);
        _dbus_assert (!tree_test_data[i].message_handled);
        ++i;
    }

/* Now start again and try the individual unregister function */
tree = _dbus_object_tree_new (NULL);
if (tree == NULL)
    goto out;

if (!do_register (tree, path0, TRUE, 0, tree_test_data))
    goto out;
if (!do_register (tree, path1, TRUE, 1, tree_test_data))
    goto out;
if (!do_register (tree, path2, TRUE, 2, tree_test_data))
    goto out;
if (!do_register (tree, path3, TRUE, 3, tree_test_data))
    goto out;
if (!do_register (tree, path4, TRUE, 4, tree_test_data))
    goto out;
if (!do_register (tree, path5, TRUE, 5, tree_test_data))
    goto out;
if (!do_register (tree, path6, TRUE, 6, tree_test_data))
    goto out;
if (!do_register (tree, path7, TRUE, 7, tree_test_data))
    goto out;
if (!do_register (tree, path8, TRUE, 8, tree_test_data))
    goto out;

_dbus_object_tree_unregister_and_unlock (tree, path0);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path0)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (find_subtree (tree, path1, NULL));

```

```

_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path1);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path1)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path2);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path2)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path3);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path3)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path4);

```



```

_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path4)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path5);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path5)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path6);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path6)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path7);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path7)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));

```

```

_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (find_subtree (tree, path8, NULL));

_dbus_object_tree_unregister_and_unlock (tree, path8);
_dbus_assert (_dbus_object_tree_get_user_data_unlocked (tree, path8)
== NULL);

_dbus_assert (!find_subtree (tree, path0, NULL));
_dbus_assert (!find_subtree (tree, path1, NULL));
_dbus_assert (!find_subtree (tree, path2, NULL));
_dbus_assert (!find_subtree (tree, path3, NULL));
_dbus_assert (!find_subtree (tree, path4, NULL));
_dbus_assert (!find_subtree (tree, path5, NULL));
_dbus_assert (!find_subtree (tree, path6, NULL));
_dbus_assert (!find_subtree (tree, path7, NULL));
_dbus_assert (!find_subtree (tree, path8, NULL));

i = 0;
while (i < (int) _DBUS_N_ELEMENTS (tree_test_data))
{
    _dbus_assert (tree_test_data[i].handler_unregister);
    _dbus_assert (!tree_test_data[i].message_handled);
    ++i;
}

/* Register it all again, and test dispatch */

if (!do_register (tree, path0, TRUE, 0, tree_test_data))
    goto out;
if (!do_register (tree, path1, FALSE, 1, tree_test_data))
    goto out;
if (!do_register (tree, path2, TRUE, 2, tree_test_data))
    goto out;
if (!do_register (tree, path3, TRUE, 3, tree_test_data))
    goto out;
if (!do_register (tree, path4, TRUE, 4, tree_test_data))
    goto out;
if (!do_register (tree, path5, TRUE, 5, tree_test_data))
    goto out;
if (!do_register (tree, path6, FALSE, 6, tree_test_data))
    goto out;
if (!do_register (tree, path7, TRUE, 7, tree_test_data))
    goto out;
if (!do_register (tree, path8, TRUE, 8, tree_test_data))
    goto out;

#if 0
    spew_tree (tree);
#endif

    if (!do_test_dispatch (tree, path0, 0, tree_test_data,
_DBUS_N_ELEMENTS (tree_test_data)))

```

```

        goto out;
    if (!do_test_dispatch (tree, path1, 1, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path2, 2, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path3, 3, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path4, 4, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path5, 5, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path6, 6, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path7, 7, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;
    if (!do_test_dispatch (tree, path8, 8, tree_test_data,
        _DBUS_N_ELEMENTS (tree_test_data)))
        goto out;

out:
    if (tree)
    {
        /* test ref */
        _dbus_object_tree_ref (tree);
        _dbus_object_tree_unref (tree);
        _dbus_object_tree_unref (tree);
    }

    return TRUE;
}

/**
 * @ingroup DBusObjectTree
 * Unit test for DBusObjectTree
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_object_tree_test (void)
{
    _dbus_test_oom_handling ("object tree",
        object_tree_test_iteration,
        NULL);

    return TRUE;
}

```

```

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

#endif /* DBUS_BUILD_TESTS */

File = dbus-object-tree.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-object-tree.h DBusObjectTree (internals of DBusConnection)
 *
 * Copyright (C) 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifdef DBUS_OBJECT_TREE_H
#define DBUS_OBJECT_TREE_H

#include <dbus/dbus-connection.h>

DBUS_BEGIN_DECLS

typedef struct DBusObjectTree DBusObjectTree;

DBusObjectTree* _dbus_object_tree_new (DBusConnection *connection);
DBusObjectTree* _dbus_object_tree_ref (DBusObjectTree *tree);
void _dbus_object_tree_unref (DBusObjectTree *tree);

dbus_bool_t _dbus_object_tree_register
(DBusObjectTree *tree,

dbus_bool_t fallback,

```

```

**path,                                     const char
DBusObjectPathVTable *vtable,               const
*user_data,                                 void
*error);                                     DBusError
void _dbus_object_tree_unregister_and_unlock
(DBusObjectTree *tree,                       const char
**path);
DBusHandlerResult _dbus_object_tree_dispatch_and_unlock
(DBusObjectTree *tree,
DBusMessage *message,
dbus_bool_t *found_object);
void* _dbus_object_tree_get_user_data_unlocked
(DBusObjectTree *tree,                       const char
**path);
void _dbus_object_tree_free_all_unlocked
(DBusObjectTree *tree);

dbus_bool_t _dbus_object_tree_list_registered_and_unlock
(DBusObjectTree *tree,                       const char
**parent_path,                               char
***child_entries);

dbus_bool_t _dbus_decompose_path (const char *data,
int len,
char ***path,
int *path_len);

DBUS_END_DECLS

#endif /* DBUS_OBJECT_TREE_H */

```

File = dbus-pending-call-internal.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pending-call-internal.h DBusPendingCall internal interfaces
 *
 * Copyright (C) 2002 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1

```

```

*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#endif DBUS_PENDING_CALL_INTERNAL_H
#define DBUS_PENDING_CALL_INTERNAL_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-message.h>
#include <dbus/dbus-connection.h>
#include <dbus/dbus-list.h>

DBUS_BEGIN_DECLS

dbus_bool_t      _dbus_pending_call_is_timeout_added_unlocked
(DBusPendingCall *pending);
void             _dbus_pending_call_set_timeout_added_unlocked
(DBusPendingCall *pending,

dbus_bool_t      is_added);
DBusTimeout     * _dbus_pending_call_get_timeout_unlocked
(DBusPendingCall *pending);
dbus_uint32_t    _dbus_pending_call_get_reply_serial_unlocked
(DBusPendingCall *pending);
void             _dbus_pending_call_set_reply_serial_unlocked
(DBusPendingCall *pending,

dbus_uint32_t    serial);
DBusConnection * _dbus_pending_call_get_connection_and_lock
(DBusPendingCall *pending);
DBusConnection * _dbus_pending_call_get_connection_unlocked
(DBusPendingCall *pending);
dbus_bool_t      _dbus_pending_call_get_completed_unlocked
(DBusPendingCall *pending);
void             _dbus_pending_call_complete
(DBusPendingCall *pending);

```

```

void          _dbus_pending_call_set_reply_unlocked
(DBusPendingCall  *pending,

DBusMessage      *message);
void          _dbus_pending_call_queue_timeout_error_unlocked
(DBusPendingCall  *pending,

DBusConnection   *connection);
void          _dbus_pending_call_set_reply_serial_unlocked
(DBusPendingCall  *pending,

dbus_uint32_t     serial);
dbus_bool_t     _dbus_pending_call_set_timeout_error_unlocked
(DBusPendingCall  *pending,

DBusMessage      *message,

dbus_uint32_t     serial);
DBusPendingCall* _dbus_pending_call_new_unlocked
(DBusConnection   *connection,

                                int
timeout_milliseconds,

DBusTimeoutHandler timeout_handler);
DBusPendingCall* _dbus_pending_call_ref_unlocked
(DBusPendingCall  *pending);
void          _dbus_pending_call_unref_and_unlock
(DBusPendingCall  *pending);
dbus_bool_t     _dbus_pending_call_set_data_unlocked
(DBusPendingCall  *pending,

dbus_int32_t      slot,

                                void
*data,

DBusFreeFunction  free_data_func);

DBUS_END_DECLS

#endif /* DBUS_PENDING_CALL_INTERNAL_H */

File = dbus-pending-call.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pending-call.c Object representing a call in progress.
 *
 * Copyright (C) 2002, 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1

```

```
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-internals.h"
#include "dbus-connection-internal.h"
#include "dbus-message-internal.h"
#include "dbus-pending-call-internal.h"
#include "dbus-pending-call.h"
#include "dbus-list.h"
#include "dbus-threads.h"
#include "dbus-test.h"
```

```
/**
 * @defgroup DBusPendingCallInternals DBusPendingCall implementation
details
 * @ingroup DBusInternals
 * @brief DBusPendingCall private implementation details.
 *
 * The guts of DBusPendingCall and its methods.
 *
 * @{
 */
```

```
/**
 * @brief Internals of DBusPendingCall
 *
 * Opaque object representing a reply message that we're waiting for.
 */
```

```
/**
 * shorter and more visible way to write _dbus_connection_lock()
 */
```

```
#define CONNECTION_LOCK(connection)
_dbus_connection_lock(connection)
```



```

/**
 * shorter and more visible way to write _dbus_connection_unlock()
 */
#define CONNECTION_UNLOCK(connection)
_dbus_connection_unlock(connection)

/**
 * Implementation details of #DBusPendingCall - all fields are
 private.
 */
struct DBusPendingCall
{
    DBusAtomic refcount;           /**< reference count
 */

    DBusDataSlotList slot_list;   /**< Data stored by
 allocated integer ID */

    DBusPendingCallNotifyFunction function; /**< Notifier when
 reply arrives. */

    DBusConnection *connection;   /**< Connections
 we're associated with */
    DBusMessage *reply;           /**< Reply (after
 we've received it) */
    DBusTimeout *timeout;         /**< Timeout */

    DBusList *timeout_link;       /**< Preallocated
 timeout response */

    dbus_uint32_t reply_serial;    /**< Expected serial
 of reply */

    unsigned int completed : 1;    /**< TRUE if
 completed */
    unsigned int timeout_added : 1; /**< Have added the
 timeout */
};

#ifdef DBUS_ENABLE_VERBOSE_MODE
static void
_dbus_pending_call_trace_ref (DBusPendingCall *pending_call,
    int old_refcount,
    int new_refcount,
    const char *why)
{
    static int enabled = -1;

    _dbus_trace_ref ("DBusPendingCall", pending_call, old_refcount,
        new_refcount, why, "DBUS_PENDING_CALL_TRACE", &enabled);
}
#else

```



```

    if (timeout == NULL)
    {
        dbus_pending_call_free_data_slot (&notify_user_data_slot);
        dbus_free (pending);
        return NULL;
    }

    pending->timeout = timeout;
}
else
{
    pending->timeout = NULL;
}

_dbus_atomic_inc (&pending->refcount);
pending->connection = connection;
_dbus_connection_ref_unlocked (pending->connection);

_dbus_data_slot_list_init (&pending->slot_list);

_dbus_pending_call_trace_ref (pending, 0, 1, "new_unlocked");

return pending;
}

/**
 * Sets the reply of a pending call with the given message,
 * or if the message is #NULL, by timing out the pending call.
 *
 * @param pending the pending call
 * @param message the message to complete the call with, or #NULL
 * to time out the call
 */
void
_dbus_pending_call_set_reply_unlocked (DBusPendingCall *pending,
                                     DBusMessage      *message)
{
    if (message == NULL)
    {
        message = pending->timeout_link->data;
        _dbus_list_clear (&pending->timeout_link);
    }
    else
        dbus_message_ref (message);

    _dbus_verbose (" handling message %p (%s) to pending call serial
%u\n",
                 message,
                 dbus_message_get_type (message) ==
DBUS_MESSAGE_TYPE_METHOD_RETURN ?
                 "method return" :

```

```

        dbus_message_get_type (message) ==
DBUS_MESSAGE_TYPE_ERROR ?
        "error" : "other type",
        pending->reply_serial);

    _dbus_assert (pending->reply == NULL);
    _dbus_assert (pending->reply_serial == dbus_message_get_reply_serial
(message));
    pending->reply = message;
}

/**
 * Calls notifier function for the pending call
 * and sets the call to completed.
 *
 * @param pending the pending call
 */
void
_dbus_pending_call_complete (DBusPendingCall *pending)
{
    _dbus_assert (!pending->completed);

    pending->completed = TRUE;

    if (pending->function)
    {
        void *user_data;
        user_data = dbus_pending_call_get_data (pending,
                                                notify_user_data_slot);

        (* pending->function) (pending, user_data);
    }
}

/**
 * If the pending call hasn't been timed out, add its timeout
 * error reply to the connection's incoming message queue.
 *
 * @param pending the pending call
 * @param connection the connection the call was sent to
 */
void
_dbus_pending_call_queue_timeout_error_unlocked (DBusPendingCall
*pending,
                                                DBusConnection
*connection)
{
    _dbus_assert (connection == pending->connection);

    if (pending->timeout_link)
    {

```

```

        _dbus_connection_queue_synthesized_message_link (connection,
                                                         pending->timeout_link);
    pending->timeout_link = NULL;
}
}

/**
 * Checks to see if a timeout has been added
 *
 * @param pending the pending_call
 * @returns #TRUE if there is a timeout or #FALSE if not
 */
dbus_bool_t
_dbus_pending_call_is_timeout_added_unlocked (DBusPendingCall
*pending)
{
    _dbus_assert (pending != NULL);

    return pending->timeout_added;
}

/**
 * Sets whether the timeout has been added
 *
 * @param pending the pending_call
 * @param is_added whether or not a timeout is added
 */
void
_dbus_pending_call_set_timeout_added_unlocked (DBusPendingCall
*pending,
                                               dbus_bool_t
is_added)
{
    _dbus_assert (pending != NULL);

    pending->timeout_added = is_added;
}

/**
 * Retrieves the timeout
 *
 * @param pending the pending_call
 * @returns a timeout object or NULL if call has no timeout
 */
DBusTimeout *
_dbus_pending_call_get_timeout_unlocked (DBusPendingCall *pending)
{
    _dbus_assert (pending != NULL);

    return pending->timeout;
}

```

```

}

/**
 * Gets the reply's serial number
 *
 * @param pending the pending_call
 * @returns a serial number for the reply or 0
 */
dbus_uint32_t
_dbus_pending_call_get_reply_serial_unlocked (DBusPendingCall
*pending)
{
    _dbus_assert (pending != NULL);

    return pending->reply_serial;
}

/**
 * Sets the reply's serial number
 *
 * @param pending the pending_call
 * @param serial the serial number
 */
void
_dbus_pending_call_set_reply_serial_unlocked (DBusPendingCall
*pending,
                                             dbus_uint32_t serial)
{
    _dbus_assert (pending != NULL);
    _dbus_assert (pending->reply_serial == 0);

    pending->reply_serial = serial;
}

/**
 * Gets the connection associated with this pending call.
 *
 * @param pending the pending_call
 * @returns the connection associated with the pending call
 */
DBusConnection *
_dbus_pending_call_get_connection_and_lock (DBusPendingCall *pending)
{
    _dbus_assert (pending != NULL);

    CONNECTION_LOCK (pending->connection);
    return pending->connection;
}

/**
 * Gets the connection associated with this pending call.
 *

```

```

* @param pending the pending_call
* @returns the connection associated with the pending call
*/
DBusConnection *
_dbus_pending_call_get_connection_unlocked (DBusPendingCall *pending)
{
    _dbus_assert (pending != NULL);

    return pending->connection;
}

/**
* Sets the reply message associated with the pending call to a
timeout error
*
* @param pending the pending_call
* @param message the message we are sending the error reply to
* @param serial serial number for the reply
* @return #FALSE on OOM
*/
dbus_bool_t
_dbus_pending_call_set_timeout_error_unlocked (DBusPendingCall
*pending,
                                             DBusMessage
*message,
                                             dbus_uint32_t
serial)
{
    DBusList *reply_link;
    DBusMessage *reply;

    reply = dbus_message_new_error (message, DBUS_ERROR_NO_REPLY,
    "Did not receive a reply. Possible
causes include: "
    "the remote application did not send
a reply, "
    "the message bus security policy
blocked the reply, "
    "the reply timeout expired, or "
    "the network connection was
broken.");
    if (reply == NULL)
        return FALSE;

    reply_link = _dbus_list_alloc_link (reply);
    if (reply_link == NULL)
    {
        /* it's OK to unref this, nothing that could have attached a
callback
* has ever seen it */
        dbus_message_unref (reply);
        return FALSE;
    }
}

```

```

    }

    pending->timeout_link = reply_link;

    _dbus_pending_call_set_reply_serial_unlocked (pending, serial);

    return TRUE;
}

/**
 * Increments the reference count on a pending call,
 * while the lock on its connection is already held.
 *
 * @param pending the pending call object
 * @returns the pending call object
 */
DBusPendingCall *
_dbus_pending_call_ref_unlocked (DBusPendingCall *pending)
{
    dbus_int32_t old_refcount;

    old_refcount = _dbus_atomic_inc (&pending->refcount);
    _dbus_pending_call_trace_ref (pending, old_refcount, old_refcount +
1,
    "ref_unlocked");

    return pending;
}

static void
_dbus_pending_call_last_unref (DBusPendingCall *pending)
{
    DBusConnection *connection;

    /* If we get here, we should be already detached
     * from the connection, or never attached.
     */
    _dbus_assert (!pending->timeout_added);

    connection = pending->connection;

    /* this assumes we aren't holding connection lock... */
    _dbus_data_slot_list_free (&pending->slot_list);

    if (pending->timeout != NULL)
        _dbus_timeout_unref (pending->timeout);

    if (pending->timeout_link)
    {
        dbus_message_unref ((DBusMessage *)pending->timeout_link->data);
        _dbus_list_free_link (pending->timeout_link);
    }
}

```



```

    pending->timeout_link = NULL;
}

if (pending->reply)
{
    dbus_message_unref (pending->reply);
    pending->reply = NULL;
}

dbus_free (pending);

dbus_pending_call_free_data_slot (&notify_user_data_slot);

/* connection lock should not be held. */
/* Free the connection last to avoid a weird state while
 * calling out to application code where the pending exists
 * but not the connection.
 */
dbus_connection_unref (connection);
}

/**
 * Decrements the reference count on a pending call,
 * freeing it if the count reaches 0. Assumes
 * connection lock is already held.
 *
 * @param pending the pending call object
 */
void
_dbus_pending_call_unref_and_unlock (DBusPendingCall *pending)
{
    dbus_int32_t old_refcount;

    old_refcount = _dbus_atomic_dec (&pending->refcount);
    _dbus_assert (old_refcount > 0);
    _dbus_pending_call_trace_ref (pending, old_refcount,
        old_refcount - 1, "unref_and_unlock");

    CONNECTION_UNLOCK (pending->connection);

    if (old_refcount == 1)
        _dbus_pending_call_last_unref (pending);
}

/**
 * Checks whether the pending call has received a reply
 * yet, or not. Assumes connection lock is held.
 *
 * @param pending the pending call
 * @returns #TRUE if a reply has been received
 */
dbus_bool_t

```

```

_dbus_pending_call_get_completed_unlocked (DBusPendingCall
*pending)
{
    return pending->completed;
}

static DBusDataSlotAllocator slot_allocator;
_DBUS_DEFINE_GLOBAL_LOCK (pending_call_slots);

/**
 * Stores a pointer on a #DBusPendingCall, along
 * with an optional function to be used for freeing
 * the data when the data is set again, or when
 * the pending call is finalized. The slot number
 * must have been allocated with
dbus_pending_call_allocate_data_slot().
 *
 * @param pending the pending_call
 * @param slot the slot number
 * @param data the data to store
 * @param free_data_func finalizer function for the data
 * @returns #TRUE if there was enough memory to store the data
 */
dbus_bool_t
_dbus_pending_call_set_data_unlocked (DBusPendingCall *pending,
                                     dbus_int32_t      slot,
                                     void                *data,
                                     DBusFreeFunction    free_data_func)
{
    DBusFreeFunction old_free_func;
    void *old_data;
    dbus_bool_t retval;

    retval = _dbus_data_slot_list_set (&slot_allocator,
                                       &pending->slot_list,
                                       slot, data, free_data_func,
                                       &old_free_func, &old_data);

    /* Drop locks to call out to app code */
    CONNECTION_UNLOCK (pending->connection);

    if (retval)
    {
        if (old_free_func)
            (* old_free_func) (old_data);
    }

    CONNECTION_LOCK (pending->connection);

    return retval;
}

```

```

/** @} */

/**
 * @defgroup DBusPendingCall DBusPendingCall
 * @ingroup DBus
 * @brief Pending reply to a method call message
 *
 * A DBusPendingCall is an object representing an
 * expected reply. A #DBusPendingCall can be created
 * when you send a message that should have a reply.
 *
 * @{
 */

/**
 * @def DBUS_TIMEOUT_INFINITE
 *
 * An integer constant representing an infinite timeout. This has the
 * numeric value 0x7fffffff (the largest 32-bit signed integer).
 *
 * For source compatibility with D-Bus versions earlier than 1.4.12,
use
 * 0x7fffffff, or INT32_MAX (assuming your platform has it).
 */

/**
 * @def DBUS_TIMEOUT_USE_DEFAULT
 *
 * An integer constant representing a request to use the default
timeout.
 * This has numeric value -1.
 *
 * For source compatibility with D-Bus versions earlier than 1.4.12,
use a
 * literal -1.
 */

/**
 * @typedef DBusPendingCall
 *
 * Opaque data type representing a message pending.
 */

/**
 * Increments the reference count on a pending call.
 *
 * @param pending the pending call object
 * @returns the pending call object
 */
DBusPendingCall *
dbus_pending_call_ref (DBusPendingCall *pending)
{

```

```

    dbus_int32_t old_refcount;

    _dbus_return_val_if_fail (pending != NULL, NULL);

    old_refcount = _dbus_atomic_inc (&pending->refcount);
    _dbus_pending_call_trace_ref (pending, old_refcount, old_refcount +
1,
        "ref");

    return pending;
}

/**
 * Decrements the reference count on a pending call,
 * freeing it if the count reaches 0.
 *
 * @param pending the pending call object
 */
void
dbus_pending_call_unref (DBusPendingCall *pending)
{
    dbus_int32_t old_refcount;

    _dbus_return_if_fail (pending != NULL);

    old_refcount = _dbus_atomic_dec (&pending->refcount);
    _dbus_pending_call_trace_ref (pending, old_refcount, old_refcount -
1,
        "unref");

    if (old_refcount == 1)
        _dbus_pending_call_last_unref (pending);
}

/**
 * Sets a notification function to be called when the reply is
 * received or the pending call times out.
 *
 * @param pending the pending call
 * @param function notifier function
 * @param user_data data to pass to notifier function
 * @param free_user_data function to free the user data
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
dbus_pending_call_set_notify (DBusPendingCall *pending,
                             DBusPendingCallNotifyFunction function,
                             void *user_data,
                             DBusFreeFunction
free_user_data)
{
    dbus_bool_t ret = FALSE;

```

```

    _dbus_return_val_if_fail (pending != NULL, FALSE);

    CONNECTION_LOCK (pending->connection);

    /* could invoke application code! */
    if (!_dbus_pending_call_set_data_unlocked (pending,
notify_user_data_slot,
                                                user_data,
free_user_data))
        goto out;

    pending->function = function;
    ret = TRUE;

out:
    CONNECTION_UNLOCK (pending->connection);

    return ret;
}

/**
 * Cancels the pending call, such that any reply or error received
 * will just be ignored. Drops the dbus library's internal reference
 * to the #DBusPendingCall so will free the call if nobody else is
 * holding a reference. However you usually get a reference from
 * dbus_connection_send_with_reply() so probably your app owns a ref
 * also.
 *
 * Note that canceling a pending call will not simulate a
 * timed-out call; if a call times out, then a timeout error reply is
 * received. If you cancel the call, no reply is received unless the
 * the reply was already received before you canceled.
 *
 * @param pending the pending call
 */
void
dbus_pending_call_cancel (DBusPendingCall *pending)
{
    _dbus_return_if_fail (pending != NULL);

    _dbus_connection_remove_pending_call (pending->connection,
                                                pending);
}

/**
 * Checks whether the pending call has received a reply
 * yet, or not.
 *
 * @param pending the pending call
 * @returns #TRUE if a reply has been received
 */

```

```

dbus_bool_t
dbus_pending_call_get_completed (DBusPendingCall *pending)
{
    dbus_bool_t completed;

    _dbus_return_val_if_fail (pending != NULL, FALSE);

    CONNECTION_LOCK (pending->connection);
    completed = pending->completed;
    CONNECTION_UNLOCK (pending->connection);

    return completed;
}

/**
 * Gets the reply, or returns #NULL if none has been received
 * yet. Ownership of the reply message passes to the caller. This
 * function can only be called once per pending call, since the reply
 * message is transferred to the caller.
 *
 * @param pending the pending call
 * @returns the reply message or #NULL.
 */
DBusMessage*
dbus_pending_call_steal_reply (DBusPendingCall *pending)
{
    DBusMessage *message;

    _dbus_return_val_if_fail (pending != NULL, NULL);
    _dbus_return_val_if_fail (pending->completed, NULL);
    _dbus_return_val_if_fail (pending->reply != NULL, NULL);

    CONNECTION_LOCK (pending->connection);

    message = pending->reply;
    pending->reply = NULL;

    CONNECTION_UNLOCK (pending->connection);

    _dbus_message_trace_ref (message, -1, -1,
"dbus_pending_call_steal_reply");
    return message;
}

/**
 * Block until the pending call is completed. The blocking is as with
 * dbus_connection_send_with_reply_and_block(); it does not enter the
 * main loop or process other messages, it simply waits for the reply
 * in question.
 *
 * If the pending call is already completed, this function returns
 * immediately.

```

```

*
* @todo when you start blocking, the timeout is reset, but it should
* really only use time remaining since the pending call was created.
* This requires storing timestamps instead of intervals in the
timeout
*
* @param pending the pending call
*/
void
dbus_pending_call_block (DBusPendingCall *pending)
{
    _dbus_return_if_fail (pending != NULL);

    _dbus_connection_block_pending_call (pending);
}

/**
 * Allocates an integer ID to be used for storing application-specific
 * data on any DBusPendingCall. The allocated ID may then be used
 * with dbus_pending_call_set_data() and dbus_pending_call_get_data().
 * The passed-in slot must be initialized to -1, and is filled in
 * with the slot ID. If the passed-in slot is not -1, it's assumed
 * to be already allocated, and its refcount is incremented.
 *
 * The allocated slot is global, i.e. all DBusPendingCall objects will
 * have a slot with the given integer ID reserved.
 *
 * @param slot_p address of a global variable storing the slot
 * @returns #FALSE on failure (no memory)
 */
dbus_bool_t
dbus_pending_call_allocate_data_slot (dbus_int32_t *slot_p)
{
    _dbus_return_val_if_fail (slot_p != NULL, FALSE);

    return _dbus_data_slot_allocator_alloc (&slot_allocator,
                                           &_DBUS_LOCK_NAME
                                           (pending_call_slots),
                                           slot_p);
}

/**
 * Deallocates a global ID for #DBusPendingCall data slots.
 * dbus_pending_call_get_data() and dbus_pending_call_set_data() may
 * no longer be used with this slot. Existing data stored on existing
 * DBusPendingCall objects will be freed when the #DBusPendingCall is
 * finalized, but may not be retrieved (and may only be replaced if
 * someone else reallocates the slot). When the refcount on the
 * passed-in slot reaches 0, it is set to -1.
 *
 * @param slot_p address storing the slot to deallocate
 */

```

```

void
dbus_pending_call_free_data_slot (dbus_int32_t *slot_p)
{
    _dbus_return_if_fail (slot_p != NULL);
    _dbus_return_if_fail (*slot_p >= 0);

    _dbus_data_slot_allocator_free (&slot_allocator, slot_p);
}

/**
 * Stores a pointer on a #DBusPendingCall, along
 * with an optional function to be used for freeing
 * the data when the data is set again, or when
 * the pending call is finalized. The slot number
 * must have been allocated with
dbus_pending_call_allocate_data_slot().
 *
 * @param pending the pending_call
 * @param slot the slot number
 * @param data the data to store
 * @param free_data_func finalizer function for the data
 * @returns #TRUE if there was enough memory to store the data
 */
dbus_bool_t
dbus_pending_call_set_data (DBusPendingCall *pending,
                           dbus_int32_t slot,
                           void *data,
                           DBusFreeFunction free_data_func)
{
    dbus_bool_t retval;

    _dbus_return_val_if_fail (pending != NULL, FALSE);
    _dbus_return_val_if_fail (slot >= 0, FALSE);

    CONNECTION_LOCK (pending->connection);
    retval = _dbus_pending_call_set_data_unlocked (pending, slot, data,
free_data_func);
    CONNECTION_UNLOCK (pending->connection);
    return retval;
}

/**
 * Retrieves data previously set with dbus_pending_call_set_data().
 * The slot must still be allocated (must not have been freed).
 *
 * @param pending the pending_call
 * @param slot the slot to get data from
 * @returns the data, or #NULL if not found
 */
void*
dbus_pending_call_get_data (DBusPendingCall *pending,

```



```

                                dbus_int32_t      slot)
{
    void *res;

    _dbus_return_val_if_fail (pending != NULL, NULL);

    CONNECTION_LOCK (pending->connection);
    res = _dbus_data_slot_list_get (&slot_allocator,
                                    &pending->slot_list,
                                    slot);
    CONNECTION_UNLOCK (pending->connection);

    return res;
}

/** @} */

```

File = dbus-pending-call.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pending-call.h Object representing a call in progress.
 *
 * Copyright (C) 2002, 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifdef !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifdef DBUS_PENDING_CALL_H

```

```

#define DBUS_PENDING_CALL_H

#include <dbus/dbus-macros.h>
#include <dbus/dbus-types.h>
#include <dbus/dbus-connection.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DBusPendingCall
 * @{
 */

#define DBUS_TIMEOUT_INFINITE ((int) 0x7fffffff)
#define DBUS_TIMEOUT_USE_DEFAULT (-1)

DBUS_EXPORT
DBusPendingCall* dbus_pending_call_ref      (DBusPendingCall
*pending);
DBUS_EXPORT
void            dbus_pending_call_unref      (DBusPendingCall
*pending);
DBUS_EXPORT
dbus_bool_t     dbus_pending_call_set_notify (DBusPendingCall
*pending,

DBusPendingCallNotifyFunction  function,

void

*user_data,

DBusFreeFunction

free_user_data);
DBUS_EXPORT
void            dbus_pending_call_cancel      (DBusPendingCall
*pending);
DBUS_EXPORT
dbus_bool_t     dbus_pending_call_get_completed (DBusPendingCall
*pending);
DBUS_EXPORT
DBusMessage*    dbus_pending_call_steal_reply (DBusPendingCall
*pending);
DBUS_EXPORT
void            dbus_pending_call_block      (DBusPendingCall
*pending);

DBUS_EXPORT
dbus_bool_t     dbus_pending_call_allocate_data_slot (dbus_int32_t
*slot_p);
DBUS_EXPORT
void            dbus_pending_call_free_data_slot (dbus_int32_t
*slot_p);
DBUS_EXPORT

```

```

dbus_bool_t dbus_pending_call_set_data      (DBusPendingCall
*pending,
                                           dbus_int32_t
slot,
                                           void
*data,
                                           DBusFreeFunction
free_data_func);
DBUS_EXPORT
void*      dbus_pending_call_get_data      (DBusPendingCall
*pending,
                                           dbus_int32_t
slot);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_PENDING_CALL_H */

```

File = dbus-pipe-unix.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pipe-unix.c unix related pipe implementation
 *
 * Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

```

```

#include <config.h>

```

```

#include "dbus-protocol.h"
#include "dbus-string.h"
#include "dbus-internals.h"
#include "dbus-pipe.h"
#include "dbus-sysdeps-unix.h"

#include <errno.h>

/**
 * write data to a pipe.
 *
 * @param pipe the pipe instance
 * @param buffer the buffer to write data from
 * @param start the first byte in the buffer to write
 * @param len the number of bytes to try to write
 * @param error error return
 * @returns the number of bytes written or -1 on error
 */
int
_dbus_pipe_write (DBusPipe      *pipe,
                  const DBusString *buffer,
                  int            start,
                  int            len,
                  DBusError      *error)
{
    int written;

    written = _dbus_write (pipe->fd, buffer, start, len);
    if (written < 0)
        {
            dbus_set_error (error, DBUS_ERROR_FAILED,
                           "Writing to pipe: %s\n",
                           _dbus_strerror (errno));
        }
    return written;
}

/**
 * close a pipe.
 *
 * @param pipe the pipe instance
 * @param error return location for an error
 * @returns -1 if error is set
 */
int
_dbus_pipe_close (DBusPipe      *pipe,
                  DBusError      *error)
{
    if (!_dbus_close (pipe->fd, error))
        {
            return -1;
        }
}

```

```
else
{
    _dbus_pipe_invalidate (pipe);
    return 0;
}
}
```

File = dbus-pipe-win.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pipe-win.c windows related pipe implementation
 *
 * Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
```

```
#include <config.h>
#include "dbus-protocol.h"
#include "dbus-string.h"
#include "dbus-internals.h"
#include "dbus-pipe.h"
```

```
#include <windows.h>
#include <io.h>
```

```
/**
 * write data to a pipe.
 *
 * @param pipe the pipe instance
 * @param buffer the buffer to write data from
```

```

* @param start the first byte in the buffer to write
* @param len the number of bytes to try to write
* @param error error return
* @returns the number of bytes written or -1 on error
*/
int
_dbus_pipe_write (DBusPipe      *pipe,
                  const DBusString *buffer,
                  int            start,
                  int            len,
                  DBusError      *error)
{
    const char *buffer_c = _dbus_string_get_const_data (buffer);
    int written;

    written = _write (pipe->fd, buffer_c + start, len);

    if (written >= 0)
        return written;

    dbus_set_error (error, _dbus_error_from_system_errno (),
                  "Writing to pipe: %s",
                  _dbus_strerror_from_errno ());
    return -1;
}

/**
 * close a pipe.
 *
 * @param pipe the pipe instance
 * @param error return location for an error
 * @returns #FALSE if error is set
 */
int
_dbus_pipe_close (DBusPipe      *pipe,
                  DBusError      *error)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (_close (pipe->fd) != 0)
    {
        dbus_set_error (error, _dbus_error_from_system_errno (),
                      "Could not close pipe fd %d: %s", pipe->fd,
                      _dbus_strerror_from_errno ());
        return -1;
    }
    else
    {
        _dbus_pipe_invalidate (pipe);
        return 0;
    }
}

```

File = dbus-pipe.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-pipe.c pipe implementation (internal to D-Bus implementation)
 *
 * Copyright (C) 2002, 2003, 2006 Red Hat, Inc.
 * Copyright (C) 2003 CodeFactory AB
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#include "dbus-pipe.h"

/*
 * init a pipe instance.
 *
 * @param pipe the pipe
 * @param fd the file descriptor to init from
 */
void
_dbus_pipe_init (DBusPipe *pipe,
                 int      fd)
{
    pipe->fd = fd;
}

/**
 * init a pipe with stdout
 *
 * @param pipe the pipe
 */
```

```

 */
void
_dbus_pipe_init_stdout (DBusPipe *pipe)
{
    _dbus_pipe_init (pipe, 1);
}

/**
 * check if a pipe is valid; pipes can be set invalid, similar to
 * a -1 file descriptor.
 *
 * @param pipe the pipe instance
 * @returns #FALSE if pipe is not valid
 */
dbus_bool_t
_dbus_pipe_is_valid(DBusPipe *pipe)
{
    return pipe->fd >= 0;
}

/**
 * Check if a pipe is stdout or stderr.
 *
 * @param pipe the pipe instance
 * @returns #TRUE if pipe is one of the standard out/err channels
 */
dbus_bool_t
_dbus_pipe_is_stdout_or_stderr (DBusPipe *pipe)
{
    return pipe->fd == 1 || pipe->fd == 2;
}

/**
 * Initializes a pipe to an invalid value.
 * @param pipe the pipe
 */
void
_dbus_pipe_invalidate (DBusPipe *pipe)
{
    pipe->fd = -1;
}

```

File = dbus-pipe.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-sysdeps.h Wrappers around system/libc features (internal to D-
Bus implementation)
 *
 * Copyright (C) 2002, 2003 Red Hat, Inc.
 * Copyright (C) 2003 CodeFactory AB

```



```

*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

```

```

#ifndef DBUS_PIPE_H
#define DBUS_PIPE_H

```

```

#ifdef HAVE_STDINT_H
#include <stdint.h>
#endif

```

```

#ifdef HAVE_INTTYPES_H
#include <inttypes.h>
#endif

```

```

#include <dbus/dbus-types.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-string.h>
#include <dbus/dbus-sysdeps.h>

```

```

struct DBusPipe {
    int fd;
};

```

```

void          _dbus_pipe_init          (DBusPipe      *pipe,
int          fd);
void          _dbus_pipe_init_stdout  (DBusPipe      *pipe);
int          _dbus_pipe_write         (DBusPipe      *pipe,
const DBusString *buffer,
int          start,
int          len,
DBusError    *error);
int          _dbus_pipe_close         (DBusPipe      *pipe,
DBusError    *error);

```

```
dbus_bool_t _dbus_pipe_is_valid          (DBusPipe      *pipe);
void        _dbus_pipe_invalidate       (DBusPipe      *pipe);
dbus_bool_t _dbus_pipe_is_stdout_or_stderr (DBusPipe      *pipe);
```

```
#endif
```

```
File = dbus-protocol.h
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-protocol.h  D-Bus protocol constants
 *
 * Copyright (C) 2002, 2003  CodeFactory AB
 * Copyright (C) 2004, 2005 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA
 */
```

```
#ifndef DBUS_PROTOCOL_H
#define DBUS_PROTOCOL_H
```

```
/* Don't include anything in here from anywhere else. It's
 * intended for use by any random library.
 */
```

```
#ifdef __cplusplus
extern "C" {
#if 0
} /* avoids confusing emacs indentation */
#endif
#endif
```

```

/* Normally docs are in .c files, but there isn't a .c file for this.
*/
/**
 * @defgroup DBusProtocol Protocol constants
 * @ingroup DBus
 *
 * @brief Defines constants which are part of the D-Bus protocol
 *
 * This header is intended for use by any library, not only libdbus.
 *
 * @{
 */

/* Message byte order */
#define DBUS_LITTLE_ENDIAN ('l') /**< Code marking LSB-first byte
order in the wire protocol. */
#define DBUS_BIG_ENDIAN ('B') /**< Code marking MSB-first byte
order in the wire protocol. */

/** Protocol version. */
#define DBUS_MAJOR_PROTOCOL_VERSION 1

/** Type code that is never equal to a legitimate type code */
#define DBUS_TYPE_INVALID ((int) '\0')
/** #DBUS_TYPE_INVALID as a string literal instead of a int literal */
#define DBUS_TYPE_INVALID_AS_STRING "\0"

/* Primitive types */
/** Type code marking an 8-bit unsigned integer */
#define DBUS_TYPE_BYTE ((int) 'y')
/** #DBUS_TYPE_BYTE as a string literal instead of a int literal */
#define DBUS_TYPE_BYTE_AS_STRING "y"
/** Type code marking a boolean */
#define DBUS_TYPE_BOOLEAN ((int) 'b')
/** #DBUS_TYPE_BOOLEAN as a string literal instead of a int literal */
#define DBUS_TYPE_BOOLEAN_AS_STRING "b"
/** Type code marking a 16-bit signed integer */
#define DBUS_TYPE_INT16 ((int) 'n')
/** #DBUS_TYPE_INT16 as a string literal instead of a int literal */
#define DBUS_TYPE_INT16_AS_STRING "n"
/** Type code marking a 16-bit unsigned integer */
#define DBUS_TYPE_UINT16 ((int) 'q')
/** #DBUS_TYPE_UINT16 as a string literal instead of a int literal */
#define DBUS_TYPE_UINT16_AS_STRING "q"
/** Type code marking a 32-bit signed integer */
#define DBUS_TYPE_INT32 ((int) 'i')
/** #DBUS_TYPE_INT32 as a string literal instead of a int literal */
#define DBUS_TYPE_INT32_AS_STRING "i"
/** Type code marking a 32-bit unsigned integer */
#define DBUS_TYPE_UINT32 ((int) 'u')
/** #DBUS_TYPE_UINT32 as a string literal instead of a int literal */

```

```

#define DBUS_TYPE_UINT32_AS_STRING      "u"
/** Type code marking a 64-bit signed integer */
#define DBUS_TYPE_INT64                ((int) 'x')
/** #DBUS_TYPE_INT64 as a string literal instead of a int literal */
#define DBUS_TYPE_INT64_AS_STRING      "x"
/** Type code marking a 64-bit unsigned integer */
#define DBUS_TYPE_UINT64               ((int) 't')
/** #DBUS_TYPE_UINT64 as a string literal instead of a int literal */
#define DBUS_TYPE_UINT64_AS_STRING     "t"
/** Type code marking an 8-byte double in IEEE 754 format */
#define DBUS_TYPE_DOUBLE               ((int) 'd')
/** #DBUS_TYPE_DOUBLE as a string literal instead of a int literal */
#define DBUS_TYPE_DOUBLE_AS_STRING     "d"
/** Type code marking a UTF-8 encoded, nul-terminated Unicode string */
#define DBUS_TYPE_STRING               ((int) 's')
/** #DBUS_TYPE_STRING as a string literal instead of a int literal */
#define DBUS_TYPE_STRING_AS_STRING     "s"
/** Type code marking a D-Bus object path */
#define DBUS_TYPE_OBJECT_PATH         ((int) 'o')
/** #DBUS_TYPE_OBJECT_PATH as a string literal instead of a int
literal */
#define DBUS_TYPE_OBJECT_PATH_AS_STRING "o"
/** Type code marking a D-Bus type signature */
#define DBUS_TYPE_SIGNATURE           ((int) 'g')
/** #DBUS_TYPE_SIGNATURE as a string literal instead of a int literal
*/
#define DBUS_TYPE_SIGNATURE_AS_STRING  "g"
/** Type code marking a unix file descriptor */
#define DBUS_TYPE_UNIX_FD             ((int) 'h')
/** #DBUS_TYPE_UNIX_FD as a string literal instead of a int literal */
#define DBUS_TYPE_UNIX_FD_AS_STRING    "h"

/* Compound types */
/** Type code marking a D-Bus array type */
#define DBUS_TYPE_ARRAY               ((int) 'a')
/** #DBUS_TYPE_ARRAY as a string literal instead of a int literal */
#define DBUS_TYPE_ARRAY_AS_STRING     "a"
/** Type code marking a D-Bus variant type */
#define DBUS_TYPE_VARIANT             ((int) 'v')
/** #DBUS_TYPE_VARIANT as a string literal instead of a int literal */
#define DBUS_TYPE_VARIANT_AS_STRING   "v"

/** STRUCT and DICT_ENTRY are sort of special since their codes can't
* appear in a type string, instead
* DBUS_STRUCT_BEGIN_CHAR/DBUS_DICT_ENTRY_BEGIN_CHAR have to appear
*/
/** Type code used to represent a struct; however, this type code does
not appear
* in type signatures, instead #DBUS_STRUCT_BEGIN_CHAR and
#DBUS_STRUCT_END_CHAR will
* appear in a signature.

```

```

*/
#define DBUS_TYPE_STRUCT          ((int) 'r')
/** #DBUS_TYPE_STRUCT as a string literal instead of a int literal */
#define DBUS_TYPE_STRUCT_AS_STRING      "r"
/** Type code used to represent a dict entry; however, this type code
does not appear
 * in type signatures, instead #DBUS_DICT_ENTRY_BEGIN_CHAR and
#DBUS_DICT_ENTRY_END_CHAR will
 * appear in a signature.
*/
#define DBUS_TYPE_DICT_ENTRY      ((int) 'e')
/** #DBUS_TYPE_DICT_ENTRY as a string literal instead of a int literal
*/
#define DBUS_TYPE_DICT_ENTRY_AS_STRING      "e"

/** Does not include #DBUS_TYPE_INVALID, #DBUS_STRUCT_BEGIN_CHAR,
#DBUS_STRUCT_END_CHAR,
 * #DBUS_DICT_ENTRY_BEGIN_CHAR, or #DBUS_DICT_ENTRY_END_CHAR - i.e. it
is the number of
 * valid types, not the number of distinct characters that may appear
in a type signature.
*/
#define DBUS_NUMBER_OF_TYPES      (16)

/* characters other than typecodes that appear in type signatures */

/** Code marking the start of a struct type in a type signature */
#define DBUS_STRUCT_BEGIN_CHAR      ((int) '(')
/** #DBUS_STRUCT_BEGIN_CHAR as a string literal instead of a int
literal */
#define DBUS_STRUCT_BEGIN_CHAR_AS_STRING      "("
/** Code marking the end of a struct type in a type signature */
#define DBUS_STRUCT_END_CHAR        ((int) ')')
/** #DBUS_STRUCT_END_CHAR a string literal instead of a int literal */
#define DBUS_STRUCT_END_CHAR_AS_STRING      ")"
/** Code marking the start of a dict entry type in a type signature */
#define DBUS_DICT_ENTRY_BEGIN_CHAR  ((int) '{')
/** #DBUS_DICT_ENTRY_BEGIN_CHAR as a string literal instead of a int
literal */
#define DBUS_DICT_ENTRY_BEGIN_CHAR_AS_STRING      "{"
/** Code marking the end of a dict entry type in a type signature */
#define DBUS_DICT_ENTRY_END_CHAR    ((int) '}')
/** #DBUS_DICT_ENTRY_END_CHAR as a string literal instead of a int
literal */
#define DBUS_DICT_ENTRY_END_CHAR_AS_STRING      "}"

/** Max length in bytes of a bus name, interface, or member (not
object
 * path, paths are unlimited). This is limited because lots of stuff
 * is O(n) in this number, plus it would be obnoxious to type in a
 * paragraph-long method name so most likely something like that would
 * be an exploit.

```

```

*/
#define DBUS_MAXIMUM_NAME_LENGTH 255

/** This one is 255 so it fits in a byte */
#define DBUS_MAXIMUM_SIGNATURE_LENGTH 255

/** Max length of a match rule string; to keep people from hosing the
 * daemon with some huge rule
 */
#define DBUS_MAXIMUM_MATCH_RULE_LENGTH 1024

/** Max arg number you can match on in a match rule, e.g.
 * arg0='hello' is OK, arg3489720987='hello' is not
 */
#define DBUS_MAXIMUM_MATCH_RULE_ARG_NUMBER 63

/** Max length of a marshaled array in bytes (64M, 2^26) We use signed
 * int for lengths so must be INT_MAX or less. We need something a
 * bit smaller than INT_MAX because the array is inside a message with
 * header info, etc. so an INT_MAX array wouldn't allow the message
 * overhead. The 64M number is an attempt at a larger number than
 * we'd reasonably ever use, but small enough that your bus would chew
 * through it fairly quickly without locking up forever. If you have
 * data that's likely to be larger than this, you should probably be
 * sending it in multiple incremental messages anyhow.
 */
#define DBUS_MAXIMUM_ARRAY_LENGTH (67108864)
/** Number of bits you need in an unsigned to store the max array size
 */
#define DBUS_MAXIMUM_ARRAY_LENGTH_BITS 26

/** The maximum total message size including header and body; similar
 * rationale to max array size.
 */
#define DBUS_MAXIMUM_MESSAGE_LENGTH (DBUS_MAXIMUM_ARRAY_LENGTH * 2)
/** Number of bits you need in an unsigned to store the max message
size */
#define DBUS_MAXIMUM_MESSAGE_LENGTH_BITS 27

/** The maximum total number of unix fds in a message. Similar
 * rationale as DBUS_MAXIMUM_MESSAGE_LENGTH. However we divide by four
 * given that one fd is an int and hence at least 32 bits.
 */
#define DBUS_MAXIMUM_MESSAGE_UNIX_FDS (DBUS_MAXIMUM_MESSAGE_LENGTH/4)
/** Number of bits you need in an unsigned to store the max message
unix fds */
#define DBUS_MAXIMUM_MESSAGE_UNIX_FDS_BITS
(DBUS_MAXIMUM_MESSAGE_LENGTH_BITS-2)

/** Depth of recursion in the type tree. This is automatically limited
 * to DBUS_MAXIMUM_SIGNATURE_LENGTH since you could only have an array
 * of array of array of ... that fit in the max signature. But that's

```

```

    * probably a bit too large.
    */
#define DBUS_MAXIMUM_TYPE_RECURSION_DEPTH 32

/* Types of message */

/** This value is never a valid message type, see
dbus_message_get_type() */
#define DBUS_MESSAGE_TYPE_INVALID 0
/** Message type of a method call message, see dbus_message_get_type()
*/
#define DBUS_MESSAGE_TYPE_METHOD_CALL 1
/** Message type of a method return message, see
dbus_message_get_type() */
#define DBUS_MESSAGE_TYPE_METHOD_RETURN 2
/** Message type of an error reply message, see
dbus_message_get_type() */
#define DBUS_MESSAGE_TYPE_ERROR 3
/** Message type of a signal message, see dbus_message_get_type() */
#define DBUS_MESSAGE_TYPE_SIGNAL 4

#define DBUS_NUM_MESSAGE_TYPES 5

/* Header flags */

/** If set, this flag means that the sender of a message does not care
about getting
* a reply, so the recipient need not send one. See
dbus_message_set_no_reply().
*/
#define DBUS_HEADER_FLAG_NO_REPLY_EXPECTED 0x1
/**
* If set, this flag means that even if the message bus knows how to
start an owner for
* the destination bus name (see dbus_message_set_destination()), it
should not
* do so. If this flag is not set, the bus may launch a program to
process the
* message.
*/
#define DBUS_HEADER_FLAG_NO_AUTO_START 0x2

/* Header fields */

/** Not equal to any valid header field code */
#define DBUS_HEADER_FIELD_INVALID 0
/** Header field code for the path - the path is the object emitting a
signal or the object receiving a method call.
* See dbus_message_set_path().
*/
#define DBUS_HEADER_FIELD_PATH 1

```

```

/** Header field code for the interface containing a member (method or
signal).
 * See dbus_message_set_interface().
 */
#define DBUS_HEADER_FIELD_INTERFACE      2
/** Header field code for a member (method or signal). See
dbus_message_set_member(). */
#define DBUS_HEADER_FIELD_MEMBER        3
/** Header field code for an error name (found in
#DBUS_MESSAGE_TYPE_ERROR messages).
 * See dbus_message_set_error_name().
 */
#define DBUS_HEADER_FIELD_ERROR_NAME     4
/** Header field code for a reply serial, used to match a
#DBUS_MESSAGE_TYPE_METHOD_RETURN message with the
 * message that it's a reply to. See dbus_message_set_reply_serial().
 */
#define DBUS_HEADER_FIELD_REPLY_SERIAL   5
/**
 * Header field code for the destination bus name of a message. See
dbus_message_set_destination().
 */
#define DBUS_HEADER_FIELD_DESTINATION    6
/**
 * Header field code for the sender of a message; usually initialized
by the message bus.
 * See dbus_message_set_sender().
 */
#define DBUS_HEADER_FIELD_SENDER         7
/**
 * Header field code for the type signature of a message.
 */
#define DBUS_HEADER_FIELD_SIGNATURE      8
/**
 * Header field code for the number of unix file descriptors
associated
 * with this message.
 */
#define DBUS_HEADER_FIELD_UNIX_FDS       9

/**
 * Value of the highest-numbered header field code, can be used to
determine
 * the size of an array indexed by header field code. Remember though
 * that unknown codes must be ignored, so check for that before
 * indexing the array.
 */
#define DBUS_HEADER_FIELD_LAST DBUS_HEADER_FIELD_UNIX_FDS

/** Header format is defined as a signature:
 *   byte                               byte order

```



```

*   byte                message type ID
*   byte                flags
*   byte                protocol version
*   uint32              body length
*   uint32              serial
*   array of struct (byte,variant) (field name, value)
*
* The length of the header can be computed as the
* fixed size of the initial data, plus the length of
* the array at the end, plus padding to an 8-boundary.
*/
#define DBUS_HEADER_SIGNATURE \
    DBUS_TYPE_BYTE_AS_STRING \
    DBUS_TYPE_BYTE_AS_STRING \
    DBUS_TYPE_BYTE_AS_STRING \
    DBUS_TYPE_BYTE_AS_STRING \
    DBUS_TYPE_UINT32_AS_STRING \
    DBUS_TYPE_UINT32_AS_STRING \
    DBUS_TYPE_ARRAY_AS_STRING \
    DBUS_STRUCT_BEGIN_CHAR_AS_STRING \
    DBUS_TYPE_BYTE_AS_STRING \
    DBUS_TYPE_VARIANT_AS_STRING \
    DBUS_STRUCT_END_CHAR_AS_STRING

/**
 * The smallest header size that can occur. (It won't be valid due to
 * missing required header fields.) This is 4 bytes, two uint32, an
 * array length. This isn't any kind of resource limit, just the
 * necessary/logical outcome of the header signature.
 */
#define DBUS_MINIMUM_HEADER_SIZE 16

/* Errors */
/* WARNING these get autoconverted to an enum in dbus-glib.h. Thus,
 * if you change the order it breaks the ABI. Keep them in order.
 * Also, don't change the formatting since that will break the sed
 * script.
 */
/** A generic error; "something went wrong" - see the error message
for more. */
#define DBUS_ERROR_FAILED
"org.freedesktop.DBus.Error.Failed"
/** There was not enough memory to complete an operation. */
#define DBUS_ERROR_NO_MEMORY
"org.freedesktop.DBus.Error.NoMemory"
/** The bus doesn't know how to launch a service to supply the bus
name you wanted. */
#define DBUS_ERROR_SERVICE_UNKNOWN
"org.freedesktop.DBus.Error.ServiceUnknown"
/** The bus name you referenced doesn't exist (i.e. no application
owns it). */

```

```

#define DBUS_ERROR_NAME_HAS_NO_OWNER
"org.freedesktop.DBus.Error.NameHasNoOwner"
/** No reply to a message expecting one, usually means a timeout
occurred. */
#define DBUS_ERROR_NO_REPLY
"org.freedesktop.DBus.Error.NoReply"
/** Something went wrong reading or writing to a socket, for example.
*/
#define DBUS_ERROR_IO_ERROR
"org.freedesktop.DBus.Error.IOError"
/** A D-Bus bus address was malformed. */
#define DBUS_ERROR_BAD_ADDRESS
"org.freedesktop.DBus.Error.BadAddress"
/** Requested operation isn't supported (like ENOSYS on UNIX). */
#define DBUS_ERROR_NOT_SUPPORTED
"org.freedesktop.DBus.Error.NotSupported"
/** Some limited resource is exhausted. */
#define DBUS_ERROR_LIMITS_EXCEEDED
"org.freedesktop.DBus.Error.LimitsExceeded"
/** Security restrictions don't allow doing what you're trying to do.
*/
#define DBUS_ERROR_ACCESS_DENIED
"org.freedesktop.DBus.Error.AccessDenied"
/** Authentication didn't work. */
#define DBUS_ERROR_AUTH_FAILED
"org.freedesktop.DBus.Error.AuthFailed"
/** Unable to connect to server (probably caused by ECONNREFUSED on a
socket). */
#define DBUS_ERROR_NO_SERVER
"org.freedesktop.DBus.Error.NoServer"
/** Certain timeout errors, possibly ETIMEDOUT on a socket.
 * Note that #DBUS_ERROR_NO_REPLY is used for message reply timeouts.
 * @warning this is confusingly-named given that #DBUS_ERROR_TIMED_OUT
also exists. We can't fix
 * it for compatibility reasons so just be careful.
*/
#define DBUS_ERROR_TIMEOUT
"org.freedesktop.DBus.Error.Timeout"
/** No network access (probably ENETUNREACH on a socket). */
#define DBUS_ERROR_NO_NETWORK
"org.freedesktop.DBus.Error.NoNetwork"
/** Can't bind a socket since its address is in use (i.e. EADDRINUSE).
*/
#define DBUS_ERROR_ADDRESS_IN_USE
"org.freedesktop.DBus.Error.AddressInUse"
/** The connection is disconnected and you're trying to use it. */
#define DBUS_ERROR_DISCONNECTED
"org.freedesktop.DBus.Error.Disconnected"
/** Invalid arguments passed to a method call. */
#define DBUS_ERROR_INVALID_ARGS
"org.freedesktop.DBus.Error.InvalidArgs"
/** Missing file. */

```

```

#define DBUS_ERROR_FILE_NOT_FOUND
"org.freedesktop.DBus.Error.FileNotFound"
/** Existing file and the operation you're using does not silently
overwrite. */
#define DBUS_ERROR_FILE_EXISTS
"org.freedesktop.DBus.Error.FileExists"
/** Method name you invoked isn't known by the object you invoked it
on. */
#define DBUS_ERROR_UNKNOWN_METHOD
"org.freedesktop.DBus.Error.UnknownMethod"
/** Object you invoked a method on isn't known. */
#define DBUS_ERROR_UNKNOWN_OBJECT
"org.freedesktop.DBus.Error.UnknownObject"
/** Interface you invoked a method on isn't known by the object. */
#define DBUS_ERROR_UNKNOWN_INTERFACE
"org.freedesktop.DBus.Error.UnknownInterface"
/** Property you tried to access isn't known by the object. */
#define DBUS_ERROR_UNKNOWN_PROPERTY
"org.freedesktop.DBus.Error.UnknownProperty"
/** Property you tried to set is read-only. */
#define DBUS_ERROR_PROPERTY_READ_ONLY
"org.freedesktop.DBus.Error.PropertyReadOnly"
/** Certain timeout errors, e.g. while starting a service.
 * @warning this is confusingly-named given that #DBUS_ERROR_TIMEOUT
also exists. We can't fix
 * it for compatibility reasons so just be careful.
 */
#define DBUS_ERROR_TIMED_OUT
"org.freedesktop.DBus.Error.TimedOut"
/** Tried to remove or modify a match rule that didn't exist. */
#define DBUS_ERROR_MATCH_RULE_NOT_FOUND
"org.freedesktop.DBus.Error.MatchRuleNotFound"
/** The match rule isn't syntactically valid. */
#define DBUS_ERROR_MATCH_RULE_INVALID
"org.freedesktop.DBus.Error.MatchRuleInvalid"
/** While starting a new process, the exec() call failed. */
#define DBUS_ERROR_SPAWN_EXEC_FAILED
"org.freedesktop.DBus.Error.Spawn.ExecFailed"
/** While starting a new process, the fork() call failed. */
#define DBUS_ERROR_SPAWN_FORK_FAILED
"org.freedesktop.DBus.Error.Spawn.ForkFailed"
/** While starting a new process, the child exited with a status code.
 */
#define DBUS_ERROR_SPAWN_CHILD_EXITED
"org.freedesktop.DBus.Error.Spawn.ChildExited"
/** While starting a new process, the child exited on a signal. */
#define DBUS_ERROR_SPAWN_CHILD_SIGNALED
"org.freedesktop.DBus.Error.Spawn.ChildSignaled"
/** While starting a new process, something went wrong. */
#define DBUS_ERROR_SPAWN_FAILED
"org.freedesktop.DBus.Error.Spawn.Failed"
/** We failed to setup the environment correctly. */

```

```

#define DBUS_ERROR_SPAWN_SETUP_FAILED
"org.freedesktop.DBus.Error.Spawn.FailedToSetup"
/** We failed to setup the config parser correctly. */
#define DBUS_ERROR_SPAWN_CONFIG_INVALID
"org.freedesktop.DBus.Error.Spawn.ConfigInvalid"
/** Bus name was not valid. */
#define DBUS_ERROR_SPAWN_SERVICE_INVALID
"org.freedesktop.DBus.Error.Spawn.ServiceNotValid"
/** Service file not found in system-services directory. */
#define DBUS_ERROR_SPAWN_SERVICE_NOT_FOUND
"org.freedesktop.DBus.Error.Spawn.ServiceNotFound"
/** Permissions are incorrect on the setuid helper. */
#define DBUS_ERROR_SPAWN_PERMISSIONS_INVALID
"org.freedesktop.DBus.Error.Spawn.PermissionsInvalid"
/** Service file invalid (Name, User or Exec missing). */
#define DBUS_ERROR_SPAWN_FILE_INVALID
"org.freedesktop.DBus.Error.Spawn.FileInvalid"
/** Tried to get a UNIX process ID and it wasn't available. */
#define DBUS_ERROR_SPAWN_NO_MEMORY
"org.freedesktop.DBus.Error.Spawn.NoMemory"
/** Tried to get a UNIX process ID and it wasn't available. */
#define DBUS_ERROR_UNIX_PROCESS_ID_UNKNOWN
"org.freedesktop.DBus.Error.UnixProcessIdUnknown"
/** A type signature is not valid. */
#define DBUS_ERROR_INVALID_SIGNATURE
"org.freedesktop.DBus.Error.InvalidSignature"
/** A file contains invalid syntax or is otherwise broken. */
#define DBUS_ERROR_INVALID_FILE_CONTENT
"org.freedesktop.DBus.Error.InvalidFileContent"
/** Asked for SELinux security context and it wasn't available. */
#define DBUS_ERROR_SELINUX_SECURITY_CONTEXT_UNKNOWN
"org.freedesktop.DBus.Error.SELinuxSecurityContextUnknown"
/** Asked for ADT audit data and it wasn't available. */
#define DBUS_ERROR_ADT_AUDIT_DATA_UNKNOWN
"org.freedesktop.DBus.Error.AdtAuditDataUnknown"
/** There's already an object with the requested object path. */
#define DBUS_ERROR_OBJECT_PATH_IN_USE
"org.freedesktop.DBus.Error.ObjectPathInUse"
/** The message meta data does not match the payload. e.g. expected
    number of file descriptors were not sent over the socket this
message was received on. */
#define DBUS_ERROR_INCONSISTENT_MESSAGE
"org.freedesktop.DBus.Error.InconsistentMessage"

/* XML introspection format */

/** XML namespace of the introspection format version 1.0 */
#define DBUS_INTROSPECT_1_0_XML_NAMESPACE
"http://www.freedesktop.org/standards/dbus"
/** XML public identifier of the introspection format version 1.0 */
#define DBUS_INTROSPECT_1_0_XML_PUBLIC_IDENTIFIER "-//freedesktop//DTD
D-BUS Object Introspection 1.0//EN"

```

```

/** XML system identifier of the introspection format version 1.0 */
#define DBUS_INTROSPECT_1_0_XML_SYSTEM_IDENTIFIER
"http://www.freedesktop.org/standards/dbus/1.0/introspect.dtd"
/** XML document type declaration of the introspection format version
1.0 */
#define DBUS_INTROSPECT_1_0_XML_DOCTYPE_DECL_NODE "<!DOCTYPE node
PUBLIC \"\" DBUS_INTROSPECT_1_0_XML_PUBLIC_IDENTIFIER \"\"\n\"\"
DBUS_INTROSPECT_1_0_XML_SYSTEM_IDENTIFIER \"\">\n"

/** @} */

#ifdef __cplusplus
#if 0
{ /* avoids confusing emacs indentation */
#endif
}
#endif

#endif /* DBUS_PROTOCOL_H */

File = dbus-resources.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-resources.c Resource tracking/limits
*
* Copyright (C) 2003 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>
#include <dbus/dbus-resources.h>

```

```

#include <dbus/dbus-internals.h>

/**
 * @defgroup DBusResources Resource limits related code
 * @ingroup DBusInternals
 * @brief DBusCounter and other stuff related to resource limits
 *
 * Types and functions related to tracking resource limits,
 * such as the maximum amount of memory/unix fds a connection can use
 * for messages, etc.
 */

/**
 * @defgroup DBusResourcesInternals Resource limits implementation
 * details
 * @ingroup DBusInternals
 * @brief Resource limits implementation details
 *
 * Implementation details of resource limits code.
 *
 * @{
 */

/**
 * @brief Internals of DBusCounter.
 *
 * DBusCounter internals. DBusCounter is an opaque object, it must be
 * used via accessor functions.
 */
struct DBusCounter
{
    int refcount; /**< reference count */

    long size_value; /**< current size counter value */
    long unix_fd_value; /**< current unix fd counter value */

#ifdef DBUS_ENABLE_STATS
    long peak_size_value; /**< largest ever size counter value */
    long peak_unix_fd_value; /**< largest ever unix fd counter value */
#endif

    long notify_size_guard_value; /**< call notify function when
    crossing this size value */
    long notify_unix_fd_guard_value; /**< call notify function when
    crossing this unix fd value */

    DBusCounterNotifyFunction notify_function; /**< notify function */
    void *notify_data; /**< data for notify function */
    dbus_bool_t notify_pending : 1; /**< TRUE if the guard value has
    been crossed */
};

```

```

/** @} */ /* end of resource limits internals docs */

/**
 * @addtogroup DBusResources
 * @{
 */

/**
 * Creates a new DBusCounter. DBusCounter is used
 * to count usage of some resource such as memory.
 *
 * @returns new counter or #NULL on failure
 */
DBusCounter*
_dbus_counter_new (void)
{
    DBusCounter *counter;

    counter = dbus_new0 (DBusCounter, 1);
    if (counter == NULL)
        return NULL;

    counter->refcount = 1;

    return counter;
}

/**
 * Increments refcount of the counter
 *
 * @param counter the counter
 * @returns the counter
 */
DBusCounter *
_dbus_counter_ref (DBusCounter *counter)
{
    _dbus_assert (counter->refcount > 0);

    counter->refcount += 1;

    return counter;
}

/**
 * Decrements refcount of the counter and possibly
 * finalizes the counter.
 *
 * @param counter the counter
 */
void
_dbus_counter_unref (DBusCounter *counter)
{

```

```

_dbus_assert (counter->refcount > 0);

counter->refcount -= 1;

if (counter->refcount == 0)
{
    dbus_free (counter);
}
}

/**
 * Adjusts the value of the size counter by the given
 * delta which may be positive or negative.
 *
 * This function may be called with locks held. After calling it, when
 * any relevant locks are no longer held you must call
_dbus_counter_notify().
 *
 * @param counter the counter
 * @param delta value to add to the size counter's current value
 */
void
_dbus_counter_adjust_size (DBusCounter *counter,
                           long         delta)
{
    long old = counter->size_value;

    counter->size_value += delta;

#ifdef DBUS_ENABLE_STATS
    if (counter->peak_size_value < counter->size_value)
        counter->peak_size_value = counter->size_value;
#endif

#ifdef 0
    _dbus_verbose ("Adjusting counter %ld by %ld = %ld\n",
                  old, delta, counter->size_value);
#endif

    if (counter->notify_function != NULL &&
        ((old < counter->notify_size_guard_value &&
         counter->size_value >= counter->notify_size_guard_value) ||
         (old >= counter->notify_size_guard_value &&
          counter->size_value < counter->notify_size_guard_value)))
        counter->notify_pending = TRUE;
}

/**
 * Calls the notify function from _dbus_counter_set_notify(),
 * if that function has been specified and the counter has crossed the

```



```

    * guard value (in either direction) since the last call to this
    function.
    *
    * This function must not be called with locks held, since it can call
    out
    * to user code.
    */
void
_dbus_counter_notify (DBusCounter *counter)
{
    if (counter->notify_pending)
    {
        counter->notify_pending = FALSE;
        (* counter->notify_function) (counter, counter->notify_data);
    }
}

/**
 * Adjusts the value of the unix fd counter by the given
 * delta which may be positive or negative.
 *
 * This function may be called with locks held. After calling it, when
 * any relevant locks are no longer held you must call
 * _dbus_counter_notify().
 *
 * @param counter the counter
 * @param delta value to add to the unix fds counter's current value
 */
void
_dbus_counter_adjust_unix_fd (DBusCounter *counter,
                              long          delta)
{
    long old = counter->unix_fd_value;

    counter->unix_fd_value += delta;

#ifdef DBUS_ENABLE_STATS
    if (counter->peak_unix_fd_value < counter->unix_fd_value)
        counter->peak_unix_fd_value = counter->unix_fd_value;
#endif

#ifdef 0
    _dbus_verbose ("Adjusting counter %ld by %ld = %ld\n",
                  old, delta, counter->unix_fd_value);
#endif

    if (counter->notify_function != NULL &&
        ((old < counter->notify_unix_fd_guard_value &&
         counter->unix_fd_value >= counter->notify_unix_fd_guard_value)
        ||
         (old >= counter->notify_unix_fd_guard_value &&

```

```

        counter->unix_fd_value < counter-
>notify_unix_fd_guard_value)))
        counter->notify_pending = TRUE;
    }

/**
 * Gets the current value of the size counter.
 *
 * @param counter the counter
 * @returns its current size value
 */
long
_dbus_counter_get_size_value (DBusCounter *counter)
{
    return counter->size_value;
}

/**
 * Gets the current value of the unix fd counter.
 *
 * @param counter the counter
 * @returns its current unix fd value
 */
long
_dbus_counter_get_unix_fd_value (DBusCounter *counter)
{
    return counter->unix_fd_value;
}

/**
 * Sets the notify function for this counter; the notify function is
 * called whenever the counter's values cross the guard values in
 * either direction (moving up, or moving down).
 *
 * @param counter the counter
 * @param size_guard_value the value we're notified if the size
counter crosses
 * @param unix_fd_guard_value the value we're notified if the unix fd
counter crosses
 * @param function function to call in order to notify
 * @param user_data data to pass to the function
 */
void
_dbus_counter_set_notify (DBusCounter *counter,
                        long size_guard_value,
                        long
unix_fd_guard_value,
                        DBusCounterNotifyFunction function,
                        void *user_data)
{
    counter->notify_size_guard_value = size_guard_value;
    counter->notify_unix_fd_guard_value = unix_fd_guard_value;
}

```

```

    counter->notify_function = function;
    counter->notify_data = user_data;
    counter->notify_pending = FALSE;
}

#ifdef DBUS_ENABLE_STATS
long
_dbus_counter_get_peak_size_value (DBusCounter *counter)
{
    return counter->peak_size_value;
}

long
_dbus_counter_get_peak_unix_fd_value (DBusCounter *counter)
{
    return counter->peak_unix_fd_value;
}
#endif

/** @} */ /* end of resource limits exported API */

```

File = dbus-resources.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-resources.h Resource tracking/limits
 *
 * Copyright (C) 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#endif DBUS_RESOURCES_H
#define DBUS_RESOURCES_H

```

```

#include <dbus/dbus-macros.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-connection.h>

DBUS_BEGIN_DECLS

typedef struct DBusCounter DBusCounter;

typedef void (* DBusCounterNotifyFunction) (DBusCounter *counter,
                                             void *user_data);

DBusCounter* _dbus_counter_new      (void);
DBusCounter* _dbus_counter_ref      (DBusCounter *counter);
void          _dbus_counter_unref    (DBusCounter *counter);

void          _dbus_counter_adjust_size      (DBusCounter *counter,
                                             long          delta);
void          _dbus_counter_adjust_unix_fd  (DBusCounter *counter,
                                             long          delta);
void          _dbus_counter_notify         (DBusCounter *counter);
long         _dbus_counter_get_size_value  (DBusCounter *counter);
long         _dbus_counter_get_unix_fd_value (DBusCounter *counter);

void _dbus_counter_set_notify      (DBusCounter *counter,
                                   long          size_guard_value,
                                   long          unix_fd_guard_value,
                                   DBusCounterNotifyFunction function,
                                   void          *user_data);

/* if DBUS_ENABLE_STATS */
long _dbus_counter_get_peak_size_value  (DBusCounter *counter);
long _dbus_counter_get_peak_unix_fd_value (DBusCounter *counter);

DBUS_END_DECLS

#endif /* DBUS_RESOURCES_H */

File = dbus-send.1

.\"
.\" dbus\--send manual page.
.\" Copyright (C) 2003 Red Hat, Inc.
.\"
.TH dbus\--send 1
.SH NAME
dbus\--send \-- Send a message to a message bus

```

.SH SYNOPSIS

.PP

.B dbus\--send

[\fB\--system\fP | \fB\--session\fP]

[\fB\--dest=\fINAME\fP]

[\fB\--print\--reply\fP[\fB=literal\fP]]

[\fB\--reply\--timeout=\fIMSEC\fP]

[\fB\--type=\fITYPE\fP]

\fIOBJECT_PATH\fP \fIINTERFACE\fB.\fIMEMBER\fP [\fICONTENTS\fP ...]

.SH DESCRIPTION

The `\fIdbus\--send\fP` command is used to send a message to a D\--Bus message bus. See <http://www.freedesktop.org/software/dbus/> for more information about the big picture.

.PP

There are two well\--known message buses: the systemwide message bus (installed on many systems as the "messagebus" service) and the per\--user\--login\--session message bus (started each time a user logs in).

The `\fB\--system\fP` and `\fB\--session\fP` options direct `\fBdbus\--send\fP` to send messages to the system or session buses respectively.

If neither is specified, `\fBdbus\--send\fP` sends to the session bus.

.PP

Nearly all uses of `\fBdbus\--send\fP` must provide the `\fB\--dest\fP` argument which is the name of a connection on the bus to send the message to. If

`\fB\--dest\fP` is omitted, no destination is set.

.PP

The object path and the name of the message to send must always be specified. Following arguments, if any, are the message contents (message arguments). These are given as `type\--specified` values and may include containers (arrays, dicts, and variants) as described below.

.nf

```
<contents> ::= <item> | <container> [ <item> | <container>... ]
<item>      ::= <type>:<value>
<container> ::= <array> | <dict> | <variant>
<array>     ::= array:<type>:<value>[,<value>...]
<dict>      ::= dict:<type>:<type>:<key>,<value>[,<key>,<value>...]
<variant>   ::= variant:<type>:<value>
<type>      ::= string | int16 | uint 16 | int32 | uint32 | int64 |
uint64 | double | byte | boolean | objpath
```

.fi

D\bus supports more types than these, but `\fBdbus\send\FP` currently does not. Also, `\fBdbus\send\FP` does not permit empty containers or nested containers (e.g. arrays of variants).

.PP

Here is an example invocation:

.nf

```
dbus\send \-\-dest=org.freedesktop.ExampleName           \\  
    /org/freedesktop/sample/object/name                 \\  
    org.freedesktop.ExampleInterface.ExampleMethod      \\  
    int32:47 string:'hello world' double:65.32          \\  
    array:string:"1st item","next item","last item"     \\  
    dict:string:int32:"one",1,"two",2,"three",3         \\  
    variant:int32:\-8                                    \\  
    objpath:/org/freedesktop/sample/object/name
```

.fi

Note that the interface is separated from a method or signal name by a dot, though in the actual protocol the interface and the interface member are separate fields.

.SH OPTIONS

The following options are supported:

.TP

.BI \-\-dest= NAME

Specify the name of the connection to receive the message.

.TP

.B "\-\-print\reply"

Block for a reply to the message sent, and print any reply received in a human-readable form.

.TP

.B "\-\-print\reply=literal"

Block for a reply to the message sent, and print the body of the reply. If the reply is an object path or a string, it is printed literally, with no punctuation, escape characters etc.

.TP

.BI \-\-reply\timeout= MSEC

Wait for a reply for up to `\fIMSEC\FP` milliseconds.

The default is implementation\(\hydefined, typically 25 seconds.

.TP

.B "\-\-system"

Send to the system message bus.

.TP

.B "\-\-session"

Send to the session message bus. (This is the default.)

.TP

.BI \-\-type= TYPE

Specify `\fBmethod_call\FP` or `\fBsignal\FP` (defaults to `"\fBsignal\FP"`).

.SH AUTHOR
dbus\-send was written by Philip Blundell.

.SH BUGS
Please send bug reports to the D\ -Bus mailing list or bug tracker,
see <http://www.freedesktop.org/software/dbus/>

File = dbus-send.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
    "http://www.oasis-
open.org/docbook/xml/4.1.2/docbookx.dtd">
<!-- lifted from troff+man by doclifter -->
<refentry id='dbussend1'>
<!-- -->
<!-- dbus\ -send manual page. -->
<!-- Copyright (C) 2003 Red Hat, Inc. -->

<refmeta>
<refentrytitle>dbus-send</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>
<refnamediv id='name'>
<refname>dbus-send</refname>
<refpurpose>Send a message to a message bus</refpurpose>
</refnamediv>
<!-- body begins here -->
<refsynopsisdiv id='synopsis'>
<cmdsynopsis>
  <command>dbus-send</command>
  <group choice='opt'><arg choice='plain'>--system </arg><arg
choice='plain'>--session </arg></group>
  <arg choice='opt'>--dest=<replaceable>NAME</replaceable></arg>
  <arg choice='opt'>--print-reply </arg>
  <arg choice='opt'>--type=<replaceable>TYPE</replaceable></arg>
  <arg
choice='plain'><replaceable>&lt;destination</replaceable></arg>
  <arg choice='plain'><replaceable>object</replaceable></arg>
  <arg choice='plain'><replaceable>path&gt;</replaceable></arg>
  <arg choice='plain'><replaceable>&lt;message</replaceable></arg>
  <arg choice='plain'><replaceable>name&gt;</replaceable></arg>
  <arg choice='opt'
rep='repeat'><replaceable>contents</replaceable></arg>
  <sbr/>
</cmdsynopsis>
</refsynopsisdiv>

<refsect1 id='description'><title>DESCRIPTION</title>
```

The `dbus-send` command is used to send a message to a D-Bus message bus. See <http://www.freedesktop.org/software/dbus/> for more information about the big picture.

There are two well-known message buses: the systemwide message bus (installed on many systems as the "messagebus" service) and the per-user-login-session message bus (started each time a user logs in). The `--system` and `--session` options direct `dbus-send` to send messages to the system or session buses respectively. If neither is specified, `dbus-send` sends to the session bus.

Nearly all uses of `dbus-send` must provide the `--dest` argument which is the name of a connection on the bus to send the message to. If `--dest` is omitted, no destination is set.

The object path and the name of the message to send must always be specified. Following arguments, if any, are the message contents (message arguments). These are given as type-specified values and may include containers (arrays, dicts, and variants) as described below.

```
<literallayout remap='.nf'>
<contents> ::= <item> | <container> [ <item>
| <container>...]
<item> ::= <type>:<value>;
<container> ::= <array> | <dict> | <variant>;
<array> ::=
array:<type>:<value>[,<value>...]
<dict> ::=
dict:<type>:<type>:<key>,<value>[,<key>,<value>...]
<variant> ::= variant:<type>:<value>;
<type> ::= string | int16 | uint 16 | int32 | uint32 |
int64 | uint64 | double | byte | boolean | objpath
</literallayout> <!-- .fi -->
```

D-Bus supports more types than these, but `dbus-send` currently does not. Also, `dbus-send` does not permit empty containers

or nested containers (e.g. arrays of variants).</para>

<para>Here is an example invocation:</para>

<literallayout remap='.nf'>

```
dbus-send --dest=org.freedesktop.ExampleName          \  
          /org/freedesktop/sample/object/name          \  
          org.freedesktop.ExampleInterface.ExampleMethod \  
          int32:47 string:'hello world' double:65.32   \  
          array:string:"1st item","next item","last item" \  
          dict:string:int32:"one",1,"two",2,"three",3   \  
          variant:int32:-8                              \  
          objpath:/org/freedesktop/sample/object/name
```

</literallayout> <!-- .fi -->

<para>Note that the interface is separated from a method or signal name by a dot, though in the actual protocol the interface and the interface member are separate fields.</para>

</refsect1>

<refsect1 id='options'><title>OPTIONS</title>

<para>The following options are supported:</para>

<variablelist remap='TP'>

<varlistentry>

<term><option>--dest=NAME</option></term>

<listitem>

<para>Specify the name of the connection to receive the message.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--print-reply</option></term>

<listitem>

<para>Block for a reply to the message sent, and print any reply received.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--system</option></term>

<listitem>

<para>Send to the system message bus.</para>

</listitem>

</varlistentry>

<varlistentry>

<term><option>--session</option></term>

<listitem>

<para>Send to the session message bus. (This is the default.)</para>

</listitem>

</varlistentry>

```
<varlistentry>
  <term><option>--type=TYPE</option></term>
  <listitem>
<para>Specify "method_call" or "signal" (defaults to "signal").</para>

  </listitem>
</varlistentry>
</variablelist>
</refsect1>

<refsect1 id='author'><title>AUTHOR</title>
<para>dbus-send was written by Philip Blundell.</para>

</refsect1>

<refsect1 id='bugs'><title>BUGS</title>
<para>Please send bug reports to the D-Bus mailing list or bug
tracker,
see <ulink
url='http://www.freedesktop.org/software/dbus/'>http://www.freedesktop
.org/software/dbus/</ulink></para>
</refsect1>
</refentry>
```

File = dbus-server-debug-pipe.c

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-debug-pipe.c In-proc debug server implementation
 *
 * Copyright (C) 2003 CodeFactory AB
 * Copyright (C) 2003, 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
```

```
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
```

```
#include <config.h>
#include "dbus-internals.h"
#include "dbus-server-debug-pipe.h"
#include "dbus-transport-socket.h"
#include "dbus-connection-internal.h"
#include "dbus-hash.h"
#include "dbus-string.h"
#include "dbus-protocol.h"
```

```
#ifdef DBUS_BUILD_TESTS
```

```
/**
 * @defgroup DBusServerDebugPipe DBusServerDebugPipe
 * @ingroup DBusInternals
 * @brief In-process pipe debug server used in unit tests.
 *
 * Types and functions related to DBusServerDebugPipe.
 * This is used for unit testing.
 *
 * @{
 */
```

```
/**
 * Opaque object representing a debug server implementation.
 */
```

```
typedef struct DBusServerDebugPipe DBusServerDebugPipe;
```

```
/**
 * Implementation details of DBusServerDebugPipe. All members
 * are private.
 */
```

```
struct DBusServerDebugPipe
{
    DBusServer base; /**< Parent class members. */

    char *name; /**< Server name. */

    dbus_bool_t disconnected; /**< TRUE if disconnect has been called */
};
```

```
/* FIXME not threadsafe (right now the test suite doesn't use threads
anyhow) */
```

```
static DBusHashTable *server_pipe_hash;
static int server_pipe_hash_refcount = 0;
```

```
static dbus_bool_t
pipe_hash_ref (void)
```

```

{
    if (!server_pipe_hash)
    {
        _dbus_assert (server_pipe_hash_refcount == 0);

        server_pipe_hash = _dbus_hash_table_new (DBUS_HASH_STRING, NULL,
NULL);

        if (!server_pipe_hash)
            return FALSE;
    }

    server_pipe_hash_refcount = 1;

    return TRUE;
}

static void
pipe_hash_unref (void)
{
    _dbus_assert (server_pipe_hash != NULL);
    _dbus_assert (server_pipe_hash_refcount > 0);

    server_pipe_hash_refcount -= 1;
    if (server_pipe_hash_refcount == 0)
    {
        _dbus_hash_table_unref (server_pipe_hash);
        server_pipe_hash = NULL;
    }
}

static void
debug_finalize (DBusServer *server)
{
    DBusServerDebugPipe *debug_server = (DBusServerDebugPipe*) server;

    pipe_hash_unref ();

    _dbus_server_finalize_base (server);

    dbus_free (debug_server->name);
    dbus_free (server);
}

static void
debug_disconnect (DBusServer *server)
{
    ((DBusServerDebugPipe*)server)->disconnected = TRUE;
}

static DBusServerVTable debug_vtable = {
    debug_finalize,

```

```

    debug_disconnect
};

/**
 * Creates a new debug server using an in-process pipe
 *
 * @param server_name the name of the server.
 * @param error address where an error can be returned.
 * @returns a new server, or #NULL on failure.
 */
DBusServer*
_dbus_server_debug_pipe_new (const char      *server_name,
                             DBusError      *error)
{
    DBusServerDebugPipe *debug_server;
    DBusString address;
    DBusString name_str;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (!pipe_hash_ref ())
        return NULL;

    if (_dbus_hash_table_lookup_string (server_pipe_hash, server_name)
        != NULL)
    {
        dbus_set_error (error, DBUS_ERROR_ADDRESS_IN_USE, NULL);
        pipe_hash_unref ();
        return NULL;
    }

    debug_server = dbus_new0 (DBusServerDebugPipe, 1);
    if (debug_server == NULL)
        goto nomem_0;

    if (!_dbus_string_init (&address))
        goto nomem_1;

    _dbus_string_init_const (&name_str, server_name);
    if (!_dbus_string_append (&address, "debug-pipe:name=") ||
        !_dbus_address_append_escaped (&address, &name_str))
        goto nomem_2;

    debug_server->name = _dbus_strdup (server_name);
    if (debug_server->name == NULL)
        goto nomem_2;

    if (!_dbus_server_init_base (&debug_server->base,
                                &debug_vtable, &address))
        goto nomem_3;

    if (!_dbus_hash_table_insert_string (server_pipe_hash,

```

```

        debug_server->name,
        debug_server))

    goto nomem_4;

    _dbus_string_free (&address);

    /* server keeps the pipe hash ref */

    _dbus_server_trace_ref (&debug_server->base, 0, 1,
"debug_pipe_new");
    return (DBusServer *)debug_server;

nomem_4:
    _dbus_server_finalize_base (&debug_server->base);
nomem_3:
    dbus_free (debug_server->name);
nomem_2:
    _dbus_string_free (&address);
nomem_1:
    dbus_free (debug_server);
nomem_0:
    pipe_hash_unref ();
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    return NULL;
}

/**
 * Creates the client-side transport for
 * a debug-pipe connection connected to the
 * given debug-pipe server name.
 *
 * @param server_name name of server to connect to
 * @param error address where an error can be returned.
 * @returns #NULL on no memory or transport
 */
DBusTransport*
_dbus_transport_debug_pipe_new (const char          *server_name,
                               DBusError          *error)
{
    DBusTransport *client_transport;
    DBusTransport *server_transport;
    DBusConnection *connection;
    int client_fd, server_fd;
    DBusServer *server;
    DBusString address;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (server_pipe_hash == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_NO_SERVER, NULL);
        return NULL;
    }

```

```

    }

server = _dbus_hash_table_lookup_string (server_pipe_hash,
                                       server_name);

if (server == NULL ||
    ((DBusServerDebugPipe*)server)->disconnected)
{
    dbus_set_error (error, DBUS_ERROR_NO_SERVER, NULL);
    return NULL;
}

if (!_dbus_string_init (&address))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    return NULL;
}

if (!_dbus_string_append (&address, "debug-pipe:name=") ||
    !_dbus_string_append (&address, server_name))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&address);
    return NULL;
}

if (!_dbus_full_duplex_pipe (&client_fd, &server_fd, FALSE,
                           NULL))
{
    _dbus_verbose ("failed to create full duplex pipe\n");
    dbus_set_error (error, DBUS_ERROR_FAILED, "Could not create
full-duplex pipe");
    _dbus_string_free (&address);
    return NULL;
}

client_transport = _dbus_transport_new_for_socket (client_fd,
                                                  NULL, &address);

if (client_transport == NULL)
{
    _dbus_close_socket (client_fd, NULL);
    _dbus_close_socket (server_fd, NULL);
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    _dbus_string_free (&address);
    return NULL;
}

_dbus_string_free (&address);

client_fd = -1;

server_transport = _dbus_transport_new_for_socket (server_fd,

```

```

                                                                    &server-
>guid_hex, NULL);
    if (server_transport == NULL)
    {
        _dbus_transport_unref (client_transport);
        _dbus_close_socket (server_fd, NULL);
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return NULL;
    }

    server_fd = -1;

    if (!_dbus_transport_set_auth_mechanisms (server_transport,
                                                                    (const char**) server-
>auth_mechanisms))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        _dbus_transport_unref (server_transport);
        _dbus_transport_unref (client_transport);
        return NULL;
    }

    connection = _dbus_connection_new_for_transport (server_transport);
    _dbus_transport_unref (server_transport);
    server_transport = NULL;

    if (connection == NULL)
    {
        _dbus_transport_unref (client_transport);
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return NULL;
    }

    /* See if someone wants to handle this new connection,
     * self-referencing for paranoia
     */
    if (server->new_connection_function)
    {
        dbus_server_ref (server);
        (* server->new_connection_function) (server, connection,
                                                                    server-
>new_connection_data);
        dbus_server_unref (server);
    }

    /* If no one grabbed a reference, the connection will die,
     * and the client transport will get an immediate disconnect
     */
    _dbus_connection_close_if_only_one_ref (connection);
    dbus_connection_unref (connection);

    return client_transport;

```



```

}

/**
 * Tries to interpret the address entry as a debug pipe entry.
 *
 * Sets error if the result is not OK.
 *
 * @param entry an address entry
 * @param server_p location to store a new DbusServer, or #NULL on
failure.
 * @param error location to store rationale for failure on bad address
 * @returns the outcome
 */
DBusServerListenResult
_dbus_server_listen_debug_pipe (DBusAddressEntry *entry,
                                DbusServer      **server_p,
                                DbusError        *error)
{
    const char *method;

    *server_p = NULL;

    method = dbus_address_entry_get_method (entry);

    if (strcmp (method, "debug-pipe") == 0)
    {
        const char *name = dbus_address_entry_get_value (entry, "name");

        if (name == NULL)
        {
            _dbus_set_bad_address(error, "debug-pipe", "name",
                                NULL);
            return DBUS_SERVER_LISTEN_BAD_ADDRESS;
        }

        *server_p = _dbus_server_debug_pipe_new (name, error);

        if (*server_p)
        {
            _DBUS_ASSERT_ERROR_IS_CLEAR(error);
            return DBUS_SERVER_LISTEN_OK;
        }
        else
        {
            _DBUS_ASSERT_ERROR_IS_SET(error);
            return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
        }
    }
    else
    {
        _DBUS_ASSERT_ERROR_IS_CLEAR(error);
    }
}

```

```

        return DBUS_SERVER_LISTEN_NOT_HANDLED;
    }
}

/**
 * Opens a debug pipe transport, used in the test suite.
 *
 * @param entry the address entry to try opening as debug-pipe
 * @param transport_p return location for the opened transport
 * @param error error to be set
 * @returns result of the attempt
 */
DBusTransportOpenResult
_dbus_transport_open_debug_pipe (DBusAddressEntry *entry,
                                 DBusTransport **transport_p,
                                 DBusError *error)
{
    const char *method;

    method = dbus_address_entry_get_method (entry);
    _dbus_assert (method != NULL);

    if (strcmp (method, "debug-pipe") == 0)
    {
        const char *name = dbus_address_entry_get_value (entry, "name");

        if (name == NULL)
        {
            _dbus_set_bad_address (error, "debug-pipe", "name",
                                  NULL);
            return DBUS_TRANSPORT_OPEN_BAD_ADDRESS;
        }

        *transport_p = _dbus_transport_debug_pipe_new (name, error);

        if (*transport_p == NULL)
        {
            _DBUS_ASSERT_ERROR_IS_SET (error);
            return DBUS_TRANSPORT_OPEN_DID_NOT_CONNECT;
        }
        else
        {
            _DBUS_ASSERT_ERROR_IS_CLEAR (error);
            return DBUS_TRANSPORT_OPEN_OK;
        }
    }
    else
    {
        _DBUS_ASSERT_ERROR_IS_CLEAR (error);
        return DBUS_TRANSPORT_OPEN_NOT_HANDLED;
    }
}

```

```

/** @} */

#endif /* DBUS_BUILD_TESTS */

File = dbus-server-debug-pipe.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-debug-pipe.h In-proc debug server implementation
 *
 * Copyright (C) 2003 CodeFactory AB
 * Copyright (C) 2003 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */
#ifndef DBUS_SERVER_DEBUG_PIPE_H
#define DBUS_SERVER_DEBUG_PIPE_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-server-protected.h>
#include <dbus/dbus-transport-protected.h>

DBUS_BEGIN_DECLS

DBusServer*          _dbus_server_debug_pipe_new      (const char
*server_name,
                                                         DBusError
*error);
DBusTransport*      _dbus_transport_debug_pipe_new   (const char
*server_name,

```

```

                                                                    DBusError
*error);
DBusServerListenResult _dbus_server_listen_debug_pipe
(DBusAddressEntry *entry,
                                                                    DBusServer
**server_p,
                                                                    DBusError
*error);
DBusTransportOpenResult _dbus_transport_open_debug_pipe
(DBusAddressEntry *entry,
                                                                    DBusTransport
**transport_p,
                                                                    DBusError
*error);

DBUS_END_DECLS

#endif /* DBUS_SERVER_DEBUG_PIPE_H */

File = dbus-server-protected.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-protected.h Used by subclasses of DBusServer object
(internal to D-Bus implementation)
*
* Copyright (C) 2002 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#endif DBUS_SERVER_PROTECTED_H
#define DBUS_SERVER_PROTECTED_H

```

```

#include <dbus/dbus-internals.h>
#include <dbus/dbus-threads-internal.h>
#include <dbus/dbus-server.h>
#include <dbus/dbus-address.h>
#include <dbus/dbus-timeout.h>
#include <dbus/dbus-watch.h>
#include <dbus/dbus-resources.h>
#include <dbus/dbus-dataslot.h>
#include <dbus/dbus-string.h>

DBUS_BEGIN_DECLS

typedef struct DBusServerVTable DBusServerVTable;

/**
 * Virtual table to be implemented by all server "subclasses"
 */
struct DBusServerVTable
{
    void          (* finalize)          (DBusServer *server);
    /**< The finalize method must free the server. */

    void          (* disconnect)       (DBusServer *server);
    /**< Disconnect this server. */
};

/**
 * @ingroup DBusServerInternals
 * Internals of DBusServer object
 */
struct DBusServer
{
    DBusAtomic refcount;           /**< Reference count. */
    const DBusServerVTable *vtable; /**< Virtual methods for
this instance. */
    DBusRMutex *mutex;           /**< Lock on the server
object */

    DBusGUID guid;               /**< Globally unique ID
of server */

    DBusString guid_hex;         /**< Hex-encoded version
of GUID */

    DBusWatchList *watches;      /**< Our watches */
    DBusTimeoutList *timeouts;   /**< Our timeouts */

    char *address;               /**< Address this server
is listening on. */
    dbus_bool_t published_address; /**< flag which
indicates that server has published its bus address. */
};

```

```

    int max_connections;                               /**< Max number of
connections allowed at once. */

    DBusDataSlotList slot_list; /**< Data stored by allocated integer
ID */

    DBusNewConnectionFunction new_connection_function;
/**< Callback to invoke when a new connection is created. */
    void *new_connection_data;
/**< Data for new connection callback */
    DBusFreeFunction new_connection_free_data_function;
/**< Callback to invoke to free new_connection_data
* when server is finalized or data is replaced.
*/

    char **auth_mechanisms; /**< Array of allowed authentication
mechanisms */

    unsigned int disconnected : 1;                      /**< TRUE if we are
disconnected. */

#ifdef DBUS_DISABLE_CHECKS
    unsigned int have_server_lock : 1; /**< Does someone have the server
mutex locked */
#endif
};

dbus_bool_t _dbus_server_init_base (DBusServer
*server,
                                   const DBusServerVTable
*vtable,
                                   const DBusString
*address);
void _dbus_server_finalize_base (DBusServer
*server);
dbus_bool_t _dbus_server_add_watch (DBusServer
*server,
                                   DBusWatch
*watch);
void _dbus_server_remove_watch (DBusServer
*server,
                                DBusWatch
*watch);
void _dbus_server_toggle_watch (DBusServer
*server,
                                DBusWatch
*watch,
                                dbus_bool_t
enabled);
dbus_bool_t _dbus_server_add_timeout (DBusServer
*server,

```

```

DBusTimeout
*timeout);
void _dbus_server_remove_timeout (DBusServer
*server,
DBusTimeout
*timeout);
void _dbus_server_toggle_timeout (DBusServer
*server,
DBusTimeout
*timeout,
dbus_bool_t
enabled);

void _dbus_server_ref_unlocked (DBusServer
*server);
void _dbus_server_unref_unlocked (DBusServer
*server);

typedef enum
{
    DBUS_SERVER_LISTEN_NOT_HANDLED, /**< we aren't in charge of this
address type */
    DBUS_SERVER_LISTEN_OK,          /**< we set up the listen */
    DBUS_SERVER_LISTEN_BAD_ADDRESS, /**< malformed address */
    DBUS_SERVER_LISTEN_DID_NOT_CONNECT, /**< well-formed address but
failed to set it up */
    DBUS_SERVER_LISTEN_ADDRESS_ALREADY_USED /**< address is already used
*/
} DBusServerListenResult;

DBusServerListenResult _dbus_server_listen_platform_specific
(DBusAddressEntry *entry,

DBusServer **server_p,

DBusError *error);

#ifdef DBUS_ENABLE_VERBOSE_MODE
void _dbus_server_trace_ref (DBusServer *server,
    int old_refcount,
    int new_refcount,
    const char *why);
#else
#define _dbus_server_trace_ref(s,o,n,w) \
    do \
    { \
        (void) (o); \
        (void) (n); \
    } while (0)
#endif

#ifdef DBUS_DISABLE_CHECKS

```

```

#define TOOK_LOCK_CHECK(server)
#define RELEASING_LOCK_CHECK(server)
#define HAVE_LOCK_CHECK(server)
#else
#define TOOK_LOCK_CHECK(server) do {           \
    _dbus_assert (!(server)->have_server_lock); \
    (server)->have_server_lock = TRUE;         \
} while (0)
#define RELEASING_LOCK_CHECK(server) do {           \
    _dbus_assert ((server)->have_server_lock);    \
    (server)->have_server_lock = FALSE;          \
} while (0)
#define HAVE_LOCK_CHECK(server) _dbus_assert ((server)-
>have_server_lock)
/* A "DO_NOT_HAVE_LOCK_CHECK" is impossible since we need the lock to
check the flag */
#endif

#define TRACE_LOCKS 0

#define SERVER_LOCK(server) do {
\
    if (TRACE_LOCKS) { _dbus_verbose ("LOCK\n"); } \
    _dbus_rmutex_lock ((server)->mutex);
\
    TOOK_LOCK_CHECK (server);
\
} while (0)

#define SERVER_UNLOCK(server) do {
\
    if (TRACE_LOCKS) { _dbus_verbose ("UNLOCK\n"); } \
    RELEASING_LOCK_CHECK (server);
\
    _dbus_rmutex_unlock ((server)->mutex);
\
} while (0)

DBUS_END_DECLS

#endif /* DBUS_SERVER_PROTECTED_H */

```

File = dbus-server-socket.c

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-socket.c Server implementation for sockets
*
* Copyright (C) 2002, 2003, 2004, 2006 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1

```



```

*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

```

```

#include <config.h>
#include "dbus-internals.h"
#include "dbus-server-socket.h"
#include "dbus-transport-socket.h"
#include "dbus-connection-internal.h"
#include "dbus-memory.h"
#include "dbus-nonce.h"
#include "dbus-string.h"

/**
 * @defgroup DBusServerSocket DBusServer implementations for SOCKET
 * @ingroup DBusInternals
 * @brief Implementation details of DBusServer on SOCKET
 *
 * @{
 */
/**
 * Opaque object representing a Socket server implementation.
 */
typedef struct DBusServerSocket DBusServerSocket;

/**
 * Implementation details of DBusServerSocket. All members
 * are private.
 */
struct DBusServerSocket
{
    DBusServer base;    /**< Parent class members. */
    int n_fds;         /**< Number of active file handles */
    int *fds;          /**< File descriptor or -1 if disconnected. */
    DBusWatch **watch; /**< File descriptor watch. */

```

```

    char *socket_name; /**< Name of domain socket, to unlink if
appropriate */
    DBusNonceFile *noncefile; /**< Nonce file used to authenticate
clients */
};

static void
socket_finalize (DBusServer *server)
{
    DBusServerSocket *socket_server = (DBusServerSocket*) server;
    int i;

    _dbus_server_finalize_base (server);

    for (i = 0 ; i < socket_server->n_fds ; i++)
        if (socket_server->watch[i])
            {
                _dbus_watch_unref (socket_server->watch[i]);
                socket_server->watch[i] = NULL;
            }

    dbus_free (socket_server->fds);
    dbus_free (socket_server->watch);
    dbus_free (socket_server->socket_name);
    if (socket_server->noncefile)
        _dbus_noncefile_delete (socket_server->noncefile, NULL);
    dbus_free (socket_server->noncefile);
    dbus_free (server);
}

/* Return value is just for memory, not other failures. */
static dbus_bool_t
handle_new_client_fd_and_unlock (DBusServer *server,
                                int         client_fd)
{
    DBusConnection *connection;
    DBusTransport *transport;
    DBusNewConnectionFunction new_connection_function;
    void *new_connection_data;

    _dbus_verbose ("Creating new client connection with fd %d\n",
client_fd);

    HAVE_LOCK_CHECK (server);

    if (!_dbus_set_fd_nonblocking (client_fd, NULL))
        {
            SERVER_UNLOCK (server);
            return TRUE;
        }
}

```

```

    transport = _dbus_transport_new_for_socket (client_fd, &server-
>guid_hex, FALSE);
    if (transport == NULL)
    {
        _dbus_close_socket (client_fd, NULL);
        SERVER_UNLOCK (server);
        return FALSE;
    }

    if (!_dbus_transport_set_auth_mechanisms (transport,
>auth_mechanisms)
        (const char **) server-
    {
        _dbus_transport_unref (transport);
        SERVER_UNLOCK (server);
        return FALSE;
    }

    /* note that client_fd is now owned by the transport, and will be
    * closed on transport disconnection/finalization
    */

    connection = _dbus_connection_new_for_transport (transport);
    _dbus_transport_unref (transport);
    transport = NULL; /* now under the connection lock */

    if (connection == NULL)
    {
        SERVER_UNLOCK (server);
        return FALSE;
    }

    /* See if someone wants to handle this new connection, self-
    referencing
    * for paranoia.
    */
    new_connection_function = server->new_connection_function;
    new_connection_data = server->new_connection_data;

    _dbus_server_ref_unlocked (server);
    SERVER_UNLOCK (server);

    if (new_connection_function)
    {
        (* new_connection_function) (server, connection,
        new_connection_data);
    }
    dbus_server_unref (server);

    /* If no one grabbed a reference, the connection will die. */
    _dbus_connection_close_if_only_one_ref (connection);
    dbus_connection_unref (connection);

```

```

    return TRUE;
}

static dbus_bool_t
socket_handle_watch (DBusWatch      *watch,
                    unsigned int    flags,
                    void            *data)
{
    DBusServer *server = data;
    DBusServerSocket *socket_server = data;

#ifdef DBUS_DISABLE_ASSERT
    int i;
    dbus_bool_t found = FALSE;
#endif

    SERVER_LOCK (server);

#ifdef DBUS_DISABLE_ASSERT
    for (i = 0 ; i < socket_server->n_fds ; i++)
    {
        if (socket_server->watch[i] == watch)
            found = TRUE;
    }
    _dbus_assert (found);
#endif

    _dbus_verbose ("Handling client connection, flags 0x%x\n", flags);

    if (flags & DBUS_WATCH_READABLE)
    {
        int client_fd;
        int listen_fd;

        listen_fd = dbus_watch_get_socket (watch);

        if (socket_server->noncefile)
            client_fd = _dbus_accept_with_noncefile (listen_fd,
socket_server->noncefile);
        else
            client_fd = _dbus_accept (listen_fd);

        if (client_fd < 0)
        {
            /* EINTR handled for us */

            if (_dbus_get_is_errno_eagain_or_ewouldblock ())
                _dbus_verbose ("No client available to accept after
all\n");
            else

```

```

        _dbus_verbose ("Failed to accept a client connection:
%s\n",
                        _dbus_strerror_from_errno ());

        SERVER_UNLOCK (server);
    }
    else
    {
        if (!handle_new_client_fd_and_unlock (server, client_fd))
            _dbus_verbose ("Rejected client connection due to lack of
memory\n");
    }
}

if (flags & DBUS_WATCH_ERROR)
    _dbus_verbose ("Error on server listening socket\n");

if (flags & DBUS_WATCH_HANGUP)
    _dbus_verbose ("Hangup on server listening socket\n");

return TRUE;
}

static void
socket_disconnect (DBusServer *server)
{
    DBusServerSocket *socket_server = (DBusServerSocket*) server;
    int i;

    HAVE_LOCK_CHECK (server);

    for (i = 0 ; i < socket_server->n_fds ; i++)
    {
        if (socket_server->watch[i])
        {
            _dbus_server_remove_watch (server,
                                       socket_server->watch[i]);
            _dbus_watch_invalidate (socket_server->watch[i]);
            _dbus_watch_unref (socket_server->watch[i]);
            socket_server->watch[i] = NULL;
        }

        _dbus_close_socket (socket_server->fds[i], NULL);
        socket_server->fds[i] = -1;
    }

    if (socket_server->socket_name != NULL)
    {
        DBusString tmp;
        _dbus_string_init_const (&tmp, socket_server->socket_name);
        _dbus_delete_file (&tmp, NULL);
    }
}

```

```

    if (server->published_address)
        _dbus_daemon_unpublish_session_bus_address();

    HAVE_LOCK_CHECK (server);
}

static const DBusServerVTable socket_vtable = {
    socket_finalize,
    socket_disconnect
};

/**
 * Creates a new server listening on the given file descriptor. The
 * file descriptor should be nonblocking (use
 * _dbus_set_fd_nonblocking() to make it so). The file descriptor
 * should be listening for connections, that is, listen() should have
 * been successfully invoked on it. The server will use accept() to
 * accept new client connections.
 *
 * @param fds list of file descriptors.
 * @param n_fds number of file descriptors
 * @param address the server's address
 * @param noncefile to be used for authentication (NULL if not needed)
 * @returns the new server, or #NULL if no memory.
 */
DBusServer*
_dbus_server_new_for_socket (int          *fds,
                             int          n_fds,
                             const DBusString *address,
                             DBusNonceFile *noncefile)
{
    DBusServerSocket *socket_server;
    DBusServer *server;
    int i;

    socket_server = dbus_new0 (DBusServerSocket, 1);
    if (socket_server == NULL)
        return NULL;

    socket_server->noncefile = noncefile;

    socket_server->fds = dbus_new (int, n_fds);
    if (!socket_server->fds)
        goto failed_0;

    socket_server->watch = dbus_new0 (DBusWatch *, n_fds);
    if (!socket_server->watch)
        goto failed_1;

    for (i = 0 ; i < n_fds ; i++)

```

```

{
    DBusWatch *watch;

    watch = _dbus_watch_new (fds[i],
                            DBUS_WATCH_READABLE,
                            TRUE,
                            socket_handle_watch, socket_server,
                            NULL);

    if (watch == NULL)
        goto failed_2;

    socket_server->n_fds++;
    socket_server->fds[i] = fds[i];
    socket_server->watch[i] = watch;
}

if (!_dbus_server_init_base (&socket_server->base,
                            &socket_vtable, address))
    goto failed_2;

server = (DBusServer*)socket_server;

SERVER_LOCK (server);

for (i = 0 ; i < n_fds ; i++)
{
    if (!_dbus_server_add_watch (&socket_server->base,
                                socket_server->watch[i]))
    {
        int j;
        for (j = 0 ; j < i ; j++)
            _dbus_server_remove_watch (server,
                                       socket_server->watch[j]);

        SERVER_UNLOCK (server);
        _dbus_server_finalize_base (&socket_server->base);
        goto failed_2;
    }
}

SERVER_UNLOCK (server);

_dbus_server_trace_ref (&socket_server->base, 0, 1,
"new_for_socket");
return (DBusServer*) socket_server;

failed_2:
for (i = 0 ; i < n_fds ; i++)
{
    if (socket_server->watch[i] != NULL)
    {
        _dbus_watch_unref (socket_server->watch[i]);
    }
}

```

```

        socket_server->watch[i] = NULL;
    }
}
dbus_free (socket_server->watch);

failed_1:
    dbus_free (socket_server->fds);

failed_0:
    dbus_free (socket_server);
    return NULL;
}

/**
 * Creates a new server listening on TCP.
 * If host is NULL, it will default to localhost.
 * If bind is NULL, it will default to the value for the host
 * parameter, and if that is NULL, then localhost
 * If bind is a hostname, it will be resolved and will listen
 * on all returned addresses.
 * If family is NULL, hostname resolution will try all address
 * families, otherwise it can be ipv4 or ipv6 to restrict the
 * addresses considered.
 *
 * @param host the hostname to report for the listen address
 * @param bind the hostname to listen on
 * @param port the port to listen on or 0 to let the OS choose
 * @param family
 * @param error location to store reason for failure.
 * @param use_nonce whether to use a nonce for low-level
authentication (nonce-tcp transport) or not (tcp transport)
 * @returns the new server, or #NULL on failure.
 */
DBusServer*
_dbus_server_new_for_tcp_socket (const char    *host,
                                const char    *bind,
                                const char    *port,
                                const char    *family,
                                DBusError     *error,
                                dbus_bool_t   use_nonce)
{
    DBusServer *server;
    int *listen_fds = NULL;
    int nlisten_fds = 0, i;
    DBusString address;
    DBusString host_str;
    DBusString port_str;
    DBusNonceFile *noncefile;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    noncefile = NULL;

```



```

if (!_dbus_string_init (&address))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    return NULL;
}

if (!_dbus_string_init (&port_str))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto failed_0;
}

if (host == NULL)
    host = "localhost";

if (port == NULL)
    port = "0";

if (bind == NULL)
    bind = host;
else if (strcmp (bind, "*") == 0)
    bind = NULL;

nlisten_fds = _dbus_listen_tcp_socket (bind, port, family,
                                       &port_str,
                                       &listen_fds, error);

if (nlisten_fds <= 0)
{
    _DBUS_ASSERT_ERROR_IS_SET(error);
    goto failed_1;
}

_dbus_string_init_const (&host_str, host);
if (!_dbus_string_append (&address, use_nonce ? "nonce-tcp:host=" :
"tcp:host=") ||
    !_dbus_address_append_escaped (&address, &host_str) ||
    !_dbus_string_append (&address, ",port=") ||
    !_dbus_string_append (&address,
_dbus_string_get_const_data(&port_str)))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto failed_2;
}

if (family &&
    (!_dbus_string_append (&address, ",family=") ||
     !_dbus_string_append (&address, family)))
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto failed_2;
}

```

```

if (use_nonce)
{
    noncefile = dbus_new0 (DBusNonceFile, 1);
    if (noncefile == NULL)
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto failed_2;
    }

    if (!_dbus_noncefile_create (noncefile, error))
        goto failed_3;

    if (!_dbus_string_append (&address, ",noncefile=") ||
        !_dbus_address_append_escaped (&address,
        _dbus_noncefile_get_path (noncefile)))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto failed_4;
    }

}

server = _dbus_server_new_for_socket (listen_fds, nlisten_fds,
&address, noncefile);
if (server == NULL)
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto failed_4;
}

_dbus_string_free (&port_str);
_dbus_string_free (&address);
dbus_free(listen_fds);

return server;

failed_4:
    _dbus_noncefile_delete (noncefile, NULL);

failed_3:
    dbus_free (noncefile);

failed_2:
    for (i = 0 ; i < nlisten_fds ; i++)
        _dbus_close_socket (listen_fds[i], NULL);
    dbus_free(listen_fds);

failed_1:
    _dbus_string_free (&port_str);

failed_0:
    _dbus_string_free (&address);

```

```

    return NULL;
}

/**
 * Tries to interpret the address entry for various socket-related
 * addresses (well, currently only tcp and nonce-tcp).
 *
 * Sets error if the result is not OK.
 *
 * @param entry an address entry
 * @param server_p a new DBusServer, or #NULL on failure.
 * @param error location to store rationale for failure on bad address
 * @returns the outcome
 */
DBusServerListenResult
_dbus_server_listen_socket (DBusAddressEntry *entry,
                           DBusServer      **server_p,
                           DBusError       *error)
{
    const char *method;

    *server_p = NULL;

    method = dbus_address_entry_get_method (entry);

    if (strcmp (method, "tcp") == 0 || strcmp (method, "nonce-tcp") ==
0)
    {
        const char *host;
        const char *port;
        const char *bind;
        const char *family;

        host = dbus_address_entry_get_value (entry, "host");
        bind = dbus_address_entry_get_value (entry, "bind");
        port = dbus_address_entry_get_value (entry, "port");
        family = dbus_address_entry_get_value (entry, "family");

        *server_p = _dbus_server_new_for_tcp_socket (host, bind, port,
family, error,
strcmp (method, "nonce-tcp") == 0 ? TRUE : FALSE);

        if (*server_p)
        {
            _DBUS_ASSERT_ERROR_IS_CLEAR(error);
            return DBUS_SERVER_LISTEN_OK;
        }
        else
        {
            _DBUS_ASSERT_ERROR_IS_SET(error);

```

```

        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR(error);
    return DBUS_SERVER_LISTEN_NOT_HANDLED;
}
}

/**
 * This is a bad hack since it's really unix domain socket
 * specific. Also, the function weirdly adopts ownership
 * of the passed-in string.
 *
 * @param server a socket server
 * @param filename socket filename to report/delete
 */
void
_dbus_server_socket_own_filename (DBusServer *server,
                                  char *filename)
{
    DBusServerSocket *socket_server = (DBusServerSocket*) server;

    socket_server->socket_name = filename;
}

/** @} */

```

File = dbus-server-socket.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-socket.h Server implementation for sockets
 *
 * Copyright (C) 2002, 2006 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of

```

```

* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_SERVER_SOCKET_H
#define DBUS_SERVER_SOCKET_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-server-protected.h>
#include <dbus/dbus-nonce.h>

DBUS_BEGIN_DECLS

DBusServer* _dbus_server_new_for_socket      (int
*fds,
                                             int
n_fds,
                                             const DBusString
*address,
                                             DBusNonceFile
*noncefile);
DBusServer* _dbus_server_new_for_autolaunch (const DBusString
*address,
                                             DBusError
*error);
DBusServer* _dbus_server_new_for_tcp_socket (const char
*host,
                                             const char
*bind,
                                             const char
*port,
                                             const char
*family,
                                             DBusError
*error,
                                             dbus_bool_t
use_nonce);
DBusServerListenResult _dbus_server_listen_socket (DBusAddressEntry
*entry,
                                                   DBusServer
**server_p,
                                                   DBusError
*error);

void _dbus_server_socket_own_filename (DBusServer *server,
char *filename);

```

```
DBUS_END_DECLS
```

```
#endif /* DBUS_SERVER_SOCKET_H */
```

```
File = dbus-server-unix.c
```

```
/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */  
/* dbus-server-unix.c Server implementation for Unix network  
protocols.
```

```
 *  
 * Copyright (C) 2002, 2003, 2004 Red Hat Inc.  
 *  
 * Licensed under the Academic Free License version 2.1  
 *  
 * This program is free software; you can redistribute it and/or  
modify  
 * it under the terms of the GNU General Public License as published  
by  
 * the Free Software Foundation; either version 2 of the License, or  
 * (at your option) any later version.  
 *  
 * This program is distributed in the hope that it will be useful,  
 * but WITHOUT ANY WARRANTY; without even the implied warranty of  
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
 * GNU General Public License for more details.  
 *  
 * You should have received a copy of the GNU General Public License  
 * along with this program; if not, write to the Free Software  
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA  
 *  
 */
```

```
#include <config.h>  
#include "dbus-internals.h"  
#include "dbus-server-unix.h"  
#include "dbus-server-socket.h"  
#include "dbus-server-launchd.h"  
#include "dbus-transport-unix.h"  
#include "dbus-connection-internal.h"  
#include "dbus-sysdeps-unix.h"  
#include "dbus-string.h"
```

```
/**  
 * @defgroup DBusServerUnix DBusServer implementations for UNIX  
 * @ingroup DBusInternals  
 * @brief Implementation details of DBusServer on UNIX  
 *  
 * @{
```

```

*/

/**
 * Tries to interpret the address entry in a platform-specific
 * way, creating a platform-specific server type if appropriate.
 * Sets error if the result is not OK.
 *
 * @param entry an address entry
 * @param server_p location to store a new DBusServer, or #NULL on
failure.
 * @param error location to store rationale for failure on bad address
 * @returns the outcome
 *
 */
DBusServerListenResult
_dbus_server_listen_platform_specific (DBusAddressEntry *entry,
                                       DBusServer      **server_p,
                                       DBusError        *error)
{
    const char *method;

    *server_p = NULL;

    method = dbus_address_entry_get_method (entry);

    if (strcmp (method, "unix") == 0)
    {
        const char *path = dbus_address_entry_get_value (entry, "path");
        const char *tmpdir = dbus_address_entry_get_value (entry,
"tmpdir");
        const char *abstract = dbus_address_entry_get_value (entry,
"abstract");

        if (path == NULL && tmpdir == NULL && abstract == NULL)
        {
            _dbus_set_bad_address(error, "unix",
                                "path or tmpdir or abstract",
                                NULL);
            return DBUS_SERVER_LISTEN_BAD_ADDRESS;
        }

        if ((path && tmpdir) ||
            (path && abstract) ||
            (tmpdir && abstract))
        {
            _dbus_set_bad_address(error, NULL, NULL,
                                "cannot specify two of \"path\" and
\"tmpdir\" and \"abstract\" at the same time");
            return DBUS_SERVER_LISTEN_BAD_ADDRESS;
        }

        if (tmpdir != NULL)

```

```

{
    DBusString full_path;
    DBusString filename;

    if (!_dbus_string_init (&full_path))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }

    if (!_dbus_string_init (&filename))
    {
        _dbus_string_free (&full_path);
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }

    if (!_dbus_string_append (&filename,
                             "dbus-") ||
        !_dbus_generate_random_ascii (&filename, 10) ||
        !_dbus_string_append (&full_path, tmpdir) ||
        !_dbus_concat_dir_and_file (&full_path, &filename))
    {
        _dbus_string_free (&full_path);
        _dbus_string_free (&filename);
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }

    /* Always use abstract namespace if possible with tmpdir */

    *server_p =
        _dbus_server_new_for_domain_socket
        (_dbus_string_get_const_data (&full_path),
#ifdef HAVE_ABSTRACT_SOCKETS
        TRUE,
#else
        FALSE,
#endif
        error);

    _dbus_string_free (&full_path);
    _dbus_string_free (&filename);
}
else
{
    if (path)
        *server_p = _dbus_server_new_for_domain_socket (path,
FALSE, error);
    else
        *server_p = _dbus_server_new_for_domain_socket (abstract,
TRUE, error);
}

```



```

    }

    if (*server_p != NULL)
    {
        _DBUS_ASSERT_ERROR_IS_CLEAR(error);
        return DBUS_SERVER_LISTEN_OK;
    }
    else
    {
        _DBUS_ASSERT_ERROR_IS_SET(error);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }
}
else if (strcmp (method, "systemd") == 0)
{
    int i, n, *fds;
    DBusString address;

    n = _dbus_listen_systemd_sockets (&fds, error);
    if (n < 0)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }

    if (!_dbus_string_init (&address))
        goto systemd_oom;

    for (i = 0; i < n; i++)
    {
        if (i > 0)
        {
            if (!_dbus_string_append (&address, ";"))
                goto systemd_oom;
        }
        if (!_dbus_append_address_from_socket (fds[i], &address,
error))
            goto systemd_err;
    }

    *server_p = _dbus_server_new_for_socket (fds, n, &address,
NULL);
    if (*server_p == NULL)
        goto systemd_oom;

    dbus_free (fds);

    return DBUS_SERVER_LISTEN_OK;
systemd_oom:
    _DBUS_SET_OOM (error);
systemd_err:
    for (i = 0; i < n; i++)

```

```

        {
            _dbus_close_socket (fds[i], NULL);
        }
        dbus_free (fds);
        _dbus_string_free (&address);

        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }
#ifdef DBUS_ENABLE_LAUNCHD
    else if (strcmp (method, "launchd") == 0)
    {
        const char *launchd_env_var = dbus_address_entry_get_value
(entry, "env");
        if (launchd_env_var == NULL)
        {
            _dbus_set_bad_address (error, "launchd", "env", NULL);
            return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
        }
        *server_p = _dbus_server_new_for_launchd (launchd_env_var,
error);

        if (*server_p != NULL)
        {
            _DBUS_ASSERT_ERROR_IS_CLEAR(error);
            return DBUS_SERVER_LISTEN_OK;
        }
        else
        {
            _DBUS_ASSERT_ERROR_IS_SET(error);
            return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
        }
    }
#endif
    else
    {
        /* If we don't handle the method, we return NULL with the
        * error unset
        */
        _DBUS_ASSERT_ERROR_IS_CLEAR(error);
        return DBUS_SERVER_LISTEN_NOT_HANDLED;
    }
}

/**
 * Creates a new server listening on the given Unix domain socket.
 *
 * @param path the path for the domain socket.
 * @param abstract #TRUE to use abstract socket namespace
 * @param error location to store reason for failure.
 * @returns the new server, or #NULL on failure.
 */
DBusServer*

```

```

_dbus_server_new_for_domain_socket (const char      *path,
                                   dbus_bool_t     abstract,
                                   DBusError       *error)
{
    DBusServer *server;
    int listen_fd;
    DBusString address;
    char *path_copy;
    DBusString path_str;

    _DBUS_ASSERT_ERROR_IS_CLEAR (error);

    if (!_dbus_string_init (&address))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        return NULL;
    }

    _dbus_string_init_const (&path_str, path);
    if ((abstract &&
         !_dbus_string_append (&address, "unix:abstract=")) ||
        (!abstract &&
         !_dbus_string_append (&address, "unix:path=")) ||
        !_dbus_address_append_escaped (&address, &path_str))
    {
        dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
        goto failed_0;
    }

    if (abstract)
    {
        path_copy = NULL;
    }
    else
    {
        path_copy = _dbus_strdup (path);
        if (path_copy == NULL)
        {
            dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
            goto failed_0;
        }
    }

    listen_fd = _dbus_listen_unix_socket (path, abstract, error);

    if (listen_fd < 0)
    {
        _DBUS_ASSERT_ERROR_IS_SET (error);
        goto failed_1;
    }

    server = _dbus_server_new_for_socket (&listen_fd, 1, &address, 0);
}

```

```

if (server == NULL)
{
    dbus_set_error (error, DBUS_ERROR_NO_MEMORY, NULL);
    goto failed_2;
}

if (path_copy != NULL)
    _dbus_server_socket_own_filename(server, path_copy);

_dbus_string_free (&address);

return server;

failed_2:
    _dbus_close_socket (listen_fd, NULL);
failed_1:
    dbus_free (path_copy);
failed_0:
    _dbus_string_free (&address);

return NULL;
}

/** @} */

```

File = dbus-server-unix.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-unix.h Server implementation for Unix network
protocols.
*
* Copyright (C) 2002 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software

```

```

* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_SERVER_UNIX_H
#define DBUS_SERVER_UNIX_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-server-protected.h>

DBUS_BEGIN_DECLS

DBusServer* _dbus_server_new_for_domain_socket (const char
*path,
                                                dbus_bool_t
abstract,
                                                DBusError
*error);

DBUS_END_DECLS

#endif /* DBUS_SERVER_UNIX_H */

File = dbus-server-win.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-win.c Server implementation for WIN network protocols.
*
* Copyright (C) 2002, 2003, 2004 Red Hat Inc.
* Copyright (C) 2007 Ralf Habacker <ralf.habacker@freenet.de>
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*

```

```

*/

#include <config.h>
#include "dbus-internals.h"
#include "dbus-server-win.h"
#include "dbus-server-socket.h"

/**
 * @defgroup DBusServerWin DBusServer implementations for Windows
 * @ingroup DBusInternals
 * @brief Implementation details of DBusServer on Windows
 *
 * @{
 */

/**
 * Tries to interpret the address entry in a platform-specific
 * way, creating a platform-specific server type if appropriate.
 * Sets error if the result is not OK.
 *
 * @param entry an address entry
 * @param server_p location to store a new DBusServer, or #NULL on
failure.
 * @param error location to store rationale for failure on bad address
 * @returns the outcome
 */
DBusServerListenResult
_dbus_server_listen_platform_specific (DBusAddressEntry *entry,
                                       DBusServer      **server_p,
                                       DBusError        *error)
{
    const char *method;

    *server_p = NULL;

    method = dbus_address_entry_get_method (entry);

    if (strcmp (method, "autolaunch") == 0)
    {
        const char *host = "localhost";
        const char *bind = "localhost";
        const char *port = "0";
        const char *family = "ipv4";
        const char *scope = dbus_address_entry_get_value (entry,
"scope");

        if (_dbus_daemon_is_session_bus_address_published (scope))
            return DBUS_SERVER_LISTEN_ADDRESS_ALREADY_USED;

        *server_p = _dbus_server_new_for_tcp_socket (host, bind, port,

```

```

family, error,
FALSE);
    if (*server_p)
    {
        _DBUS_ASSERT_ERROR_IS_CLEAR(error);
        (*server_p)->published_address =
            _dbus_daemon_publish_session_bus_address ((*server_p)-
>address, scope);
        return DBUS_SERVER_LISTEN_OK;
    }
    else
    {
        // make sure no handle is open
        _dbus_daemon_unpublish_session_bus_address ();
        _DBUS_ASSERT_ERROR_IS_SET(error);
        return DBUS_SERVER_LISTEN_DID_NOT_CONNECT;
    }
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR(error);
    return DBUS_SERVER_LISTEN_NOT_HANDLED;
}
}

/** @} */

```

File = dbus-server-win.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server-win.h Server implementation for windows network
protocols.
*
* Copyright (C) 2002 Red Hat Inc.
* Copyright (C) 2007 Ralf Habacker <ralf.habacker@freenet.de>
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

```

```

*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#ifndef DBUS_SERVER_WIN_H
#define DBUS_SERVER_WIN_H

#include <dbus/dbus-internals.h>
#include <dbus/dbus-server-protected.h>

DBUS_BEGIN_DECLS

/* add definitions here */

DBUS_END_DECLS

#endif /* DBUS_SERVER_WIN_H */

File = dbus-server.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server.c DbusServer object
*
* Copyright (C) 2002, 2003, 2004, 2005 Red Hat Inc.
*
* Licensed under the Academic Free License version 2.1
*
* This program is free software; you can redistribute it and/or
modify
* it under the terms of the GNU General Public License as published
by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#include <config.h>

```



```

#include "dbus-server.h"
#include "dbus-server-unix.h"
#include "dbus-server-socket.h"
#include "dbus-string.h"
#ifdef DBUS_BUILD_TESTS
#include "dbus-server-debug-pipe.h"
#endif
#include "dbus-address.h"
#include "dbus-protocol.h"

/**
 * @defgroup DBusServer DBusServer
 * @ingroup DBus
 * @brief Server that listens for new connections.
 *
 * A DBusServer represents a server that other applications
 * can connect to. Each connection from another application
 * is represented by a #DBusConnection.
 *
 * @todo Thread safety hasn't been tested much for #DBusServer
 * @todo Need notification to apps of disconnection, may matter for
some transports
 */

/**
 * @defgroup DBusServerInternals DBusServer implementation details
 * @ingroup DBusInternals
 * @brief Implementation details of DBusServer
 *
 * @{
 */

#ifdef _dbus_server_trace_ref
void
_dbus_server_trace_ref (DBusServer *server,
                        int old_refcount,
                        int new_refcount,
                        const char *why)
{
    static int enabled = -1;

    _dbus_trace_ref ("DBusServer", server, old_refcount, new_refcount,
why,
                    "DBUS_SERVER_TRACE", &enabled);
}
#endif

/* this is a little fragile since it assumes the address doesn't
 * already have a guid, but it shouldn't
 */
static char*
copy_address_with_guid_appended (const DBusString *address,

```

```

                                const DBusString *guid_hex)
{
    DBusString with_guid;
    char *retval;

    if (!_dbus_string_init (&with_guid))
        return NULL;

    if (!_dbus_string_copy (address, 0, &with_guid,
                            _dbus_string_get_length (&with_guid)) ||
        !_dbus_string_append (&with_guid, "guid=") ||
        !_dbus_string_copy (guid_hex, 0,
                            &with_guid, _dbus_string_get_length
(&with_guid)))
    {
        _dbus_string_free (&with_guid);
        return NULL;
    }

    retval = NULL;
    _dbus_string_steal_data (&with_guid, &retval);

    _dbus_string_free (&with_guid);

    return retval; /* may be NULL if steal_data failed */
}

/**
 * Initializes the members of the DBusServer base class.
 * Chained up to by subclass constructors.
 *
 * @param server the server.
 * @param vtable the vtable for the subclass.
 * @param address the server's address
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_server_init_base (DBusServer          *server,
                        const DBusServerVTable *vtable,
                        const DBusString     *address)
{
    server->vtable = vtable;

#ifdef DBUS_DISABLE_ASSERT
    _dbus_atomic_inc (&server->refcount);
#else
    {
        dbus_int32_t old_refcount = _dbus_atomic_inc (&server-
>refcount);

        _dbus_assert (old_refcount == 0);
    }
}

```

```

#endif

server->address = NULL;
server->watches = NULL;
server->timeouts = NULL;
server->published_address = FALSE;

if (!_dbus_string_init (&server->guid_hex))
    return FALSE;

_dbus_generate_uuid (&server->guid);

if (!_dbus_uuid_encode (&server->guid, &server->guid_hex))
    goto failed;

server->address = copy_address_with_guid_appended (address,
>server-
>guid_hex);
if (server->address == NULL)
    goto failed;

_dbus_rmutex_new_at_location (&server->mutex);
if (server->mutex == NULL)
    goto failed;

server->watches = _dbus_watch_list_new ();
if (server->watches == NULL)
    goto failed;

server->timeouts = _dbus_timeout_list_new ();
if (server->timeouts == NULL)
    goto failed;

_dbus_data_slot_list_init (&server->slot_list);

_dbus_verbose ("Initialized server on address %s\n", server-
>address);

return TRUE;

failed:
_dbus_rmutex_free_at_location (&server->mutex);
server->mutex = NULL;
if (server->watches)
    {
        _dbus_watch_list_free (server->watches);
        server->watches = NULL;
    }
if (server->timeouts)
    {
        _dbus_timeout_list_free (server->timeouts);
        server->timeouts = NULL;
    }

```

```

    }
    if (server->address)
    {
        dbus_free (server->address);
        server->address = NULL;
    }
    _dbus_string_free (&server->guid_hex);

    return FALSE;
}

/**
 * Finalizes the members of the DbusServer base class.
 * Chained up to by subclass finalizers.
 *
 * @param server the server.
 */
void
_dbus_server_finalize_base (DBusServer *server)
{
    /* We don't have the lock, but nobody should be accessing
     * concurrently since they don't have a ref
     */
#ifdef DBUS_DISABLE_CHECKS
    _dbus_assert (!server->have_server_lock);
#endif
    _dbus_assert (server->disconnected);

    /* calls out to application code... */
    _dbus_data_slot_list_free (&server->slot_list);

    dbus_server_set_new_connection_function (server, NULL, NULL, NULL);

    _dbus_watch_list_free (server->watches);
    _dbus_timeout_list_free (server->timeouts);

    _dbus_rmutex_free_at_location (&server->mutex);

    dbus_free (server->address);

    dbus_free_string_array (server->auth_mechanisms);

    _dbus_string_free (&server->guid_hex);
}

/** Function to be called in protected_change_watch() with refcount
held */
typedef dbus_bool_t (* DbusWatchAddFunction) (DBusWatchList *list,
DBusWatch
*watch);

```

```

/** Function to be called in protected_change_watch() with refcount
held */
typedef void          (* DbusWatchRemoveFunction) (DBusWatchList *list,
                                                  DbusWatch
*watch);
/** Function to be called in protected_change_watch() with refcount
held */
typedef void          (* DbusWatchToggleFunction) (DBusWatchList *list,
                                                  DbusWatch
*watch,
                                                  dbus_bool_t
enabled);

static dbus_bool_t
protected_change_watch (DBusServer          *server,
                      DbusWatch           *watch,
                      DbusWatchAddFunction  add_function,
                      DbusWatchRemoveFunction remove_function,
                      DbusWatchToggleFunction toggle_function,
                      dbus_bool_t          enabled)
{
    DbusWatchList *watches;
    dbus_bool_t retval;

    HAVE_LOCK_CHECK (server);

    /* This isn't really safe or reasonable; a better pattern is the "do
    * everything, then drop lock and call out" one; but it has to be
    * propagated up through all callers
    */

    watches = server->watches;
    if (watches)
    {
        server->watches = NULL;
        _dbus_server_ref_unlocked (server);
        SERVER_UNLOCK (server);

        if (add_function)
            retval = (* add_function) (watches, watch);
        else if (remove_function)
        {
            retval = TRUE;
            (* remove_function) (watches, watch);
        }
        else
        {
            retval = TRUE;
            (* toggle_function) (watches, watch, enabled);
        }

        SERVER_LOCK (server);

```

```

        server->watches = watches;
        _dbus_server_unref_unlocked (server);

        return retval;
    }
    else
        return FALSE;
}

/**
 * Adds a watch for this server, chaining out to application-provided
 * watch handlers.
 *
 * @param server the server.
 * @param watch the watch to add.
 */
dbus_bool_t
_dbus_server_add_watch (DBusServer *server,
                       DBusWatch *watch)
{
    HAVE_LOCK_CHECK (server);
    return protected_change_watch (server, watch,
                                   _dbus_watch_list_add_watch,
                                   NULL, NULL, FALSE);
}

/**
 * Removes a watch previously added with _dbus_server_remove_watch().
 *
 * @param server the server.
 * @param watch the watch to remove.
 */
void
_dbus_server_remove_watch (DBusServer *server,
                           DBusWatch *watch)
{
    HAVE_LOCK_CHECK (server);
    protected_change_watch (server, watch,
                            NULL,
                            _dbus_watch_list_remove_watch,
                            NULL, FALSE);
}

/**
 * Toggles a watch and notifies app via server's
 * DBusWatchToggledFunction if available. It's an error to call this
 * function on a watch that was not previously added.
 *
 * @param server the server.
 * @param watch the watch to toggle.
 * @param enabled whether to enable or disable
 */

```

```

void
_dbus_server_toggle_watch (DBusServer *server,
                           DBusWatch *watch,
                           dbus_bool_t enabled)
{
    _dbus_assert (watch != NULL);

    HAVE_LOCK_CHECK (server);
    protected_change_watch (server, watch,
                            NULL, NULL,
                            _dbus_watch_list_toggle_watch,
                            enabled);
}

/** Function to be called in protected_change_timeout() with refcount
held */
typedef dbus_bool_t (* DBusTimeoutAddFunction) (DBusTimeoutList
*list,
                                                DBusTimeout
*timeout);
/** Function to be called in protected_change_timeout() with refcount
held */
typedef void (* DBusTimeoutRemoveFunction) (DBusTimeoutList
*list,
                                            DBusTimeout
*timeout);
/** Function to be called in protected_change_timeout() with refcount
held */
typedef void (* DBusTimeoutToggleFunction) (DBusTimeoutList
*list,
                                            DBusTimeout
*timeout,
                                            dbus_bool_t
enabled);

static dbus_bool_t
protected_change_timeout (DBusServer *server,
                         DBusTimeout *timeout,
                         DBusTimeoutAddFunction add_function,
                         DBusTimeoutRemoveFunction remove_function,
                         DBusTimeoutToggleFunction toggle_function,
                         dbus_bool_t enabled)
{
    DBusTimeoutList *timeouts;
    dbus_bool_t retval;

    HAVE_LOCK_CHECK (server);

    /* This isn't really safe or reasonable; a better pattern is the "do
everything, then

```



```

/**
 * Removes a timeout previously added with _dbus_server_add_timeout().
 *
 * @param server the server.
 * @param timeout the timeout to remove.
 */
void
_dbus_server_remove_timeout (DBusServer *server,
                             DBusTimeout *timeout)
{
    protected_change_timeout (server, timeout,
                              NULL,
                              _dbus_timeout_list_remove_timeout,
                              NULL, FALSE);
}

/**
 * Toggles a timeout and notifies app via server's
 * DBusTimeoutToggledFunction if available. It's an error to call this
 * function on a timeout that was not previously added.
 *
 * @param server the server.
 * @param timeout the timeout to toggle.
 * @param enabled whether to enable or disable
 */
void
_dbus_server_toggle_timeout (DBusServer *server,
                             DBusTimeout *timeout,
                             dbus_bool_t enabled)
{
    protected_change_timeout (server, timeout,
                              NULL, NULL,
                              _dbus_timeout_list_toggle_timeout,
                              enabled);
}

/**
 * Like dbus_server_ref() but does not acquire the lock (must already
 * be held)
 *
 * @param server the server.
 */
void
_dbus_server_ref_unlocked (DBusServer *server)
{
    dbus_int32_t old_refcount;

    _dbus_assert (server != NULL);
    HAVE_LOCK_CHECK (server);

    old_refcount = _dbus_atomic_inc (&server->refcount);

```

```

    _dbus_assert (old_refcount > 0);
    _dbus_server_trace_ref (server, old_refcount, old_refcount + 1,
        "ref_unlocked");
}

/**
 * Like dbus_server_unref() but does not acquire the lock (must
 * already be held)
 *
 * @param server the server.
 */
void
dbus_server_unref_unlocked (DBusServer *server)
{
    dbus_int32_t old_refcount;

    /* Keep this in sync with dbus_server_unref */

    _dbus_assert (server != NULL);

    HAVE_LOCK_CHECK (server);

    old_refcount = _dbus_atomic_dec (&server->refcount);
    _dbus_assert (old_refcount > 0);

    _dbus_server_trace_ref (server, old_refcount, old_refcount - 1,
        "unref_unlocked");

    if (old_refcount == 1)
    {
        _dbus_assert (server->disconnected);

        SERVER_UNLOCK (server);

        _dbus_assert (server->vtable->finalize != NULL);

        (* server->vtable->finalize) (server);
    }
}

/** @} */

/**
 * @addtogroup DBusServer
 *
 * @{
 */

/**
 * @typedef DBusServer
 */

```

```

* An opaque object representing a server that listens for
* connections from other applications. Each time a connection
* is made, a new DBusConnection is created and made available
* via an application-provided DBusNewConnectionFunction.
* The DBusNewConnectionFunction is provided with
* dbus_server_set_new_connection_function().
*
*/

static const struct {
    DBusServerListenResult (* func) (DBusAddressEntry *entry,
                                     DBusServer      **server_p,
                                     DBusError       *error);
} listen_funcs[] = {
    { _dbus_server_listen_socket }
    , { _dbus_server_listen_platform_specific }
#ifdef DBUS_BUILD_TESTS
    , { _dbus_server_listen_debug_pipe }
#endif
};

/**
* Listens for new connections on the given address. If there are
* multiple semicolon-separated address entries in the address, tries
* each one and listens on the first one that works.
*
* Returns #NULL and sets error if listening fails for any reason.
* Otherwise returns a new #DBusServer.
* dbus_server_set_new_connection_function(),
* dbus_server_set_watch_functions(), and
* dbus_server_set_timeout_functions() should be called immediately to
* render the server fully functional.
*
* To free the server, applications must call first
* dbus_server_disconnect() and then dbus_server_unref().
*
* @param address the address of this server.
* @param error location to store reason for failure.
* @returns a new #DBusServer, or #NULL on failure.
*
*/
DBusServer*
dbus_server_listen (const char      *address,
                   DBusError       *error)
{
    DBusServer *server;
    DBusAddressEntry **entries;
    int len, i;
    DBusError first_connect_error = DBUS_ERROR_INIT;
    dbus_bool_t handled_once;

    _dbus_return_val_if_fail (address != NULL, NULL);

```

```

_dbus_return_val_if_error_is_set (error, NULL);

if (!dbus_parse_address (address, &entries, &len, error))
    return NULL;

server = NULL;
handled_once = FALSE;

for (i = 0; i < len; i++)
{
    int j;

    for (j = 0; j < (int) _DBUS_N_ELEMENTS (listen_funcs); ++j)
    {
        DBusServerListenResult result;
        DBusError tmp_error = DBUS_ERROR_INIT;

        result = (* listen_funcs[j].func) (entries[i],
                                           &server,
                                           &tmp_error);

        if (result == DBUS_SERVER_LISTEN_OK)
        {
            _dbus_assert (server != NULL);
            _DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);
            handled_once = TRUE;
            goto out;
        }
        else if (result == DBUS_SERVER_LISTEN_ADDRESS_ALREADY_USED)
        {
            _dbus_assert (server == NULL);
            dbus_set_error (error,
                          DBUS_ERROR_ADDRESS_IN_USE,
                          "Address '%s' already used",
                          dbus_address_entry_get_method (entries[0]));
            handled_once = TRUE;
            goto out;
        }
        else if (result == DBUS_SERVER_LISTEN_BAD_ADDRESS)
        {
            _dbus_assert (server == NULL);
            _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);
            dbus_move_error (&tmp_error, error);
            handled_once = TRUE;
            goto out;
        }
        else if (result == DBUS_SERVER_LISTEN_NOT_HANDLED)
        {
            _dbus_assert (server == NULL);
            _DBUS_ASSERT_ERROR_IS_CLEAR (&tmp_error);

            /* keep trying addresses */
        }
    }
}

```

```

    }
    else if (result == DBUS_SERVER_LISTEN_DID_NOT_CONNECT)
    {
        _dbus_assert (server == NULL);
        _DBUS_ASSERT_ERROR_IS_SET (&tmp_error);
        if (!dbus_error_is_set (&first_connect_error))
            dbus_move_error (&tmp_error, &first_connect_error);
        else
            dbus_error_free (&tmp_error);

        handled_once = TRUE;

        /* keep trying addresses */
    }
}

_dbus_assert (server == NULL);
_DBUS_ASSERT_ERROR_IS_CLEAR (error);
}

out:

if (!handled_once)
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    if (len > 0)
        dbus_set_error (error,
                        DBUS_ERROR_BAD_ADDRESS,
                        "Unknown address type '%s'",
                        dbus_address_entry_get_method (entries[0]));
    else
        dbus_set_error (error,
                        DBUS_ERROR_BAD_ADDRESS,
                        "Empty address '%s'",
                        address);
}

dbus_address_entries_free (entries);

if (server == NULL)
{
    _dbus_assert (error == NULL || dbus_error_is_set
(&first_connect_error) ||
                dbus_error_is_set (error));

    if (error && dbus_error_is_set (error))
    {
        /* already set the error */
    }
    else
    {
        /* didn't set the error but either error should be

```

```

        * NULL or first_connect_error should be set.
        */
        _dbus_assert (error == NULL || dbus_error_is_set
(&first_connect_error));
        dbus_move_error (&first_connect_error, error);
    }

    _DBUS_ASSERT_ERROR_IS_CLEAR (&first_connect_error); /* be sure
we freed it */
    _DBUS_ASSERT_ERROR_IS_SET (error);

    return NULL;
}
else
{
    _DBUS_ASSERT_ERROR_IS_CLEAR (error);
    return server;
}
}

/**
 * Increments the reference count of a DbusServer.
 *
 * @param server the server.
 * @returns the server
 */
DBusServer *
dbus_server_ref (DBusServer *server)
{
    dbus_int32_t old_refcount;

    _dbus_return_val_if_fail (server != NULL, NULL);

    /* can't get the refcount without a side-effect */
    old_refcount = _dbus_atomic_inc (&server->refcount);

#ifdef DBUS_DISABLE_CHECKS
    if (_DBUS_UNLIKELY (old_refcount <= 0))
    {
        /* undo side-effect first */
        _dbus_atomic_dec (&server->refcount);
        _dbus_warn_check_failed (_dbus_return_if_fail_warning_format,
                                _DBUS_FUNCTION_NAME, "old_refcount >
0",
                                __FILE__, __LINE__);
        return NULL;
    }
#endif

    _dbus_server_trace_ref (server, old_refcount, old_refcount + 1,
"ref");
}

```

```

    return server;
}

/**
 * Decrements the reference count of a DbusServer. Finalizes the
 * server if the reference count reaches zero.
 *
 * The server must be disconnected before the refcount reaches zero.
 *
 * @param server the server.
 */
void
dbus_server_unref (DBusServer *server)
{
    dbus_int32_t old_refcount;

    /* keep this in sync with unref_unlocked */

    _dbus_return_if_fail (server != NULL);

    /* can't get the refcount without a side-effect */
    old_refcount = _dbus_atomic_dec (&server->refcount);

#ifdef DBUS_DISABLE_CHECKS
    if (_DBUS_UNLIKELY (old_refcount <= 0))
    {
        /* undo side-effect first */
        _dbus_atomic_inc (&server->refcount);
        _dbus_warn_check_failed (_dbus_return_if_fail_warning_format,
                                _DBUS_FUNCTION_NAME, "old_refcount >
0",
                                __FILE__, __LINE__);
        return;
    }
#endif

    _dbus_server_trace_ref (server, old_refcount, old_refcount - 1,
"unref");

    if (old_refcount == 1)
    {
        /* lock not held! */
        _dbus_assert (server->disconnected);

        _dbus_assert (server->vtable->finalize != NULL);

        (* server->vtable->finalize) (server);
    }
}

/**
 * Releases the server's address and stops listening for

```

```

* new clients. If called more than once, only the first
* call has an effect. Does not modify the server's
* reference count.
*
* @param server the server.
*/
void
dbus_server_disconnect (DBusServer *server)
{
    _dbus_return_if_fail (server != NULL);

#ifdef DBUS_DISABLE_CHECKS
    _dbus_atomic_inc (&server->refcount);
#else
    {
        dbus_int32_t old_refcount = _dbus_atomic_inc (&server-
>refcount);

        _dbus_return_if_fail (old_refcount > 0);
    }
#endif

    SERVER_LOCK (server);

    _dbus_assert (server->vtable->disconnect != NULL);

    if (!server->disconnected)
    {
        /* this has to be first so recursive calls to disconnect don't
happen */
        server->disconnected = TRUE;

        (* server->vtable->disconnect) (server);
    }

    SERVER_UNLOCK (server);
    dbus_server_unref (server);
}

/**
* Returns #TRUE if the server is still listening for new connections.
*
* @param server the server.
*/
dbus_bool_t
dbus_server_get_is_connected (DBusServer *server)
{
    dbus_bool_t retval;

    _dbus_return_val_if_fail (server != NULL, FALSE);

    SERVER_LOCK (server);

```



```

    retval = !server->disconnected;
    SERVER_UNLOCK (server);

    return retval;
}

/**
 * Returns the address of the server, as a newly-allocated
 * string which must be freed by the caller.
 *
 * @param server the server
 * @returns the address or #NULL if no memory
 */
char*
dbus_server_get_address (DBusServer *server)
{
    char *retval;

    _dbus_return_val_if_fail (server != NULL, NULL);

    SERVER_LOCK (server);
    retval = _dbus_strdup (server->address);
    SERVER_UNLOCK (server);

    return retval;
}

/**
 * Returns the unique ID of the server, as a newly-allocated
 * string which must be freed by the caller. This ID is
 * normally used by clients to tell when two #DBusConnection
 * would be equivalent (because the server address passed
 * to dbus_connection_open() will have the same guid in the
 * two cases). dbus_connection_open() can re-use an existing
 * connection with the same ID instead of opening a new
 * connection.
 *
 * This is an ID unique to each #DBusServer. Remember that
 * a #DBusServer represents only one mode of connecting,
 * so e.g. a bus daemon can listen on multiple addresses
 * which will mean it has multiple #DBusServer each with
 * their own ID.
 *
 * The ID is not a UUID in the sense of RFC4122; the details
 * are explained in the D-Bus specification.
 *
 * @param server the server
 * @returns the id of the server or #NULL if no memory
 */
char*
dbus_server_get_id (DBusServer *server)
{

```

```

char *retval;

_dbus_return_val_if_fail (server != NULL, NULL);

SERVER_LOCK (server);
retval = NULL;
_dbus_string_copy_data (&server->guid_hex, &retval);
SERVER_UNLOCK (server);

return retval;
}

/**
 * Sets a function to be used for handling new connections. The given
 * function is passed each new connection as the connection is
 * created. If the new connection function increments the connection's
 * reference count, the connection will stay alive. Otherwise, the
 * connection will be unreferenced and closed. The new connection
 * function may also close the connection itself, which is considered
 * good form if the connection is not wanted.
 *
 * The connection here is private in the sense of
 * dbus_connection_open_private(), so if the new connection function
 * keeps a reference it must arrange for the connection to be closed.
 * i.e. libdbus does not own this connection once the new connection
 * function takes a reference.
 *
 * @param server the server.
 * @param function a function to handle new connections.
 * @param data data to pass to the new connection handler.
 * @param free_data_function function to free the data.
 */
void
dbus_server_set_new_connection_function (DBusServer
*server,
                                     DBusNewConnectionFunction
function,
                                     void
*data,
                                     DBusFreeFunction
free_data_function)
{
    DBusFreeFunction old_free_function;
    void *old_data;

    _dbus_return_if_fail (server != NULL);

    SERVER_LOCK (server);
    old_free_function = server->new_connection_free_data_function;
    old_data = server->new_connection_data;

    server->new_connection_function = function;

```



```

        toggled_function,
        data,
        free_data_function);

    SERVER_LOCK (server);
}
else
{
    _dbus_warn_check_failed ("Re-entrant call to %s\n",
_DBUS_FUNCTION_NAME);
    result = FALSE;
}
server->watches = watches;
SERVER_UNLOCK (server);

return result;
}

/**
 * Sets the timeout functions for the server. These functions are
 * responsible for making the application's main loop aware of
 * timeouts.
 *
 * This function behaves exactly like
 * dbus_connection_set_timeout_functions();
 * see the documentation for that routine.
 *
 * @param server the server.
 * @param add_function function to add a timeout.
 * @param remove_function function to remove a timeout.
 * @param toggled_function function to notify when the timeout is
 * enabled/disabled
 * @param data data to pass to add_function and remove_function.
 * @param free_data_function function to be called to free the data.
 * @returns #FALSE on failure (no memory)
 */
dbus_bool_t
dbus_server_set_timeout_functions (DBusServer *server,
    DBusAddTimeoutFunction add_function,
    DBusRemoveTimeoutFunction remove_function,
    DBusTimeoutToggledFunction
toggled_function,
    void *data,
    DBusFreeFunction free_data_function)
{
    dbus_bool_t result;
    DBusTimeoutList *timeouts;

    _dbus_return_val_if_fail (server != NULL, FALSE);

    SERVER_LOCK (server);
    timeouts = server->timeouts;
    server->timeouts = NULL;

```

```

if (timeouts)
{
    SERVER_UNLOCK (server);
    result = _dbus_timeout_list_set_functions (timeouts,
                                              add_function,
                                              remove_function,
                                              toggled_function,
                                              data,
                                              free_data_function);

    SERVER_LOCK (server);
}
else
{
    _dbus_warn_check_failed ("Re-entrant call to %s\n",
    _DBUS_FUNCTION_NAME);
    result = FALSE;
}
server->timeouts = timeouts;
SERVER_UNLOCK (server);

return result;
}

/**
 * Sets the authentication mechanisms that this server offers to
 * clients, as a #NULL-terminated array of mechanism names. This
 * function only affects connections created <em>after</em> it is
 * called. Pass #NULL instead of an array to use all available
 * mechanisms (this is the default behavior).
 *
 * The D-Bus specification describes some of the supported mechanisms.
 *
 * @param server the server
 * @param mechanisms #NULL-terminated array of mechanisms
 * @returns #FALSE if no memory
 */
dbus_bool_t
dbus_server_set_auth_mechanisms (DBusServer *server,
                                const char **mechanisms)
{
    char **copy;

    _dbus_return_val_if_fail (server != NULL, FALSE);

    SERVER_LOCK (server);

    if (mechanisms != NULL)
    {
        copy = _dbus_dup_string_array (mechanisms);
        if (copy == NULL)
            return FALSE;
    }
}

```

```

else
    copy = NULL;

dbus_free_string_array (server->auth_mechanisms);
server->auth_mechanisms = copy;

SERVER_UNLOCK (server);

return TRUE;
}

static DBusDataSlotAllocator slot_allocator;
_DBUS_DEFINE_GLOBAL_LOCK (server_slots);

/**
 * Allocates an integer ID to be used for storing application-specific
 * data on any DBusServer. The allocated ID may then be used
 * with dbus_server_set_data() and dbus_server_get_data().
 * The slot must be initialized with -1. If a nonnegative
 * slot is passed in, the refcount is incremented on that
 * slot, rather than creating a new slot.
 *
 * The allocated slot is global, i.e. all DBusServer objects will have
 * a slot with the given integer ID reserved.
 *
 * @param slot_p address of global variable storing the slot ID
 * @returns #FALSE on no memory
 */
dbus_bool_t
dbus_server_allocate_data_slot (dbus_int32_t *slot_p)
{
    return _dbus_data_slot_allocator_alloc (&slot_allocator,
                                           (DBusRMutex
**)&_DBUS_LOCK_NAME (server_slots),
                                           slot_p);
}

/**
 * Deallocates a global ID for server data slots.
 * dbus_server_get_data() and dbus_server_set_data()
 * may no longer be used with this slot.
 * Existing data stored on existing DBusServer objects
 * will be freed when the server is finalized,
 * but may not be retrieved (and may only be replaced
 * if someone else reallocates the slot).
 *
 * @param slot_p address of the slot to deallocate
 */
void
dbus_server_free_data_slot (dbus_int32_t *slot_p)
{

```

```

    _dbus_return_if_fail (*slot_p >= 0);

    _dbus_data_slot_allocator_free (&slot_allocator, slot_p);
}

/**
 * Stores a pointer on a DBusServer, along
 * with an optional function to be used for freeing
 * the data when the data is set again, or when
 * the server is finalized. The slot number
 * must have been allocated with dbus_server_allocate_data_slot().
 *
 * @param server the server
 * @param slot the slot number
 * @param data the data to store
 * @param free_data_func finalizer function for the data
 * @returns #TRUE if there was enough memory to store the data
 */
dbus_bool_t
dbus_server_set_data (DBusServer      *server,
                     int              slot,
                     void             *data,
                     DBusFreeFunction free_data_func)
{
    DBusFreeFunction old_free_func;
    void *old_data;
    dbus_bool_t retval;

    _dbus_return_val_if_fail (server != NULL, FALSE);

    SERVER_LOCK (server);

    retval = _dbus_data_slot_list_set (&slot_allocator,
                                       &server->slot_list,
                                       slot, data, free_data_func,
                                       &old_free_func, &old_data);

    SERVER_UNLOCK (server);

    if (retval)
    {
        /* Do the actual free outside the server lock */
        if (old_free_func)
            (* old_free_func) (old_data);
    }

    return retval;
}

/**
 * Retrieves data previously set with dbus_server_set_data().

```

```

* The slot must still be allocated (must not have been freed).
*
* @param server the server
* @param slot the slot to get data from
* @returns the data, or #NULL if not found
*/
void*
dbus_server_get_data (DBusServer  *server,
                    int           slot)
{
    void *res;

    _dbus_return_val_if_fail (server != NULL, NULL);

    SERVER_LOCK (server);

    res = _dbus_data_slot_list_get (&slot_allocator,
                                    &server->slot_list,
                                    slot);

    SERVER_UNLOCK (server);

    return res;
}

/** @} */

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <string.h>

dbus_bool_t
_dbus_server_test (void)
{
    const char *valid_addresses[] = {
        "tcp:port=1234",
        "tcp:host=localhost,port=1234",
        "tcp:host=localhost,port=1234;tcp:port=5678",
#ifdef DBUS_UNIX
        "unix:path=./boogie",
        "tcp:port=1234;unix:path=./boogie",
#endif
    };

    DBusServer *server;
    int i;

    for (i = 0; i < _DBUS_N_ELEMENTS (valid_addresses); i++)
    {
        DBusError error = DBUS_ERROR_INIT;
        char *address;
        char *id;

```



```

        server = dbus_server_listen (valid_addresses[i], &error);
        if (server == NULL)
            {
                _dbus_warn ("server listen error: %s: %s\n", error.name,
error.message);
                dbus_error_free (&error);
                _dbus_assert_not_reached ("Failed to listen for valid
address.");
            }

        id = dbus_server_get_id (server);
        _dbus_assert (id != NULL);
        address = dbus_server_get_address (server);
        _dbus_assert (address != NULL);

        if (strstr (address, id) == NULL)
            {
                _dbus_warn ("server id '%s' is not in the server address
'%s'\n",
                            id, address);
                _dbus_assert_not_reached ("bad server id or address");
            }

        dbus_free (id);
        dbus_free (address);

        dbus_server_disconnect (server);
        dbus_server_unref (server);
    }

    return TRUE;
}

#endif /* DBUS_BUILD_TESTS */

```

File = dbus-server.h

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-server.h DBusServer object
 *
 * Copyright (C) 2002, 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or

```

```

* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/
#if !defined (DBUS_INSIDE_DBUS_H) && !defined (DBUS_COMPILATION)
#error "Only <dbus/dbus.h> can be included directly, this file may
disappear or change contents."
#endif

#ifndef DBUS_SERVER_H
#define DBUS_SERVER_H

#include <dbus/dbus-errors.h>
#include <dbus/dbus-message.h>
#include <dbus/dbus-connection.h>
#include <dbus/dbus-protocol.h>

DBUS_BEGIN_DECLS

/**
 * @addtogroup DbusServer
 * @{
 */

typedef struct DbusServer DbusServer;

/** Called when a new connection to the server is available. Must
reference and save the new
 * connection, or close the new connection. Set with
dbus_server_set_new_connection_function().
 */
typedef void (* DbusNewConnectionFunction) (DBusServer *server,
DBusConnection
*new_connection,
void *data);

DBUS_EXPORT
DBusServer* dbus_server_listen (const char *address,
DBusError *error);

DBUS_EXPORT
DBusServer* dbus_server_ref (DBusServer *server);
DBUS_EXPORT
void dbus_server_unref (DBusServer *server);

```

```

DBUS_EXPORT
void      dbus_server_disconnect      (DBusServer      *server);
DBUS_EXPORT
dbus_bool_t dbus_server_get_is_connected (DBusServer      *server);
DBUS_EXPORT
char*     dbus_server_get_address     (DBusServer      *server);
DBUS_EXPORT
char*     dbus_server_get_id          (DBusServer      *server);
DBUS_EXPORT
void      dbus_server_set_new_connection_function (DBusServer
*server,

DBusNewConnectionFunction  function,

void

*data,

DBusFreeFunction

free_data_function);
DBUS_EXPORT
dbus_bool_t dbus_server_set_watch_functions (DBusServer

DBusAddWatchFunction      add_function,

DBusRemoveWatchFunction   remove_function,

DBusWatchToggledFunction  toggled_function,

void

*data,

DBusFreeFunction

free_data_function);
DBUS_EXPORT
dbus_bool_t dbus_server_set_timeout_functions (DBusServer

DBusAddTimeoutFunction    add_function,

DBusRemoveTimeoutFunction remove_function,

DBusTimeoutToggledFunction toggled_function,

void

*data,

DBusFreeFunction

free_data_function);
DBUS_EXPORT
dbus_bool_t dbus_server_set_auth_mechanisms (DBusServer

const char

**mechanisms);

DBUS_EXPORT
dbus_bool_t dbus_server_allocate_data_slot (dbus_int32_t      *slot_p);
DBUS_EXPORT

```

```

void          dbus_server_free_data_slot      (dbus_int32_t      *slot_p);
DBUS_EXPORT
dbus_bool_t  dbus_server_set_data           (DBusServer       *server,
int             slot,
void           *data,
DBusFreeFunction

free_data_func);
DBUS_EXPORT
void*        dbus_server_get_data           (DBusServer       *server,
int             slot);

/** @} */

DBUS_END_DECLS

#endif /* DBUS_SERVER_H */

File = dbus-sha.c

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-sha.c SHA-1 implementation
 *
 * Copyright (C) 2003 Red Hat Inc.
 * Copyright (C) 1995 A. M. Kuchling
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

#include <config.h>
#include "dbus-internals.h"
#include "dbus-sha.h"
#include "dbus-marshal-basic.h" /* for byteswap routines */

```

```

#include <string.h>

/* The following comments have the history of where this code
 * comes from. I actually copied it from GNet in GNOME CVS.
 * - hp@redhat.com
 */

/*
 * sha.h : Implementation of the Secure Hash Algorithm
 *
 * Part of the Python Cryptography Toolkit, version 1.0.0
 *
 * Copyright (C) 1995, A.M. Kuchling
 *
 * Distribute and use freely; there are no restrictions on further
 * dissemination and usage except those imposed by the laws of your
 * country of residence.
 */

/* SHA: NIST's Secure Hash Algorithm */

/* Based on SHA code originally posted to sci.crypt by Peter Gutmann
   in message <30ajo5$oe8@ccu2.auckland.ac.nz>.
   Modified to test for endianness on creation of SHA objects by AMK.
   Also, the original specification of SHA was found to have a
weakness
   by NSA/NIST. This code implements the fixed version of SHA.
 */

/* Here's the first paragraph of Peter Gutmann's posting:

The following is my SHA (FIPS 180) code updated to allow use of the
"fixed"
SHA, thanks to Jim Gillogly and an anonymous contributor for the
information on
what's changed in the new version. The fix is a simple change which
involves
adding a single rotate in the initial expansion function. It is
unknown
whether this is an optimal solution to the problem which was
discovered in the
SHA or whether it's simply a bandaid which fixes the problem with a
minimum of
effort (for example the reengineering of a great many Capstone chips).
 */

/**
 * @defgroup DBusSHA SHA implementation
 * @ingroup DBusInternals
 * @brief SHA-1 hash
 */

```

```

* Types and functions related to computing SHA-1 hash.
*/

/**
 * @defgroup DBusSHAInternals SHA implementation details
 * @ingroup DBusInternals
 * @brief Internals of SHA implementation.
 *
 * The implementation of SHA-1 (see
http://www.itl.nist.gov/fipspubs/fip180-1.htm).
 * This SHA implementation was written by A.M. Kuchling
 *
 * @{
 */

#ifndef DOXYGEN_SHOULD_SKIP_THIS

/* The SHA block size and message digest sizes, in bytes */

#define SHA_DATASIZE    64
#define SHA_DIGESTSIZE 20

/* The SHA f()-functions. The f1 and f3 functions can be optimized to
save one boolean operation each - thanks to Rich Schroepel,
rcs@cs.arizona.edu for discovering this */

/*#define f1(x,y,z) ( ( x & y ) | ( ~x & z ) )           // Rounds 0-
19 */
#define f1(x,y,z) ( z ^ ( x & ( y ^ z ) ) )           /* Rounds 0-19
*/
#define f2(x,y,z) ( x ^ y ^ z )                       /* Rounds 20-39
*/
/*#define f3(x,y,z) ( ( x & y ) | ( x & z ) | ( y & z ) ) // Rounds
40-59 */
#define f3(x,y,z) ( ( x & y ) | ( z & ( x | y ) ) ) /* Rounds 40-59
*/
#define f4(x,y,z) ( x ^ y ^ z )                       /* Rounds 60-79
*/

/* The SHA Mysterious Constants */

#define K1 0x5A827999L                                /* Rounds 0-
19 */
#define K2 0x6ED9EBA1L                                /* Rounds 20-
39 */
#define K3 0x8F1BBCDCL                                /* Rounds 40-
59 */
#define K4 0xCA62C1D6L                                /* Rounds 60-
79 */

/* SHA initial values */

```

```

#define h0init 0x67452301L
#define h1init 0xEFCDAB89L
#define h2init 0x98BADCFEL
#define h3init 0x10325476L
#define h4init 0xC3D2E1F0L

/* Note that it may be necessary to add parentheses to these macros if
they
are to be called with expressions as arguments */
/* 32-bit rotate left - kludged with shifts */

#define ROTL(n,X) ( ( ( X ) << n ) | ( ( X ) >> ( 32 - n ) ) )

/* The initial expanding function. The hash function is defined over
an
80-word expanded input array W, where the first 16 are copies of
the input
data, and the remaining 64 are defined by

    W[ i ] = W[ i - 16 ] ^ W[ i - 14 ] ^ W[ i - 8 ] ^ W[ i - 3 ]

This implementation generates these values on the fly in a circular
buffer - thanks to Colin Plumb, colin@nyx10.cs.du.edu for this
optimization.

The updated SHA changes the expanding function by adding a rotate
of 1
bit. Thanks to Jim Gillogly, jim@rand.org, and an anonymous
contributor
for this information */

#define expand(W,i) ( W[ i & 15 ] = ROTL( 1, ( W[ i & 15 ] ^ W[ (i -
14) & 15 ] ^ \
                                                    W[ (i - 8) & 15 ] ^
W[ (i - 3) & 15 ] ) ) )

/* The prototype SHA sub-round. The fundamental sub-round is:

    a' = e + ROTL( 5, a ) + f( b, c, d ) + k + data;
    b' = a;
    c' = ROTL( 30, b );
    d' = c;
    e' = d;

but this is implemented by unrolling the loop 5 times and renaming
the
variables ( e, a, b, c, d ) = ( a', b', c', d', e' ) each
iteration.
This code is then replicated 20 times for each of the 4 functions,
using
the next 20 values from the W[] array each time */

```

```

#define subRound(a, b, c, d, e, f, k, data) \
    ( e += ROTL( 5, a ) + f( b, c, d ) + k + data, b = ROTL( 30, b ) )

#endif /* !DOXYGEN_SHOULD_SKIP_THIS */

/* Perform the SHA transformation. Note that this code, like MD5,
seems to
    break some optimizing compilers due to the complexity of the
expressions
    and the size of the basic block. It may be necessary to split it
into
    sections, e.g. based on the four subrounds

    Note that this corrupts the context->data area */

static void
SHATransform(dbus_uint32_t *digest, dbus_uint32_t *data)
{
    dbus_uint32_t A, B, C, D, E;      /* Local vars */
    dbus_uint32_t eData[16];         /* Expanded data */

    /* Set up first buffer and local data buffer */
    A = digest[0];
    B = digest[1];
    C = digest[2];
    D = digest[3];
    E = digest[4];
    memmove (eData, data, SHA_DATASIZE);

    /* Heavy mangling, in 4 sub-rounds of 20 iterations each. */
    subRound (A, B, C, D, E, f1, K1, eData[0]);
    subRound (E, A, B, C, D, f1, K1, eData[1]);
    subRound (D, E, A, B, C, f1, K1, eData[2]);
    subRound (C, D, E, A, B, f1, K1, eData[3]);
    subRound (B, C, D, E, A, f1, K1, eData[4]);
    subRound (A, B, C, D, E, f1, K1, eData[5]);
    subRound (E, A, B, C, D, f1, K1, eData[6]);
    subRound (D, E, A, B, C, f1, K1, eData[7]);
    subRound (C, D, E, A, B, f1, K1, eData[8]);
    subRound (B, C, D, E, A, f1, K1, eData[9]);
    subRound (A, B, C, D, E, f1, K1, eData[10]);
    subRound (E, A, B, C, D, f1, K1, eData[11]);
    subRound (D, E, A, B, C, f1, K1, eData[12]);
    subRound (C, D, E, A, B, f1, K1, eData[13]);
    subRound (B, C, D, E, A, f1, K1, eData[14]);
    subRound (A, B, C, D, E, f1, K1, eData[15]);
    subRound (E, A, B, C, D, f1, K1, expand ( eData, 16) );
    subRound (D, E, A, B, C, f1, K1, expand ( eData, 17) );
    subRound (C, D, E, A, B, f1, K1, expand ( eData, 18) );
    subRound (B, C, D, E, A, f1, K1, expand ( eData, 19) );
}

```



```

subRound (A, B, C, D, E, f4, K4, expand ( eData, 70) );
subRound (E, A, B, C, D, f4, K4, expand ( eData, 71) );
subRound (D, E, A, B, C, f4, K4, expand ( eData, 72) );
subRound (C, D, E, A, B, f4, K4, expand ( eData, 73) );
subRound (B, C, D, E, A, f4, K4, expand ( eData, 74) );
subRound (A, B, C, D, E, f4, K4, expand ( eData, 75) );
subRound (E, A, B, C, D, f4, K4, expand ( eData, 76) );
subRound (D, E, A, B, C, f4, K4, expand ( eData, 77) );
subRound (C, D, E, A, B, f4, K4, expand ( eData, 78) );
subRound (B, C, D, E, A, f4, K4, expand ( eData, 79) );

/* Build message digest */
digest[0] += A;
digest[1] += B;
digest[2] += C;
digest[3] += D;
digest[4] += E;
}

/* When run on a little-endian CPU we need to perform byte reversal on
an
array of longwords. */

#ifdef WORDS_BIGENDIAN
#define swap_words(buffer, byte_count)
#else
static void
swap_words (dbus_uint32_t *buffer,
            int           byte_count)
{
    byte_count /= sizeof (dbus_uint32_t);
    while (byte_count--)
    {
        *buffer = DBUS_UINT32_SWAP_LE_BE (*buffer);
        ++buffer;
    }
}
#endif

static void
sha_init (DBusSHAContext *context)
{
    /* Set the h-vars to their initial values */
    context->digest[0] = h0init;
    context->digest[1] = h1init;
    context->digest[2] = h2init;
    context->digest[3] = h3init;
    context->digest[4] = h4init;

    /* Initialise bit count */
    context->count_lo = context->count_hi = 0;
}

```

```

static void
sha_append (DBusSHAContext      *context,
            const unsigned char *buffer,
            unsigned int         count)
{
    dbus_uint32_t tmp;
    unsigned int dataCount;

    /* Update bitcount */
    tmp = context->count_lo;
    if (( context->count_lo = tmp + ( ( dbus_uint32_t) count << 3) ) <
tmp)
        context->count_hi++;          /* Carry from low to high */
    context->count_hi += count >> 29;

    /* Get count of bytes already in data */
    dataCount = (int) (tmp >> 3) & 0x3F;

    /* Handle any leading odd-sized chunks */
    if (dataCount)
    {
        unsigned char *p = (unsigned char *) context->data + dataCount;

        dataCount = SHA_DATASIZE - dataCount;
        if (count < dataCount)
        {
            memmove (p, buffer, count);
            return;
        }
        memmove (p, buffer, dataCount);
        swap_words (context->data, SHA_DATASIZE);
        SHATransform (context->digest, context->data);
        buffer += dataCount;
        count -= dataCount;
    }

    /* Process data in SHA_DATASIZE chunks */
    while (count >= SHA_DATASIZE)
    {
        memmove (context->data, buffer, SHA_DATASIZE);
        swap_words (context->data, SHA_DATASIZE);
        SHATransform (context->digest, context->data);
        buffer += SHA_DATASIZE;
        count -= SHA_DATASIZE;
    }

    /* Handle any remaining bytes of data. */
    memmove (context->data, buffer, count);
}

```

```

/* Final wrapup - pad to SHA_DATASIZE-byte boundary with the bit
pattern
 1 0* (64-bit count of bits processed, MSB-first) */

static void
sha_finish (DBusSHAContext *context, unsigned char digest[20])
{
    int count;
    unsigned char *data_p;

    /* Compute number of bytes mod 64 */
    count = (int) context->count_lo;
    count = (count >> 3) & 0x3F;

    /* Set the first char of padding to 0x80.  This is safe since there
is
    always at least one byte free */
    data_p = (unsigned char *) context->data + count;
    *data_p++ = 0x80;

    /* Bytes of padding needed to make 64 bytes */
    count = SHA_DATASIZE - 1 - count;

    /* Pad out to 56 mod 64 */
    if (count < 8)
    {
        /* Two lots of padding:  Pad the first block to 64 bytes */
        memset (data_p, 0, count);
        swap_words (context->data, SHA_DATASIZE);
        SHATransform (context->digest, context->data);

        /* Now fill the next block with 56 bytes */
        memset (context->data, 0, SHA_DATASIZE - 8);
    }
    else
        /* Pad block to 56 bytes */
        memset (data_p, 0, count - 8);

    /* Append length in bits and transform */
    context->data[14] = context->count_hi;
    context->data[15] = context->count_lo;

    swap_words (context->data, SHA_DATASIZE - 8);
    SHATransform (context->digest, context->data);
    swap_words (context->digest, SHA_DIGESTSIZE);
    memmove (digest, context->digest, SHA_DIGESTSIZE);
}

/** @} */ /* End of internals */

/**
 * @addtogroup DBusSHA

```

```

*
* @{}
*/

/**
 * Initializes the SHA context.
 *
 * @param context an uninitialized context, typically on the stack.
 */
void
_dbus_sha_init (DBusSHAContext *context)
{
    sha_init (context);
}

/**
 * Feeds more data into an existing shasum computation.
 *
 * @param context the SHA context
 * @param data the additional data to hash
 */
void
_dbus_sha_update (DBusSHAContext *context,
                  const DBusString *data)
{
    unsigned int inputLen;
    const unsigned char *input;

    input = (const unsigned char*) _dbus_string_get_const_data (data);
    inputLen = _dbus_string_get_length (data);

    sha_append (context, input, inputLen);
}

/**
 * SHA finalization. Ends an SHA message-digest operation, writing the
 * the message digest and zeroing the context. The results are
 * returned as a raw 20-byte digest, not as the ascii-hex-digits
 * string form of the digest.
 *
 * @param context the SHA context
 * @param results string to append the 20-byte SHA digest to
 * @returns #FALSE if not enough memory to append the digest
 */
dbus_bool_t
_dbus_sha_final (DBusSHAContext *context,
                 DBusString *results)
{
    unsigned char digest[20];

    sha_finish (context, digest);
}

```

```

    if (!_dbus_string_append_len (results, digest, 20))
        return FALSE;

    /* some kind of security paranoia, though it seems pointless
     * to me given the nonzeroed stuff flying around
     */
    _DBUS_ZERO(*context);

    return TRUE;
}

/**
 * Computes the ASCII hex-encoded shasum of the given data and
 * appends it to the output string.
 *
 * @param data input data to be hashed
 * @param ascii_output string to append ASCII shasum to
 * @returns #FALSE if not enough memory
 */
dbus_bool_t
_dbus_sha_compute (const DBusString *data,
                  DBusString *ascii_output)
{
    DBusSHAContext context;
    DBusString digest;

    _dbus_sha_init (&context);

    _dbus_sha_update (&context, data);

    if (!_dbus_string_init (&digest))
        return FALSE;

    if (!_dbus_sha_final (&context, &digest))
        goto error;

    if (!_dbus_string_hex_encode (&digest, 0, ascii_output,
                                  _dbus_string_get_length
(ascii_output)))
        goto error;

    _dbus_string_free (&digest);

    return TRUE;

error:
    _dbus_string_free (&digest);
    return FALSE;
}

/** @} */ /* end of exported functions */

```

```

#ifdef DBUS_BUILD_TESTS
#include "dbus-test.h"
#include <stdio.h>

static dbus_bool_t
check_sha_binary (const unsigned char *input,
                  int                  input_len,
                  const char          *expected)
{
    DBusString input_str;
    DBusString expected_str;
    DBusString results;

    _dbus_string_init_const_len (&input_str, input, input_len);
    _dbus_string_init_const (&expected_str, expected);

    if (!_dbus_string_init (&results))
        _dbus_assert_not_reached ("no memory for SHA-1 results");

    if (!_dbus_sha_compute (&input_str, &results))
        _dbus_assert_not_reached ("no memory for SHA-1 results");

    if (!_dbus_string_equal (&expected_str, &results))
    {
        _dbus_warn ("Expected hash %s got %s for SHA-1 sum\n",
                   expected,
                   _dbus_string_get_const_data (&results));
        _dbus_string_free (&results);
        return FALSE;
    }

    _dbus_string_free (&results);
    return TRUE;
}

static dbus_bool_t
check_sha_str (const char *input,
               const char *expected)
{
    return check_sha_binary (input, strlen (input), expected);
}

static dbus_bool_t
decode_compact_string (const DBusString *line,
                       DBusString      *decoded)
{
    int n_bits;
    dbus_bool_t current_b;
    int offset;
    int next;
    long val;

```

```

int length_bytes;

offset = 0;
next = 0;

if (!_dbus_string_parse_int (line, offset, &val, &next))
{
    fprintf (stderr, "could not parse length at start of compact
string: %s\n",
            _dbus_string_get_const_data (line));
    return FALSE;
}

_dbus_string_skip_blank (line, next, &next);

offset = next;
if (!_dbus_string_parse_int (line, offset, &val, &next))
{
    fprintf (stderr, "could not parse start bit 'b' in compact
string: %s\n",
            _dbus_string_get_const_data (line));
    return FALSE;
}

if (!(val == 0 || val == 1))
{
    fprintf (stderr, "the value 'b' must be 0 or 1, see sha-
1/Readme.txt\n");
    return FALSE;
}

_dbus_string_skip_blank (line, next, &next);

current_b = val;
n_bits = 0;

while (next < _dbus_string_get_length (line))
{
    int total_bits;

    offset = next;

    if (_dbus_string_get_byte (line, offset) == '^')
        break;

    if (!_dbus_string_parse_int (line, offset, &val, &next))
    {
        fprintf (stderr, "could not parse bit count in compact
string\n");
        return FALSE;
    }
}

```



```

    /* We now append "val" copies of "current_b" bits to the string
*/
    total_bits = n_bits + val;
    while (n_bits < total_bits)
    {
        int byte_containing_next_bit = n_bits / 8;
        int bit_containing_next_bit = 7 - (n_bits % 8);
        unsigned char old_byte;

        if (byte_containing_next_bit >= _dbus_string_get_length
(decoded))
        {
            if (!_dbus_string_set_length (decoded,
byte_containing_next_bit + 1))
                _dbus_assert_not_reached ("no memory to extend to next
byte");
        }

        old_byte = _dbus_string_get_byte (decoded,
byte_containing_next_bit);
        old_byte |= current_b << bit_containing_next_bit;

#ifdef 0
        printf ("Appending bit %d to byte %d at bit %d resulting in
byte 0x%x\n",
                current_b, byte_containing_next_bit,
                bit_containing_next_bit, old_byte);
#endif

        _dbus_string_set_byte (decoded, byte_containing_next_bit,
old_byte);

        ++n_bits;
    }

    _dbus_string_skip_blank (line, next, &next);

    current_b = !current_b;
}

length_bytes = (n_bits / 8 + ((n_bits % 8) ? 1 : 0));

if (_dbus_string_get_length (decoded) != length_bytes)
{
    fprintf (stderr, "Expected length %d bytes %d bits for compact
string, got %d bytes\n",
            length_bytes, n_bits, _dbus_string_get_length
(decoded));
    return FALSE;
}
else
    return TRUE;

```



```

        _dbus_string_set_byte (result, i, 'd');
        break;
    case 'E':
        _dbus_string_set_byte (result, i, 'e');
        break;
    case 'F':
        _dbus_string_set_byte (result, i, 'f');
        break;
    case '^':
    case ' ':
        _dbus_string_delete (result, i, 1);
        --i; /* to offset ++i below */
        break;
    }

    ++i;
}

break;
}
}

retval = TRUE;

/* out: */
_dbus_string_free (&line);
return retval;
}

static dbus_bool_t
process_test_data (const char *test_data_dir)
{
    DBusString tests_file;
    DBusString results_file;
    DBusString tests;
    DBusString results;
    DBusString line;
    DBusString tmp;
    int line_no;
    dbus_bool_t retval;
    int success_count;
    DBusError error = DBUS_ERROR_INIT;

    retval = FALSE;

    if (!_dbus_string_init (&tests_file))
        _dbus_assert_not_reached ("no memory");

    if (!_dbus_string_init (&results_file))
        _dbus_assert_not_reached ("no memory");

    if (!_dbus_string_init (&tests))

```

```

    _dbus_assert_not_reached ("no memory");

if (!_dbus_string_init (&results))
    _dbus_assert_not_reached ("no memory");

if (!_dbus_string_init (&line))
    _dbus_assert_not_reached ("no memory");

if (!_dbus_string_append (&tests_file, test_data_dir))
    _dbus_assert_not_reached ("no memory");

if (!_dbus_string_append (&results_file, test_data_dir))
    _dbus_assert_not_reached ("no memory");

_dbus_string_init_const (&tmp, "sha-1/byte-messages.sha1");
if (!_dbus_concat_dir_and_file (&tests_file, &tmp))
    _dbus_assert_not_reached ("no memory");

_dbus_string_init_const (&tmp, "sha-1/byte-hashes.sha1");
if (!_dbus_concat_dir_and_file (&results_file, &tmp))
    _dbus_assert_not_reached ("no memory");

if (!_dbus_file_get_contents (&tests, &tests_file, &error))
{
    fprintf (stderr, "could not load test data file %s: %s\n",
            _dbus_string_get_const_data (&tests_file),
            error.message);
    dbus_error_free (&error);
    goto out;
}

if (!_dbus_file_get_contents (&results, &results_file, &error))
{
    fprintf (stderr, "could not load results data file %s: %s\n",
            _dbus_string_get_const_data (&results_file),
error.message);
    dbus_error_free (&error);
    goto out;
}

success_count = 0;
line_no = 0;
next_iteration:
while (_dbus_string_pop_line (&tests, &line))
{
    line_no += 1;

    _dbus_string_delete_leading_blanks (&line);

    if (_dbus_string_get_length (&line) == 0)
        goto next_iteration;
    else if (_dbus_string_starts_with_c_str (&line, "#"))

```

```

    goto next_iteration;
else if (_dbus_string_starts_with_c_str (&line, "H>"))
{
    printf ("SHA-1: %s\n", _dbus_string_get_const_data (&line));

    if (_dbus_string_find (&line, 0, "Type 3", NULL))
    {
        /* See sha-1/Readme.txt - the "Type 3" tests are
        * random seeds, rather than data to be hashed.
        * we'd have to do a little bit more implementation
        * to use those tests.
        */

        printf (" (ending tests due to Type 3 tests seen - this
is normal)\n");
        break;
    }
}
else if (_dbus_string_starts_with_c_str (&line, "D>") ||
        _dbus_string_starts_with_c_str (&line, "<D"))
    goto next_iteration;
else
{
    DBusString test;
    DBusString result;
    DBusString next_line;
    DBusString expected;
    dbus_bool_t success;

    success = FALSE;

    if (!_dbus_string_init (&next_line))
        _dbus_assert_not_reached ("no memory");

    if (!_dbus_string_init (&expected))
        _dbus_assert_not_reached ("no memory");

    if (!_dbus_string_init (&test))
        _dbus_assert_not_reached ("no memory");

    if (!_dbus_string_init (&result))
        _dbus_assert_not_reached ("no memory");

    /* the "compact strings" are "^"-terminated not
    * newline-terminated so readahead to find the
    * "^"
    */
    while (!_dbus_string_find (&line, 0, "^", NULL) &&
            _dbus_string_pop_line (&tests, &next_line))
    {
        if (!_dbus_string_append_byte (&line, ' ') ||
            !_dbus_string_move (&next_line, 0, &line,

```

```

                                _dbus_string_get_length
(&line))
        _dbus_assert_not_reached ("no memory");
    }

    if (!decode_compact_string (&line, &test))
    {
string\n",
        fprintf (stderr, "Failed to decode line %d as a compact
                line_no);
        goto failure;
    }

    if (!_dbus_sha_compute (&test, &result))
        _dbus_assert_not_reached ("no memory for SHA-1 result");

    if (!get_next_expected_result (&results, &expected))
    {
        fprintf (stderr, "Failed to read an expected result\n");
        goto failure;
    }

    if (!_dbus_string_equal (&result, &expected))
    {
        fprintf (stderr, " for line %d got hash %s expected
%s\n",
                line_no,
                _dbus_string_get_const_data (&result),
                _dbus_string_get_const_data (&expected));
        goto failure;
    }
    else
    {
        success_count += 1;
    }

    success = TRUE;

failure:
    _dbus_string_free (&test);
    _dbus_string_free (&result);
    _dbus_string_free (&next_line);
    _dbus_string_free (&expected);

    if (!success)
        goto out;
}

}

retval = TRUE;

printf ("Passed the %d SHA-1 tests in the test file\n",

```

```

        success_count);

out:
    _dbus_string_free (&tests_file);
    _dbus_string_free (&results_file);
    _dbus_string_free (&tests);
    _dbus_string_free (&results);
    _dbus_string_free (&line);

    return retval;
}

/**
 * @ingroup DBusSHAInternals
 * Unit test for SHA computation.
 *
 * @returns #TRUE on success.
 */
dbus_bool_t
_dbus_sha_test (const char *test_data_dir)
{
    unsigned char all_bytes[256];
    int i;

    if (test_data_dir != NULL)
    {
        if (!process_test_data (test_data_dir))
            return FALSE;
    }
    else
        printf ("No test data dir\n");

    i = 0;
    while (i < 256)
    {
        all_bytes[i] = i;
        ++i;
    }

    if (!check_sha_binary (all_bytes, 256,
                           "4916d6bdb7f78e6803698cab32d1586ea457dfc8"))
        return FALSE;

#define CHECK(input,expected) if (!check_sha_str (input, expected))
return FALSE

    CHECK ("", "da39a3ee5e6b4b0d3255bfef95601890afd80709");
    CHECK ("a", "86f7e437faa5a7fce15d1ddcb9eaeaea377667b8");
    CHECK ("abc", "a9993e364706816aba3e25717850c26c9cd0d89d");
    CHECK ("message digest",
"12252ceda8be8994d5fa0290a47231c1d16aae3");

```

```

    CHECK ("abcdefghijklmnopqrstuvwxy",
"32d10c7b8cf96570ca04ce37f2a19d84240d3a89");
    CHECK
("ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxy0123456789",
    "761c457bf73b14d27e9e9265c46f4b4dda11f940");
    CHECK
("12345678901234567890123456789012345678901234567890123456789012345678
901234567890",
    "50abf5706a150990a08b2c5ea40fa0e585554732");

    return TRUE;
}

```

```
#endif /* DBUS_BUILD_TESTS */
```

```
File = dbus-sha.h
```

```

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-sha.h SHA-1 implementation
 *
 * Copyright (C) 2003 Red Hat Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
 */

```

```
#ifndef DBUS_SHA_H
#define DBUS_SHA_H
```

```
#include <dbus/dbus-macros.h>
#include <dbus/dbus-errors.h>
#include <dbus/dbus-string.h>
```

```
DBUS_BEGIN_DECLS
```



```

typedef struct DBusSHAContext DBusSHAContext;

/**
 * Struct storing state of the SHA algorithm
 */
struct DBusSHAContext
{
    dbus_uint32_t  digest[5];           /**< Message digest */
    dbus_uint32_t  count_lo;           /**< 64-bit bit count */
    dbus_uint32_t  count_hi;           /**< No clue */
    dbus_uint32_t  data[16];           /**< SHA data buffer */
};

void      _dbus_sha_init      (DBusSHAContext  *context);
void      _dbus_sha_update    (DBusSHAContext  *context,
                               const DBusString *data);
dbus_bool_t _dbus_sha_final    (DBusSHAContext  *context,
                               DBusString       *results);
dbus_bool_t _dbus_sha_compute (const DBusString *data,
                               DBusString       *ascii_output);

DBUS_END_DECLS

#endif /* DBUS_SHA_H */

File = dbus-shared.h

/* -*- mode: C; c-file-style: "gnu"; indent-tabs-mode: nil; -*- */
/* dbus-shared.h  Stuff used by both dbus/dbus.h low-level and C/C++
binding APIs
 *
 * Copyright (C) 2004 Red Hat, Inc.
 *
 * Licensed under the Academic Free License version 2.1
 *
 * This program is free software; you can redistribute it and/or
modify
 * it under the terms of the GNU General Public License as published
by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software

```

```

* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA
*
*/

#ifndef DBUS_SHARED_H
#define DBUS_SHARED_H

/* Don't include anything in here from anywhere else. It's
* intended for use by any random library.
*/

#ifdef __cplusplus
extern "C" {
#if 0
} /* avoids confusing emacs indentation */
#endif
#endif

/* Normally docs are in .c files, but there isn't a .c file for this.
*/
/**
* @defgroup DBusShared Shared constants
* @ingroup DBus
*
* @brief Shared header included by both libdbus and C/C++ bindings
such as the GLib bindings.
*
* Usually a C/C++ binding such as the GLib or Qt binding won't want
to include dbus.h in its
* public headers. However, a few constants and macros may be useful
to include; those are
* found here and in dbus-protocol.h
*
* @{
*/

/**
* Well-known bus types. See dbus_bus_get().
*/
typedef enum
{
    DBUS_BUS_SESSION,    /**< The login session bus */
    DBUS_BUS_SYSTEM,    /**< The systemwide bus */
    DBUS_BUS_STARTER    /**< The bus that started us, if any */
} DBusBusType;

/**
* Results that a message handler can return.
*/
typedef enum

```

```

{
    DBUS_HANDLER_RESULT_HANDLED,          /**< Message has had its effect
- no need to run more handlers. */
    DBUS_HANDLER_RESULT_NOT_YET_HANDLED, /**< Message has not had any
effect - see if other handlers want it. */
    DBUS_HANDLER_RESULT_NEED_MEMORY      /**< Need more memory in order
to return #DBUS_HANDLER_RESULT_HANDLED or
#DBUS_HANDLER_RESULT_NOT_YET_HANDLED. Please try again later with more
memory. */
} DbusHandlerResult;

/* Bus names */

/** The bus name used to talk to the bus itself. */
#define DBUS_SERVICE_DBUS      "org.freedesktop.DBus"

/* Paths */
/** The object path used to talk to the bus itself. */
#define DBUS_PATH_DBUS        "/org/freedesktop/DBus"
/** The object path used in local/in-process-generated messages. */
#define DBUS_PATH_LOCAL      "/org/freedesktop/DBus/Local"

/* Interfaces, these #define don't do much other than
 * catch typos at compile time
 */
/** The interface exported by the object with #DBUS_SERVICE_DBUS and
#DBUS_PATH_DBUS */
#define DBUS_INTERFACE_DBUS      "org.freedesktop.DBus"
/** The interface supported by introspectable objects */
#define DBUS_INTERFACE_INTROSPECTABLE
"org.freedesktop.DBus.Introspectable"
/** The interface supported by objects with properties */
#define DBUS_INTERFACE_PROPERTIES
"org.freedesktop.DBus.Properties"
/** The interface supported by most dbus peers */
#define DBUS_INTERFACE_PEER      "org.freedesktop.DBus.Peer"

/** This is a special interface whose methods can only be invoked
 * by the local implementation (messages from remote apps aren't
 * allowed to specify this interface).
 */
#define DBUS_INTERFACE_LOCAL      "org.freedesktop.DBus.Local"

/* Owner flags */
#define DBUS_NAME_FLAG_ALLOW_REPLACEMENT 0x1 /**< Allow another
service to become the primary owner if requested */
#define DBUS_NAME_FLAG_REPLACE_EXISTING 0x2 /**< Request to replace
the current primary owner */
#define DBUS_NAME_FLAG_DO_NOT_QUEUE      0x4 /**< If we can not become
the primary owner do not place us in the queue */

/* Replies to request for a name */

```

```

#define DBUS_REQUEST_NAME_REPLY_PRIMARY_OWNER 1 /**< Service has
become the primary owner of the requested name */
#define DBUS_REQUEST_NAME_REPLY_IN_QUEUE 2 /**< Service could
not become the primary owner and has been placed in the queue */
#define DBUS_REQUEST_NAME_REPLY_EXISTS 3 /**< Service is
already in the queue */
#define DBUS_REQUEST_NAME_REPLY_ALREADY_OWNER 4 /**< Service is
already the primary owner */

/* Replies to releasing a name */
#define DBUS_RELEASE_NAME_REPLY_RELEASED 1 /**< Service was
released from the given name */
#define DBUS_RELEASE_NAME_REPLY_NON_EXISTENT 2 /**< The given name
does not exist on the bus */
#define DBUS_RELEASE_NAME_REPLY_NOT_OWNER 3 /**< Service is not
an owner of the given name */

/* Replies to service starts */
#define DBUS_START_REPLY_SUCCESS 1 /**< Service was auto
started */
#define DBUS_START_REPLY_ALREADY_RUNNING 2 /**< Service was already
running */

/** @} */

#ifdef __cplusplus
#if 0
{ /* avoids confusing emacs indentation */
#endif
}
#endif

#endif /* DBUS_SHARED_H */

```